A Novel Energy Efficient Wireless Sensor Network Framework for Object Tracking

Muhammad Jasim Saeed

A thesis submitted in partial fulfilment of the requirements of the Manchester Metropolitan University for the degree of Doctor of Philosophy

Future Networks and Distributed Systems Research Group
School of Computing Mathematics and Digital Technology

July 2017

Acknowledgements

First and foremost, I would like to express my most sincere gratitude and thanks to my supervisors Professor Liangxiu Han and Dr. Mohammmad Hammoudeh and especially to Professor Liangxiu Han for her continuous and sustained support, encouragement and patience throughout this research project. I am fortunate for having had her insight and guidance to proceed through this thesis successfully.

I also want to thank other members of faculty for their help and support, and the administrative staff at the School of Computing Mathematics and Digital Technology. I would also like to thank my fellow research students for their advice and encouragement throughout this research project.

My friends who are either associates with MMU or not have been a great deal of help and support to sustain me throughout this research project and they all deserve my thanks and gratitude.

My family has been a great pillar of support not only through this research project but also through out my educational career and without their prayers and guidance I would never have been able to reach this milestone in my life.

I would specially thank my parents for their unending support, love and encouragement through some of the toughest years of my life and I hope to live up to their expectations in my personal and professional life.

Publications

1. M. J. Saeed, L. Han, and M. K.Muyeba, "An energy efficient and resource preserving target tracking approach for wireless sensor networks," in 9th IEEE/IET International Symposium on Communication Systems, Networks Digital Sign (CSNDSP), pp. 232 237, July 2014.

2. M. J. Saeed, L. Han and M. Hammoudeh, "Energy Efficient Sampling Mechanism for Object Tracking with Wireless Sensor Networks," WorldS4 2017, February 2017.

Abstract

Object tracking is a typical application of Wireless Sensor Networks (WSNs), which refers to the process of locating a moving object (or multiple objects) over time using a sensor network. Object tracking in WSNs can be a time consuming and resource hungry process due to factors, such as the amount of data generated or limited resources available to the sensor network.

The traditional centralised approaches where a number of sensors transmit all information to a base station or a sink node, increase computation burden. More recently static or dynamic clustering approaches have been explored. Both clustering approaches suffer from certain problems, such as, large clusters, redundant data collection and excessive energy consumption. In addition, most existing object tracking algorithms mainly focus on tracking an object instead of predicting the destination of an object.

To address the limitations of existing approaches, this thesis presents a novel framework for efficient object tracking using sensor networks. It consists of a Hierarchical Hybrid Clustering Mechanism (HHCM) with a Prediction-based Algorithm for Destination-estimation (PAD). The proposed framework can track the destination of the object without prior information of the objects movement, while providing significant reduction in energy consumption. The costs of computation and communication are also reduced by collecting the most relevant information and discarding irrelevant information at the initial stages of communication. The contributions of this thesis are:

Firstly, a novel Prediction-based Algorithm for Destination-estimation (PAD) has been presented, that predicts the final destination of the object and the path that particular object will take to that destination. The principles of origin destination (OD) estimation have been adopted to create a set of trajectories that a particular object could follow. These paths are made up of a number of mini-clusters, formed for tracking the object, combined together. PAD also contains a Multi-level Recovery Mechanism (MRM) that recovers tracking if the object is lost. MRM minimises the number of nodes involved in the recovery process by initiating the process at local level and then expanding to add more nodes till the object is recovered.

Secondly, a network architecture called Hierarchical Hybrid Clustering Mechanism (HHCM) has been developed, that forms dynamic mini-clusters within and across static clusters to reduce the number of nodes involved in the tracking process and to distribute the initial computational tasks amoung a larger number of mini-cluster heads.

Lastly, building upon the HHCM to create a novel multi-hierarchy aggregation and next-step prediction mechanism to gather the most relevant data about the movement of the tracked object and its next-step location, a Kalman-filter based approach for prediction of next state of an object in order to increase accuracy has been proposed. In addition, a dynamic sampling mechanism has been devised to collect the most relevant data.

Extensive simulations were carried out and results were compared with the existing approaches to prove that HHCM and PAD make significant improvements in energy conservation. To the best of my knowledge the framework developed in unique and novel, which can predicts the destination of the moving object without any prior historic knowledge of the moving object.

Contents

List of Figures

List of Tables

Abbreviations

WSN	Wireless sensor network
BS	Base station
WAL	Weighted average localization
MSSL	Maximum signal strength localization
OCO	Optimized Communication and Organization
HPS	Hierarchical prediction strategy
LSM	Least square method
PAD	Prediction-based mechanism for destination-estimation
HHCM	Hierarchical hybrid clustering mechanism
KF	Kalman Filtering
MDP	Markov Decision Process
STUN	Scalable tracking using Networked Sensors
DCTC	Dynamic convoy tree-based collaboration
DAB	Drain and balance tree structure
DAT	Deviation-avoidance tree
CH	Cluster head
DPT	Distributed Predictive Tracking
DSTC	Dynamic space-time clustering
PES	Prediction-based energy saving scheme
CODA	Continuous object detection and tracking algorithm
HCCT	Hybrid cluster-based target tracking
LESOP	Low energy self-organising protocol
IDSQ	Information-driven sensor querying

Chapter 1

Introduction

Wireless sensor networks (WSNs) have been widely used in military and non-military environments due to the advancement in wireless communications, microelectronics, embedded microprocessors and networking technologies. Some typical examples of WSNs are environmental monitoring, battlefield surveillance, space exploration, health care, emergency response, disaster discovery, tracking of humans in crowded and restricted areas, tracking of vehicles such as cars in highways, hazardous environment exploration and seismic sensing [2, 3].

Object tracking is an important topic in WSNs, it is the process of locating a moving object (or multiple objects) over time using a sensor network with heterogeneous sensors. Object tracking can be a time consuming and resource hungry process due to multiple factors, such as the amount of data that can be generated and coordination with other nodes in the network through communication.

Due to the nature of WSNs, when tracking an object, the limitations of a typical sensor must be considered. A typical sensor node is made up of five components, battery, memory, processor, transceiver and a sensor. All five components have limited resources and any WSN system would have to utilise all of these resources efficiently for the network's longevity and efficiency. Sensing the environment incurs significant energy costs where as collection of large amounts of data during object tracking can lead to reduction in memory and processing efficiencies incurring delays. Whereas, transmitting large amounts of data can lead to excessive energy consumption by the transceiver. Any efficient object tracking network should be able to track an object with minimal delay and errors while also conserving energy.

1.1 Motivation and Background

Traditional object tracking methods for Wireless Sensor Networks make use of a centralized approach [4–6]. As the number of sensors rise in the network, more messages are passed on towards the Base Station (BS) and will consume additional bandwidth. In [4, 5], the authors have proposed two mechanisms, utilizing the centralized method, called weighted average localization (WAL) and maximum signal strength localization (MSSL) to evaluate their performance over WSN [7]. In these mechanisms, all sensor nodes are kept in active state to monitor the target that passes by within their sensing area. The nodes that detect the mobile target should forward their data directly to the base station to estimate the target. However, a major drawback of the centralized approaches is that the energy consumption per node is high resulting in unacceptable overall system energy consumption. Thus, this approach is not fault tolerant as there is a single point of failure at the nodes near the BS and lacks scalability. Moreover in traditional object tracking methods, sensing task is usually performed by one node at a time resulting in less accuracy and heavy computational burden on that node. With limited energy resources available to the node, traditional tracking methods based on a complex signal processing algorithm are not useful.

To overcome the shortcomings of the traditional approaches more complex object tracing techniques have been proposed in the recent years. These approaches are based on collaborative node arrangements such as clustering [8–15] or tree structures [1, 5, 13–20], bringing significant improvements to the traditional approaches.

Within Tree-based approaches, the hierarchical trees are formed dynamically according to the target movement in a networked area. A well-known tree-based tracking approach is Optimized Communication and Organization (OCO) [4, 5, 12] which provides self-organizing and routing capabilities of sensor nodes during the tracking process. The tracking process is performed in four phases, position collecting, processing, tracking, and maintenance. The major drawback of this approach is that the border sensor nodes are activated permanently. Therefore, the energy of these nodes may be depleted rapidly. Generally, tree-based approaches are not scalable and need to be evaluated from tracking accuracy perspective. In prediction-based approaches, a sensor can predict the future movement of the target based on a history of its past locations over time. Thus, the sensors' states (active, sleep) can be easily controlled [18, 21–24]. The main idea of these approaches is that the sensor nodes use past estimated locations collected during the target movement to activate a specific set of nodes in a range where the target may move toward. In [25, 26], maneuvering target tracking-Mobicast (MTT-Mobicast) algorithm is proposed to utilize the mobicast message target tracking approaches. It is a spatio-temporal multicast algorithm that distributes messages to the sensor nodes located in spatial zones that evolve over time in a predictable manner. However, the design of the MTT-Mobicast protocol faces challenges [27] in developing a fully distributed scheme to construct some special zones that can limit unnecessary retransmissions and ensure receiving the required message by all participating nodes.

Within clustering approaches, nodes are arranged into clusters which are either static or dynamic. Static clusters contain nodes within a region with a fixed structure [28–33]. Other attributes of the network remain static as well, such as, size, coverage area and member nodes. Static clustering architecture [27, 31–34], however, suffers from several drawbacks, such as, fixed membership is not fault tolerant. Also, the cluster can only exist as long as the cluster head is active, in case of power depletion of the cluster head, all the sensors in the cluster are rendered useless. Fixed membership also prevents sensor nodes in different clusters from sharing information and conduct collaborative data computation. For object tracking this approach can lead to excessive use of certain nodes where as certain nodes in a cluster could remain under utilised [34]. Dynamic clusters provide a mechanism where clusters are formed for a specific activity and are then dissolved. This provides a mechanism where nodes are equally used based on available resources, however, creation and dissolution of clusters itself can incur excessive overheads. While based on the size, each cluster can also incur noticeable communication costs [9, 35, 36].

Not many object tracking techniques have been proposed based on purely static clustering. However, mechanism in combination with dynamic clustering have been proposed

[1, 9, 28, 35, 36], refereed to as hybrid clustering. Hybrid tracking approaches for target tracking merge more than one approach, of previously described ones, for the sake of mitigating their individual drawbacks as much as possible. A well-known algorithm that follows these approaches is Hierarchical Prediction Strategy (HPS) which combines a cluster approach along with a certain prediction algorithm [37]. In HSP, the cluster is formed using Voronoi division and the next location of mobile target is predicated upon least square method (LSM) [38]. The major disadvantage of such algorithms is the extra complexity added upon combing target tracking approaches based on the application requirements leading to an additional increase in the system energy consumption. Although, the existing approaches have their merits and demerits but in the context of object tracking, node organisation does not guarantee efficient resource utilisation. For this purpose the tracking mechanism must also be able to collect data, carry-out computation and transmission of the relevant time-sensitive data, efficiently.

Moreover, data collection process for object tracking involves sensing environment around active nodes at certain intervals referred to as the sampling rate. Sampling rate can directly affect the precision of location data. Lower sampling rate can save energy by reducing sensing costs but can lead to data with low level of accuracy and may even miss the object completely [34, 39]. On the other hand, high level of sampling improves the tracking accuracy but can generate excessive amounts of data [40]. Traditionally, in target tracking schemes sampling rate is determined either for the next time of tracking or for the next several time steps of tracking. For both schemes, the sampling rate or the number of samples taken within a certain time period are fixed and known [39, 40]. However, it carries significant demerits when considered for object tracking, target loss can be exaggerated if the moving target changes speed or direction suddenly. Also fixed sampling rate for tracking an object moving at a very slow pace can collect excessive amounts of data which results in increased computational and transmission costs.

The data collected by each sensor node also has to be processed to extract the most relevant data. Timely computation of the most relevant data is an important aspect of a WSN for object tracking [41]. Any delays in computation can lead to time-sensitive data being rendered obsolete and hence consumption of resources for no gain from the network [42]. Traditionally, all data is transmitted to the base station to be processed but recent researches [10, 43–45] have shown that a distributed approach for an in-network data processing and aggregation reduces the computational times and discarding the irrelevant data close to its source can save the communication costs of the network.

In-network processing of the location data can lead to reduced delay in object localisation, one of the core functions of a WSN for object tracking. Through performing this core function periodically, the trajectory of the object over time can be tracked. Sensing nodes that detect the object send reports towards the base station. However, knowing just the location of the object or its immediate next location is not acceptable for requirements in the areas of security and surveillance [46]. A tracking mechanism should not only be able to compute the location of the object but should also be able to analyse the data and predict the destination of the object. Recent research in the fields of vehicular networks [2, 47] and traffic management systems [2, 47], object destinations and paths to any particular destination are predicted. however, that requires prior information about the object's movement. This prior information is not available for a system developed for object tracking within security, intrusion detection or surveillance domains. Use of sensors in modern cities around the world has increased exponentially over time and sensors, such as, cameras and acoustic sensors are deployed for surveillance and security. Within these modern cities security implications are also a reality and prior knowledge of a security threat is not always available, also, object tracking is done by human subjects from sensor feedback like CCTV feeds. Thus, there is a need to develop a new framework for accurate object tracking.

1.2 Aim and Objectives

The aim of this research project is to accurately track an object and predict its destination without using prior historic information in an energy efficient manner using wireless sensor networks.

FIGURE 1.1: Probable paths available to the moving enemy within the observed area[1]

Given a security scenario in Fig 1.1 where a potential hostile enemy has invaded from a point O and are planning to reach one of the five (5) destinations $(D1, D2, ..., D5)$. Enemy has more than one routes available to them towards any destination. As the enemy moves from the origin O to any of the five destinations, circular mini-clusters are formed within and across the larger static clusters. In order to capture the enemy and stop them from reaching their destination the proposed mechanism needs to predict their location based on speed and direction. Then basing on their movement through the observed area their eventual destination would be predicted. Within this security scenario, no prior information about the enemy is available as they may never have invaded this region before and even if they did invade before their motives about that invasion would have been completely different, making any prior information irrelevant to the current invasion. To provide adequate security any law enforcement organisation would require the movement information about the enemy, ranging from the likely destination of the enemy to reinforce security at that point or along the path the enemy would take to attempt to apprehend the enemy before it reaches its destination. The enemy, however, would try to evade capture and can move in a manner where they

[1]Map image source from Pirate Maps: http://www.yarrmaps.com/

might abruptly change direction or change the speed at which they move within the field which can lead to collection of inaccurate data or wrong destination predictions. In order to make sure that the most relevant and accurate data about the movement of enemy is collected, the network should provides the next state prediction of the enemy, meaning it provides the movement predictions in terms of time+1 as the enemy move through the observed area, while maintaining an acceptable level of data accuracy.

Based on the scenario above it can be extrapolated that the enemy's movement is driven by intelligence and hence the enemy makes decisions about their movements based on a predetermined destination. They move through an observed region while avoiding obstacles and keeping to the minimum distance they have to travel. In order to determine the destination of the enemy their motivation for moving towards a certain destination must be ascertained. Although it is assumed that there is no prior knowledge about the enemy available but prior knowledge of the terrain and potential destinations is known. Existing approaches where destination of the object is predicted such as in vehicular networks and traffic management, historic data about the movement patterns is used. However, this historic data or prior data about the objects movement is not available within the security scenario. Within the scope of this research this historic data is referred to as prior data.

While keeping this security scenario as a guide the following objectives have been researched through this project.

1. To accurately predict destination of object without any prior historic information about the object's movement.

2. To develop an efficient network architecture to reduce the number of nodes involved in object tracking, making sure that the least number of nodes are involved in the data collection process to maintaining energy efficiency.

3. To obtain the most relevant information from the network by reducing the amount of data collected to reduce the communication and computational costs while maintaining the required level of data accuracy.

1.3 Contributions

To address the objectives mentioned above, a novel framework for efficient object tracking has been proposed. The main contribution of this work include:

1. *A New Prediction-Based Algorithm for Destination-Estimation (PAD)*

   To predict the destination of the object (objective 1), we have developed a prediction-based mechanism for destination-estimation (PAD) based on Origin-Destination Estimation [47–49], which not only calculates destination of the object but also the trajectory that the object would follow, without any prior information about the object. Any set of origin and destination can contain multiple paths and to calculate the likely path the object is going to take, a set of environmental and physical attributes must be considered. The object's determination to take any particular path without the knowledge of any prior information about the object must be ascertained, which is almost never available in a security related or surveillance scenarios. In order to predict a path based on these attributes we have created a novel matrix of conditions that can be modified for any given scenario. As the object moves along those paths, more paths are added or discarded . Availability of these paths provides the possible geographic movement predictions that can then be used to calculate the possible location of the object at a future time which in itself is a significant aid for the purposes of security scenarios.

2. *A New Hierarchical Hybrid Clustering Mechanism (HHCM)*

   To achieve objective 2, a hierarchical hybrid clustering mechanism (HHCM) [50] has been proposed, where, nodes are placed in a multi-tier hierarchy forming dynamic mini-clusters within static clusters. The aim here is to reduce the number of nodes involved in the tracking process by keeping the largest number of nodes in a dormant or sleep mode and activating only the most relevant nodes for the purpose of tracking task. As the object moves within the observed field, mini-clusters are formed along the terrain where the object can move, all nodes within a mini-cluster are activated to observe the field and once the object moves out of the observed region of that mini-cluster, it is dissolved.

3. *Dynamic Sampling and Bayesian Filtering for Extraction of Most Relevant Data and Prediction of Next State Location*

   For objective 3, an adaptive dynamic sampling mechanism to reduce the amount of data collected has been proposed. In addition, a Bayesian filtering mechanism, in the form of Kalman filter, is used to extract the most relevant data and the prediction of next state for accuracy [50, 51]. In contrast to the existing approaches, where sampling is at a fixed rate, a dynamic sampling scheme has been devised, which dynamically increases or decreases the sampling rate till a balance is achieved where the observed data about the object movement is to an acceptable level of accuracy while keeping the sampling rate to a minimum. Furthermore, taking inspiration from Markov Decision Process (MDP) which provides a mechanism of determining the decision making process without any prior data, Kalman Filtering (KF) has been implemented. KF calculates the next state of the object while also determining the accuracy of the data collected. KF uses a series of tracking data observed over time by nodes within the mini-clusters and generates an estimates of time+1 location. KF not only calculates the next state of the object but also assigns weight to the data based on accuracy and based on that weight the most relevant data can then be transmitted to the higher level within the hierarchy of HHCM and less relevant data is discarded.

1.4 Thesis Structure

The thesis is organised as follows:

Chapter 2 contains a comprehensive literature review concerning all types of algorithms and protocols that have been proposed in recent research. The scope of the research has been defined and description of the limitations of those approaches have been discussed and how this research distinguishes itself the existing approaches. Current advances have been critically analysed and evaluated for their limitations.

Chapter 3 presents the description of the problem area within the scope of this research. The novel network architecture for object tracking, HHCM and its constitution has been discussed in detail. This chapter also explains the object tracking technique that has been employed to track the next state of the object, which provides the observed data quality that is essential to this research. Sampling mechanism has also been discussed which has been developed to further conserve energy.

Chapter 4 introduces the novel Prediction-based Algorithm for Destination-estimation (PAD). PAD mechanism is analysed and how it determines the eventual destination of the object as it moves through the observed field is discussed in detail. The destination estimation model and the path determination model of PAD has also been discussed. The novel matrix that has been developed to determine the importance of one destination over the other is also explained in detail. In this chapter, the recovery mechanism has also been presented, which is based on a combination of HHCM and PAD.

Chapter 5 includes the experimental evaluation. In this chapter, the experimental set-up and the experiments that have been performed to prove the validity of our research are discussed along with the results of the experiments and an explanation of the results.

Chapter 6 summarises the contributions of this work, discuss the limitations that warrant further research and provides directions for future areas of research that can benefit from this project.

Chapter 2

Literature Review

When tracking objects using WSNs the current approaches [18, 21, 22, 36, 52, 53] mostly try to resolve the challenges around energy-efficient sensor deployment, computing and processing energy consumption, communication energy consumption and sensing energy utilization. While other tracking mechanisms try to enhance the localization accuracy and tracking quality [24, 53, 54]. To better understand the problem of object tracking in WSNs and the proposed solutions in the literature, various literature review and survey articles have been published recently [29, 31, 32, 34, 55]. Some of these studies discussed the proposed algorithms for localizations; while other papers study the energy consumption enhancement of the tracking algorithms. In this chapter various protocols, sensing techniques and algorithms have been highlighted that have been proposed for target tracking using WSNs.

Before the approaches are discussed the criteria for evaluating those techniques has also been discussed. For that purpose the considerations that are vital for any object tracking mechanisms are highlighted.

2.1 Object Tracking Considerations

Design of object tracking protocols for WSNs is dependent on efficiency of a number of factors, the most relevant and widely discussed factors are presented in this section.

2.1.1 Energy

Whenever a node detects and object and reports that to other nodes it consumes energy. The lifetime of the network is dependent on the efficient management of the overhead communications. energy savings can be achieved by reducing the communication and sensing costs, however, that can result in reduced data accuracy. Hence, a tracking mechanism needs a robust mechanism to achieve the most accurate data while consuming the least amounts of energy.

2.1.2 Accuracy

Tracking accuracy can be defined at the difference between the actual location of the object and the the detected or predicted location of the object. Since object localisation is achieved through sensing the environment by a number of nodes it is desirable that the computational and communication load is kept to a minimum while collecting the most relevant data. This relationship would also effect the accuracy of data in relation to the loss of target [21, 56].

2.1.3 Scalability

Scalability can be defined as the capability of the object tracking sensor network to adapt to the increasing demands from the tracking mechanism and the the ability to deal with the increase in size of the network. Tracking protocols, inherently, require a system to be scalable as the demands of the tracking mechanism can vary depending on the environment and scenario. A scalable mechanism should be able to maintain energy efficiency and energy conservation even if the network increases in size or even if it is required to perform multiple tasks. Cluster-based protocols tend to be more scalable to network density variations as compared to other approaches [25, 26].

Review of existing approaches has exposed many drawbacks, specifically the ability of existing approaches to maintain energy efficiency while tracking an object [5, 8, 9, 13, 15–18]. Energy efficiency is of extreme important to any sensor network as individual nodes have limited resources and maintaining energy efficiency extends the life of the network. Apart from energy efficiency, the accuracy of detection and prediction mechanism also suffers from drawbacks [1, 8, 9, 18, 57] within existing approaches. Data accuracy is vital in security and intrusion detection mechanisms. Accurate data can be used to intercept the intruding enemy, however, less accurate data can lead to false location estimates and resource wastage [21, 56]. Another aspect of object tracking with sensor networks is the scalability of the network, its ability to engage more nodes to carry out the process or the ease with which a network can be extended [25, 26]. An efficient object tracking mechanism would, however, have to be energy efficient, while maintaining an acceptable level of data accuracy within a scalable network architecture.

In the following sections we discuss different object tracking protocols and algorithms and discuss their efficiency based on the considerations discussed above.

2.2 Protocols for Object Tracking Networks

Object tracking protocols that have been proposed by researchers can be classified based on the network architecture. Network architectures such as tree-based architecture, where nodes are organised into logical trees, cluster-based architectures where nodes are organised into clusters and self-organising architectures. Energy efficiency and information accuracy are the two most important issues to be considered in the design of any object tracking approach. Sensors provide useful information about objects within its sensing region, like its source, destination, movement path and time. Sensors and sink nodes maintain that information and provide the updated information about the tracked object when needed [10, 58]. Figure 2.1 shows the high level classification of different protocols

2.2.1 Tree-Based Protocols

Tree-based tracking protocols are based on tree network structure. Tree structure is established when a node detects an object and as a target is detected the tracking information of the target is sent back to the root node through the parent nodes. The

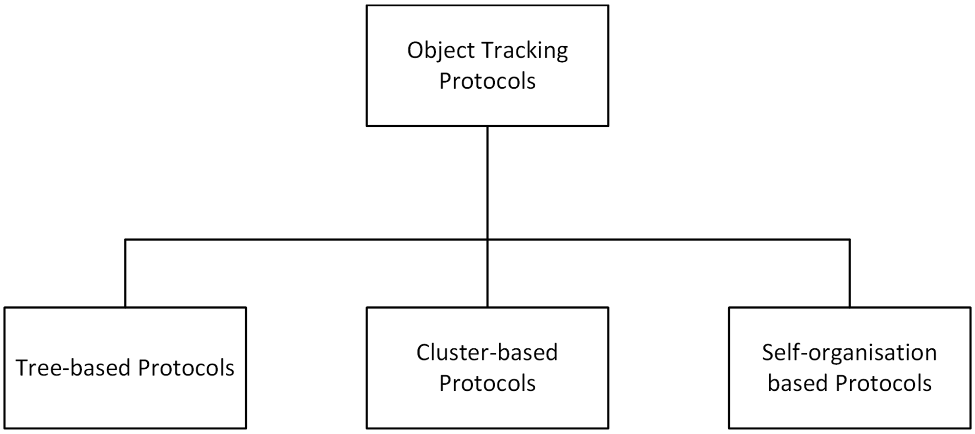

Figure 2.1: Classification of object tracking protocols

advantage of tree structure is that it creates a direct to root path without any loops in transmission. However that can lead to issues, such as, scalability, as adding more nodes to the tree can lead to excessive communication and set-up overheads.

A number of protocols built on tree-based architecture have been proposed. This literature review has focused mostly on the mechanisms that have focused on the research considerations mentioned in the previous section.

In [15] Scalable Tracking Using Networked Sensors (STUN) is proposed, which is a hierarchical structure for tracking a large number of objects. STUN is based on a tree structure, which is based on the premise that the target does not move in a uniform path in an observed environment [59, 60]. The underlying structure is a tree and the maintenance of object's data, such as its identification, is mostly done by the nodes near the edge of the tree. This is done through the detected set, where each node in the tree keeps the detected object's information that was detected by its child node. Important information, such as the speed and pattern of object's movement or the details of topology set-up are described in detail [16, 61, 62]. STUN does provide a high level of data accuracy, however, the energy requirements for keeping the network updated are excessive, which can lead to reduced network life-time.

With Dynamic Convoy Tree-Based Collaboration (DCTC) [1, 17], the network is set-up in a grid like topology and the nodes use Geographic Adaptive Fidelity (GAF) protocol for sleep scheduling. Nodes keep a direct communication link with only its neighbouring nodes and wake up periodically to observe the environment, however, the grid head stays awake at all times. All nodes in the grid have the location information of all other nodes in the grid. DCTC relies heavily on the mobility prediction algorithm to know the distance to the centre of the event and is required to keep that knowledge as a necessity [63, 64]. Also, the network maintenance is dependent on costly communications that can severely cripple the network, especially when the detected object is moving at a high speed. Root node is also responsible for adding or removing nodes from the grid which requires excessive overhead communication from nodes near the edge to the root.

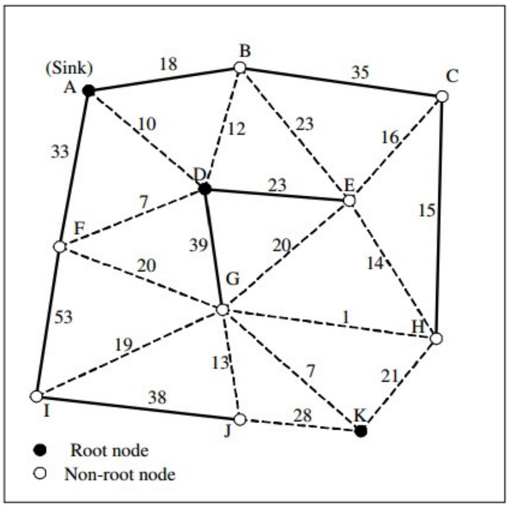

FIGURE 2.2: Representation of Deviation-Avoidance Tree (DAT)[1]

With Drain and Balance Tree Structure (DAB) a mechanism is proposed where network overheads are greatly reduced by reducing the number of update and query messages, while placing all information at the sink hence reducing the need to flood the network for information [65]. All nodes then transmit the updated information to the sink. But it has some drawbacks. DAB structure is logical and does not depend on the actual structure of the sensor network. Also communication cost can be excessive as the nodes near the edge of the tree depend on a multihop communication model. DAB also does not account for the communications costs during the querying phase. The proposed methods to overcome these issues are Deviation-Avoidance Tree (DAT) [18, 66]. DAT proposes that each node is a tree on its own and multiple trees can be joined together to formt he extended structure. DAT conserves energy in two stages, by reducing the update costs and then by reducing the query cost. Although, this approach has significant overhead communication reductions, however, as more and more sub trees are combined the structure becomes more complex and cost of transmissions to the sink increases, requiring several hops to transmit the data.

On the other hand, Optimised Communication and Organisation (OCO) [5, 67] proposes a low computation overheads on sensor nodes while target objects by providing self-organisation and routing capabilities. Processes within OCO can be explained as four distinct stages, these are, localisation, computation, tracking and maintenance. During the tracking stage OCO uses two different types of sensors, one that can detect the object and wake up the neighbouring nodes when its about to lose the object and second that can only detect the object. The maintenance phase is only activated when the network needs to be reorganised, mainly due to node failure, damage or movement of the nodes. The biggest demerit of the scheme is that most computation is done at the root node which leads to excessive communication costs, on top of the excessive communication that is required for the control messages through all four stages.

2.2.2 Cluster-Based Protocols

Cluster-based methods organise sensor nodes into clusters to support collaborative data processing. A cluster consists of cluster head (CH) and member sensor nodes. Figure 2.3 shows the classification of cluster-based protocols.

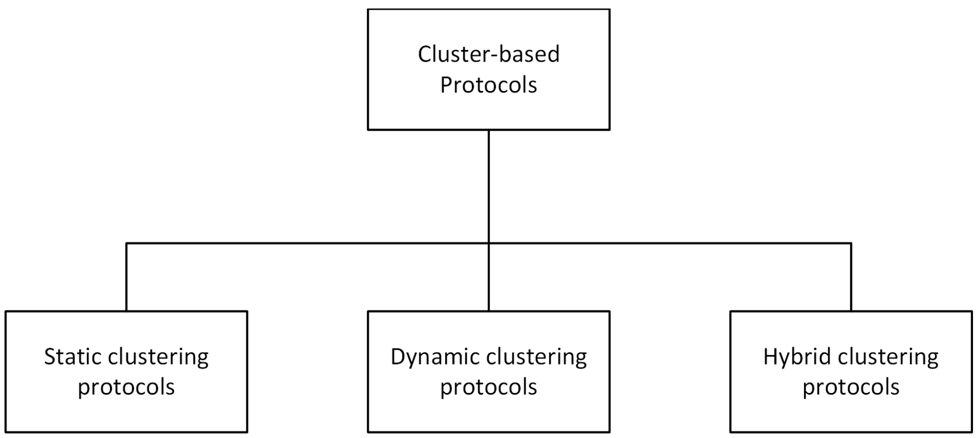

Figure 2.3: Classification of cluster-based Protocols

Upon target detection any node can declare itself as a cluster head. Depending on the approach used for target tracking, no particular election mechanism is required for cluster head selection, which limits the communication costs. If more than one node detects an object, multiple volunteer nodes may exist, hence, a decentralized mechanism is required to make sure that only one cluster head is involved in object tracking process at a time to avoid redundancy. Figure 2.4 shows the how the basic structure of the network would look like.

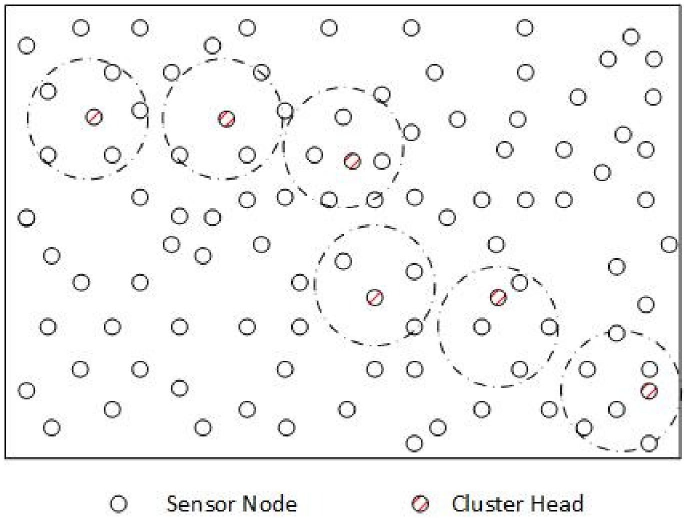

Figure 2.4: Dynamic Cluster representation highlighting the formation of clusters

If the structure is already established, then with Distributed Predictive Tracking (DPT) [8, 9, 68], One the target is acquired a four stage sensing, prediction, communicating and then then sensing again is performed. The algorithm is made of four distinct mechanisms. TD or target descriptor that contains the target's identity, its location and future prediction. Rest are Sensor selection mechanism, recovery mechanism and energy conservation mechanism. The cluster head uses the TD to activate and track the object, while the recovery mechanism is used to rediscover the object is the object is lost or the node tracking the object fails. Target's future location prediction is then used to activate the relevant nodes for tracking purposes. Although DPT is a prediction-based protocol for object tracking, its experiments show that the energy savings and prediction probabilities are not very encouraging [69]. Also, high overhead costs are involved when the extensive communication is required for the cluster head to know the location of all its nodes and the TD is sent back to the sink for every object being observed.

A CPA or Closest Point of Approach, is used to form the Dynamic Space-Time Clustering (DSTC) [14, 70]. Clusters are formed by sending out a CPA pulse and all nodes listen to it, first node to receive four CPAs declares itself a cluster head as it becomes the receiver of the highest intensity CPA. It then estimated the location of the source by using the weighted mean of the CPA and the intensity of the CPA ans the weight. Reliable detection of CPA dictates the area within which the the CPA broadcast is acknowledged and time stamp is the same as the node traversal time. This is a very efficient mechanism, however, it fails when the node density becomes too low [68]. Also the target estimates at maximum area of CPA can be very unreliable and hence, would require a smaller detection area or increased number of nodes. As more nodes become part of the cluster communication cost increase exponentially with any assurance of accuracy gains.

To reduce the number of nodes required to carry-out the tracking, Prediction-based Energy Saving (PES) scheme has been proposed in [10, 71]. Based on clustering, PES consists of three mechanisms, prediction mechanism, wake-up mechanism and recovery mechanism. PES reduces the sampling rate and also tries to reduce the overhead for object rediscovery. Prediction model activates the future nodes using prediction probability. Then the wake-up mechanism decides which nodes should be activated and when based on energy and performance. While, recovery is only initiated when the object's track is lost. PES aim to keep the nodes in sleep mode as much as possible even if they are detecting the object. The node tracking the object tries to predict the future location of the object and then activates the nodes in that region. As with any object tracking system a recovery mechanism is available as the object loss is always possible and the possibility of that cannot be reduced to zero.The recovery mechanism tried to recover the object locally by only activating the neighbouring nodes and only in case of failure to recover, the whole network id activated. Activating a flooding mechanism for recovery can cause excessive communication, computation and sensing costs.

Redundant data produced by a sensor network can lead to excessive energy consumption and delays and to reduce this redundant data two protocols have been proposed, RARE-Area (Reduced Area Reporting) [15, 72] and RARE-Node (Reduction of Active Node Redundancy) [15]. Both the approaches are based on clustering mechanism , where, RARE-Area tries to reduce the nodes available for target tracking and RARE-Node further tries to reduce the amount of data collected by the network itself. RARE-Area reduces the participating nodes in the tracking activity by measuring the weight of the data they generate, only nodes with more accurate data are allowed to track the object. RARE-Node determines the quality of data collected [71] base on the relevance of data. Nodes check their immediate neighbourhood for other nodes and if none are found then the data is considered to be important. If other nodes exist then the node nearest to the object and with ample communication resources transmits its data and rest of the nodes discard the data they collected.

Redundant data is not generated only while the object is being tracked but also during the discovery process. The Continuous Object Detection and Tracking mechanism (CODA) [11, 73] enables each sensor node to detect and track the moving boundaries of objects in the sensing field. At deployment stage static clusters are formed which remain active while they have energy resources. One an object is detected within a static cluster, that information is send to the cluster head, which then forms the a dynamic cluster around the boundary of the continuous object being detected, this information is also shared with the sink. CODA has two advantages [74], it reduces the overhead costs by forming the dynamic clusters at the static cluster head level which reduces the redundant communication costs of the local nodes and, secondly, also the dynamic clusters do not need to elect a cluster head which furhter reduces the communication overheads.

Uneven load distribution within a sensor network can lead to certain nodes depleting their energy resources quicker than others and can create holes within the sensed region. Hybrid Cluster-based Target Tracking (HCTT) is specifically designed to even load distribution. HCTT [13, 67, 75], consists of a hybrid clustering mechanism where all nodes are part of static clusters and are aware of their location in terms of geography. Static cluster is responsible for object tracking until the tracked object reaches the boundary of the static cluster, This is when a dynamic cluster is formed among the boundary nodes of the existing static cluster and that of the static cluster where the object is moving towards. Once the dynamic cluster is formed it is responsible for handing off the tracking of the object from one static cluster head to the other. Nodes determine if they are a boundary node by asking their neighbours about their static cluster, once a node establishes that it is the boundary node it forms the dynamic cluster with boundary nodes of the other cluster. Object tracking is carried out by a single node till the object reaches the boundary and that is when the dynamic cluster takes over the tracking duties. Once the object moves into the next static cluster a hand-off mechanism is triggered which hand over the tracking duties to the next static cluster, at which point the dynamic cluster is dissolved. This handing-off mechanism makes it difficult tot lose the object tracking when an object moves from one static cluster to another.

2.2.3 Self-organisation Based Protocols

As discussed earlier, self-organising sensor networks protocols for object tracking are generally lightweight and are mostly capable of tracking single objects.

In Low Energy Self-Organising Protocol (LESOP), the first node to detect the object declares itself as the leader node. At this stage the node with the second best estimate of the object is declared as the surrogate leader [18, 76]. The neighbouring nodes then observe the object and send the tracking data to the surrogate leader. The number of nodes that track the object is determined by the required level of accuracy. More nodes are added if the minimum threshold of accuracy gain has not been achieved but is achievable. The surrogate then with the help of the leader builds objects movement trajectory. As the minimum time already defined for the current set-up expires the surrogate becomes the leader and the whole process is repeated [18]. LESOP combines the application-layer with the MAC and physical layer to form a light weight mechanism, but the communication is limited to single hop due to the lack of a transport layer definition. Also, the process of leader hand-over is core to the system but it does not bring any benifit to the tracking activity which is the primery responsibility of the system.

Information-driven approach is used by Information-Driven Sensor Querying (IDSQ) to track targets and direct queries [77]. Within IDSQ, a leader is elected at the set-up of the network but it remains dormant until activated by the application. Once a leader is activated it asks the member nodes to sample the environment and report to the leader. The state of the object is recorded by the leader and any addition to that information is made by the leader, this state update is made only after a predefined threshold of accuracy of data is achieved [1]. One downside to IDSQ is that only one node detects the object and no collaboration takes place to improve the detection accuracy. Thus, any inaccurate data introduced in the initial stages of tracking has a huge impact on the remaining tracking process.

More recently, Distributed Event Localisation and Tracking Algorithm (DELTA) method [57, 78] has been proposed. It form groups within the detecting nodes dynamically. The nodes are arranged in a grid with local information about the node's location is already known. The leader announces its existence with a HEARTBEAT broadcast that is transmitted to two hops of the leader. DELTA is developed to track a single target and no multi-target mechanism has been defined. Also node recovery mechanism is non-existent in case of target loss.

The following Table 2.1 shows how the protocols discussed above provide energy saving and tracking accuracy and if the protocols provide predictive information about the future position of the tracked object.

TABLE 2.1: Comparison of Protocols against Accuracy and Energy Efficiency

	Protocols	Accuracy	Energy Efficiency	Tracking Algorithm
Tree-based protocols	STUN[15, 16]	high	low	non prediction-based
	DCTC[1, 17]	high	low	non prediction-based
	DAT[18]	low	low	non prediction-based
	OCO[5]	high		non prediction-based
Cluster-based protocols	DPT[8, 9]	low	low	prediction-based
	RARE[15]	high	high	
	CODA[11]	high	high	non prediction-based
	DSTC[14]	low	low	
	HCTT[13, 75]	high	low	prediction-based
	PES[10]	low	high	prediction-based
Self-organisation based protocols	LESOP[18]			prediction-based
	IDSQ[1]	low		
	DELTA[57]	low		

2.3 Object Tracking Algorithms

In this section, some of the latest WSN target tracking algorithms are discussed. Many different algorithms have been proposed over the years and they can be easily organised into two broad categories, Bayesian Framework based tracking algorithms and Graph-based tracking algorithms.

2.3.1 Bayesian Framework Based Tracking Algorithm

The Multiple Hypothesis Tracking (MHT) is one of the most fundamental tracking algorithms in sensor networks [56]. It is a Bayesian filtering mechanism which tries to calculate every possible probability about the object movement. However, as the object spends more time within the observed region, the probabilities become more complex and can lead to excessive computational load.

The Joint Probabilistic Data Association (JPDA) algorithm is another Bayesian filter based algorithm [79]. In the JPDA algorithm, the possible target actions are again enumerated for each time frame, but an action will be selected before making any further hypotheses, Which helps to limit the number of hypotheses that are generated.

The Particle filter is another way to reduce the cost of Bayesian filter based algorithms. Particle filters collect as much data as possible from a wide range of sensors and compress it into a single distribution. As more information is collected form the network, the amount of samples considered remain the same and hence reduce the data explosion over prolonged tracking. This data containment leads to reduced communication costs and transmission overheads. In [80, 81] extensive accuracy analysis is discussed. In [82] a distributed particle filtering mechanism is discussed where the network is divided into cliques and particle filtering takes place in all cliques at the same time.The mechanism presented in [83] aims to extend the mechanism in [82] by further compressing the data from multiple cliques.

When multiple targets present, data association problem is used to associate the measurements to the corresponding target trajectory. Typical Bayesian filter approaches require a known number of targets, and the initial state of the target in order to start the association iteration. Markov Chain Monte Carlo Data Association (MCMCDA) algorithms solved this problem. Framework of MCMCDA algorithm can be found in [84], where the target trajectories (tracks) are modeled as Markov Chain, and random sampling is used rather than iterative Bayesian filter, hence relaxed the dependency of the prior information.

2.3.2 Graph-based Tracking Algorithms

There are not many graph-based tracking algorithms developed, and most of these graph-based tracking approaches nowadays are focused on topology to physical location manipulation. In [85], for example, the rooms and the hallways are modeled as nodes in the graph, the doors and passages are modeled as links in the graph, and the target is assumed to transit from node to node. The actual tracking algorithm is still under Bayesian framework, and the graph part of the algorithm is to maintain the sensors (RFID readers) to be linked in such a topology.

Another graph-based tracking mechanism is presented in [4]. This tracking method is another implementation of the Bayesian framework. In this algorithm, binary sensor data is used, and the topology information is only used to provide neighbouring information and information routing paths [86, 87]. All mechanism discussed here require extensive knowledge of the network set-up and also the location of the nodes within the network,

2.3.3 Graph-based Event Boundary Detection Algorithms

There are some algorithms developed for detecting event boundary in sensor networks that are also graph-based. Differ from a single point target, an event is usually an area of interests, and tracking of such area is a boundary detection process. In [88] and [89], a minimum level of accuracy is the requirement for a node to decide to track on object. The algorithm proposed in [43] arranges the nodes in clusters based on the nearest neighbour and the whole event area has to be covered by the cluster. Nodes on the boundary of the event are dismissed as false alarms. Another mechanism presented in

[90] divides the whole sensing region into smaller chunks, till these sub-regions either completely detect the event or not at all.

2.3.4 Markov Network

Although Markov Network based tracking algorithms are graph-based, they did not fully utilize the data fusion, data correlation and clustering features offered by graphical model. In order to fully utilize inter-sensor correlation, cliques and junction tree algorithms should be considered.

In a graphical model, a clique is a group of fully inter-connected nodes. it is usually used to subdivide the graph into subgraphs. The cliques are independent to each other given the shared nodes, and the statistical property of the clique is described in terms of clique potentials, which is usually the joint probability of all the enclosed node.

Clique tree is a tree that all of the nodes are cliques. It provides an overview of the graph after the graph is been subdivided using cliques [91, 91]. Junction tree is a special case of clique tree that obeys the running intersection property [91, 92]. The running intersection property basic dictate the locality property in the clique tree, it requires that for any pair of cliques U and V, the cliques between U and V must contain U \V. While the cliques provide clustering and joint probability of a small group of nodes, the junction tree provide a fast route to propagate information through the network, and it is also statistically sufficient to represent the original graph.

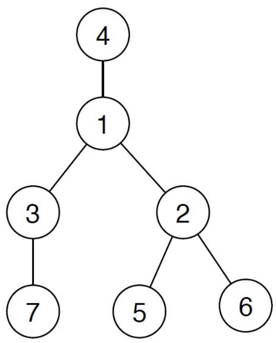

FIGURE 2.5: Representation of junction tree

Markov network is a special type of graph, where the edges correspond to direct probabilistic interaction between neighboring nodes, and is not to be intervened by other nodes, e.g. the Markov property [93, 94]. It is commonly used for segmentation in image processing [95–98]. For tracking and detection applications in sensor networks, the target is usually concerned by a small set of local sensors, hence Markov network is especially suitable for such situation. Combine Markov network, junction tree and clique concept, a randomly deployed sensor network can be converted into a more structured form, and greatly simplify the collective process and data propagation.

2.4 Object Tracking Sampling Techniques

Sampling methods can be classified based on their activity structures as naive sampling where all nodes remain active and perform tracking tasks. Continuous sampling, where one node continuously monitors the object and scheduled sampling, where certain nodes sample the object based on a predefined mechanism. Energy efficiency and data accuracy are the prime considerations when a sampling mechanism is devised. Sensors provide useful information about the observed region but too much data an lead excessive energy consumption and too less data can lead to inaccurate information from the system.

Naive sampling is known as the all-in mechanism where every possible node is activated and to object tracking. As an object moved into the sensed region it is detected and continuously sampled and all data is then communicated to the root of the sink [29, 31, 34]. It is, however, a naive system and data generated through this system can be extensive which increases the energy requirements for communication and computation exponentially. Naive method is not suitable for modern requirements where sensors have limited energy reserves and network longevity is a primary requirement.

One way to reduce the cost sampling is to employ continuous sampling where only a single node samples the environment till the tracked object leaves its detection zone in a continuous and consistent manner. As the object transitions to the detection zone of another node the responsibility of tracking is handed over to the next node and the first node drops to a dormant state [39]. This method provides energy conservation, however, data accuracy and tracking loss are increased. This method is not fault tolerant and hence, modern applications of object tracking cannot possibly use this method.

To mitigate the demerits of naive and continuous sampling, scheduled sampling methods have been proposed, where a cluster of tree of nodes collectively tracks the object, where as rest of the nodes remain in inactive state. Scheduled sampling can be classified into prediction-based and non-prediction-based. Non-Prediction-Based sampling refers to methods where the sensor nodes alternate between active and inactive states based on a predefined mechanism to conserve energy [24]. However, this method can have certain drawbacks like the sensors could miss the event that occurred while the sensor was not sampling the environment. To reduce the chances of missing an even prediction-based sampling has been introduced [18, 22]. Prediction-based sampling activated the nodes for object tracking based on a predictive algorithm and the nodes stay active under this method's instructions. With predictive activation only a small number of nodes at a particular region become active while all other nodes remain dormant. Once the object moves out of the region the nodes in the next predicted region are activated [18, 22, 24].

2.5 Discussion

Based on the analysis of this extensive background research, it is ascertained that the tracking mechanisms share some inherent characteristics:

1. An efficient mechanism limits the number of nodes involved in object tracking and sample the environment selectively for energy efficiency.
2. All tracking mechanisms strive to eliminate the redundant and inaccurate data to reduce the communication overheads.

Based on the above findings it has been learned that object tracking mechanism should be able to not only efficiently detect the object but also be able to reduce the energy consumption by keeping most nodes in sleep mode to conserve energy. An efficient protocol should also be able to reduce the amount of data collected by reducing the time nodes spend in sampling the environment while maintaining a high rate of tracking accuracy.

With the deficiencies of the existing approaches highlighted within this chapter, this research is directed towards developing an object tracking mechanism which should be able to track an object within a certain threshold of acceptable accuracy and carry out the tracking process while maintaining a high energy efficiency and ability to manipulate the number of nodes involved in the tracking process keeping the network scalable. While object tracking has seen significant research in the last few decades there exists a gap within the field of security management, although the object can be tracked to an acceptable level of accuracy the destination estimation of the intruding object or an enemy has not been the focus of research. Review of existing work has thus highlighted the research gap that would also be the focus of this research.

Keeping this research direction within focus a novel object tracking mechanism has been proposed that maintains a low number of active nodes at the most optimum sampling rate, while maintaining high tracking data accuracy. This approach is based on a Hierarchical Hybrid Clustering Mechanism (HHCM), an adaptive sampling mechanism that dynamically adjusts the sampling rate to maintain data accuracy and a Prediction-based Algorithm for Destination-estimation (PAD) that predicts the final destination of the object without any prior information.

Chapter 3

The Proposed Hierarchical Hybrid Clustering Mechanism (HHCM) for Object Tracking

The goal of this research is not only to minimise energy consumption and accurately predict the object's future location but also aims to predict the destination of the object. The factors affecting target tracking and destination estimation include the amount of data collected and the number of nodes active at any given time. Excessive data collection leads to higher processing and transmission costs in terms of energy and high number of active nodes lead to excessive data collection and sensing energy costs. In this chapter, a hierarchical hybrid clustering-based approach for object tracking is presented, While a destination estimation mechanism is presented in Chapter 4. The proposed mechanism leads to reduced data collection while obtaining the most relevant data and prediction of next state of the object as it moves within the observed field.

3.1 Problem Description

Sensor nodes inherently have limited energy supplies and these energy supplies are not only required for sensing the environment but also to compute the data collected and transmission of that data. So as a network the energy consumption increases if more nodes are in active state or if they are collecting excessive amounts of data through sensing operation.

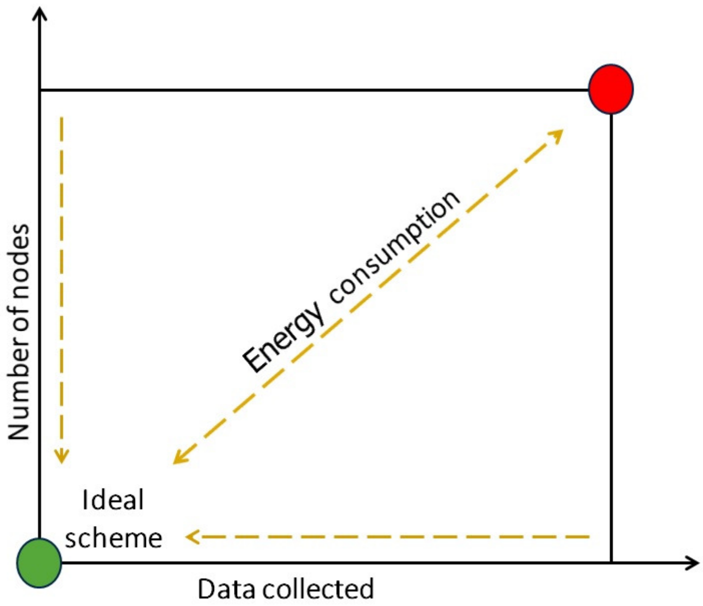

Figure 3.1: Relationship between energy consumption, number of nodes and data collected

Figure 3.1 shows that the ideal object tracking mechanism would be at the green area where least number of nodes are used to collect the least amount of data and yet be able to track the object accurately. Although, that ideal state is very difficult to achieve and several mechanisms discussed earlier have tried to get as close to that green area as possible.

Referring to the security scenario explained in Chapter 1 Section 1.2, the ideal tracking situation would be where enemy moving within the observed region can be tracked while using the least number of nodes at any given time and least amount of data is collected, all the while maintaining a high level of data accuracy.

In this chapter HHCM and a computation mechanism that filters out any irrelevant data has been discussed. Based on the security scenario, the following network architecture has been proposed.

3.2 The Proposed Hierarchical Hybrid Clustering Mechanism (HHCM)

A hybrid model of clustering can be explained as a technique to create and dissolve clusters that contain a mix of both static and dynamic clusters. Sensor networks are organised into clusters to spread the responsibility of carry-out any task among nodes to reduce load from any particular node with limited resources. Traditional hybrid clustering mechanisms [43, 75, 79, 99] have mostly tried to solve the problem of inter-cluster communication of static clusters by forming dynamic clusters from boundary nodes of multiple static clusters. Hence, the focus of hybrid clustering has been to reduce redundant communication costs associated with static clustering. Static clusters can contain a large number of nodes, which, in turn leads to collection of a huge amount of data. Therefore, a Hierarchical Hybrid Clustering Mechanism (HHCM) for object tracking has been proposed. Within HHCM, dynamic clustering is used to reduce the number nodes directly responsible for object tracking. Figure 3.2 shows the hierarchy of the nodes.

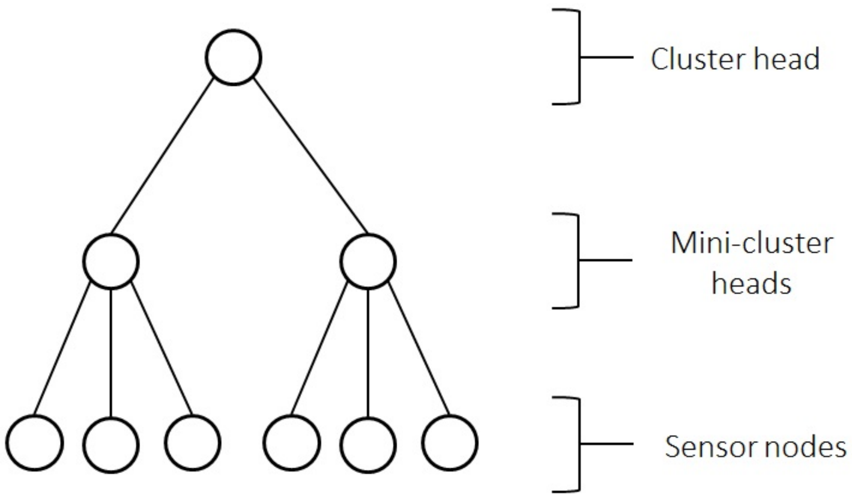

FIGURE 3.2: Network hierarchy of HHCM

HHCM's dynamic mini-clusters are formed within static clusters and across static clusters along the predicted future location of the object that is being tracked and a mini-cluster head then collect the information from its neighbouring nodes. HHCM is split across triple level hierarchy, where ordinary nodes form part of the dynamic mini-cluster which are managed through a mini-cluster head. This mini-cluster head itself reports to the cluster head of the larger static cluster. Mini-clusters perform a dual role in HHCM, not only do mini-clusters help to reduce the number of nodes made active at any given time to perform tracking which reduces the data collection and processing costs in terms of energy but also carry out part of the computation to reduce amount of data that is transmitted by filtering out the redundant data. This data is then sent to the static cluster head which not only receives the most useful data from the mini-cluster but also the most useful data, hence, reducing the load from the cluster head.

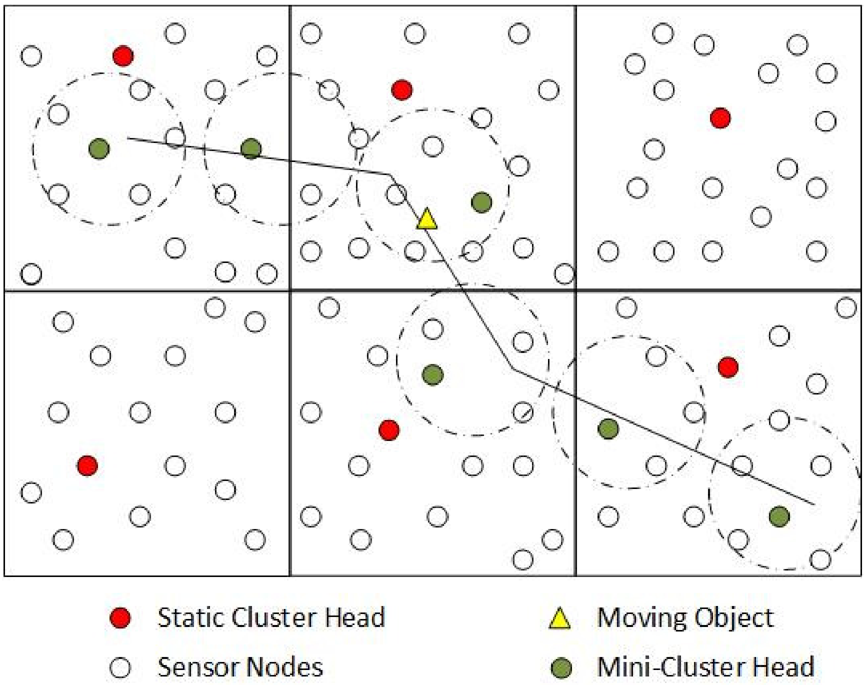

FIGURE 3.3: Network architecture of HHCM

Mini-clusters are formed based on the information received form Prediction-base Algorithm for Destination-estimation (Discussed in Chapter 4). A node is selected as a mini-cluster head and asked to form a mini-cluster by the cluster head based on the predicted path of the object being tracked. Mini-cluster head then forms a mini-cluster of its one hop neighbours, these participating nodes then collect the tracking information and pass that information to the mini-cluster head. Figure 3.3 shows the pictorial representation of cluster formation. The mini-cluster head collects information from its member nodes and then performs predictive computation and pass that information on to the cluster head. Mini-cluster is dissolved as soon as the future predictions suggest that the object being tracked has moved out of its tracking range.

3.3 Efficient Data Collection Mechanism

The purpose of any object tracking sensor network is to track the object in the most efficient manner, for this purpose the network needs to collect as little as possible amount of data to reduce the energy costs that are incurred during data collection, processing and transmission. To achieve that goal a robust network architecture is required which reduces the number of nodes involved in object tracking, an efficient prediction mechanism that can predict the next location of the object to activate the most appropriate nodes and an effective data sampling mechanism is required which only collects data at the most appropriate intervals to maintain the quality of tracking while avoiding redundant or unnecessary data.

While developing this tracking model, Markov Decision Process (MDP) has been used as inspiration for the predictive process to anticipate the future position of the target. MDP believes that at any time the process is in some state s, and the decision maker may choose any action that is available in state s. The process responds at the next time step by randomly moving into a new state s', and giving the decision maker a corresponding reward r. The probability that the process moves into its new state s' is influenced by the action a. Thus, the next state s' depends on the current state s and the decision maker's action a. But given s and a, it is conditionally independent of all previous states and actions; in other words, the state transitions of an MDP depends on the current state and not on any of the previous states.

MDP can be carried out with the use of Kalman Filter. The Kalman Filter is an algorithm that uses a series of measurements observed over time, containing random variations and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. The Kalman Filter works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state. Based on the principles of MDP and Kalman Filtering, HHCM mechanism has been proposed.

3.3.1 The HHCM Model

Within the proposed mechanism of HHCM in this thesis, we analyse the procedure explained in Figure 3.4, states are observed by the nodes and that data is transferred over to the mini-cluster head, which then uses the Kalman filter to predict the future state using the observed data and assign weight to that information. Based on the weight of the information it is then decided that the higher weighted information is sent to the cluster head and low weighted information is then discarded to reduce the communication costs.

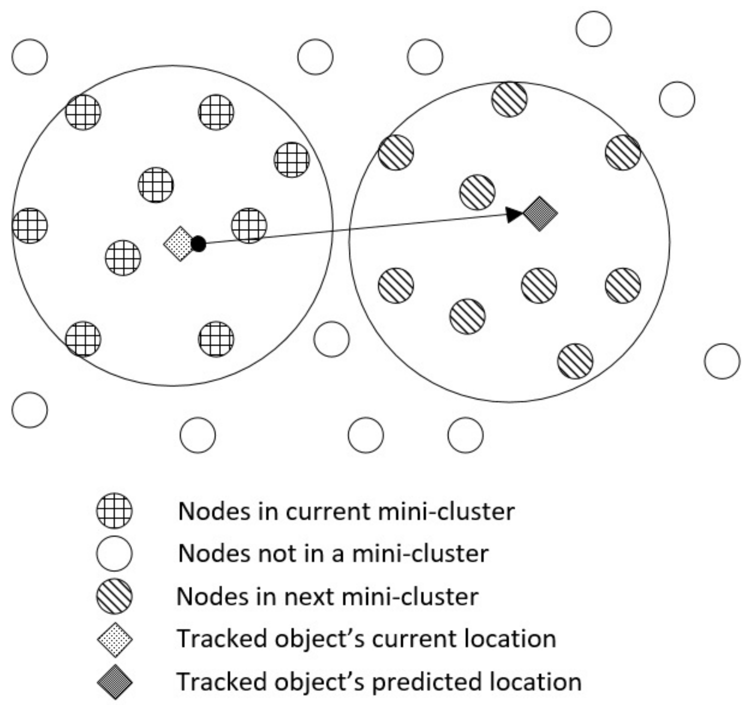

Figure 3.4: Mini-cluster formation

In order to carry-out that process of tracking the object Kalman Filter is employed to determine the time+1 prediction and to also predict the movement even if the object moves into a zone with no sensors or unreliably low number of sensors.

It is assumed that every object being tracked would move about in the observed field with a purpose hence it would move along a certain direction and at a certain speed. This information once collected by the sensor network is then used to predict the future location of the object and new mini-clusters are formed at that predicted location. We can explain the stages of our process as follows:

1. Object is observed within the sensor field at an entry point.
2. A mini-cluster is formed by the first node that observed the object by inviting its one-hop neighbours.
3. These mini-clusters would then observe the field and transfer the observation information to the mini-cluster head.
4. Mini-cluster heads filter and aggregate the data and pass the most relevant data to the cluster head.
5. Cluster head would receive information from multiple mini-cluster heads. It would then filter and aggregate that data before passing in on to the base station.

Based on this mechanism the next state of the object is calculated in combination with HHCM as demonstrated in Figure 3.5.

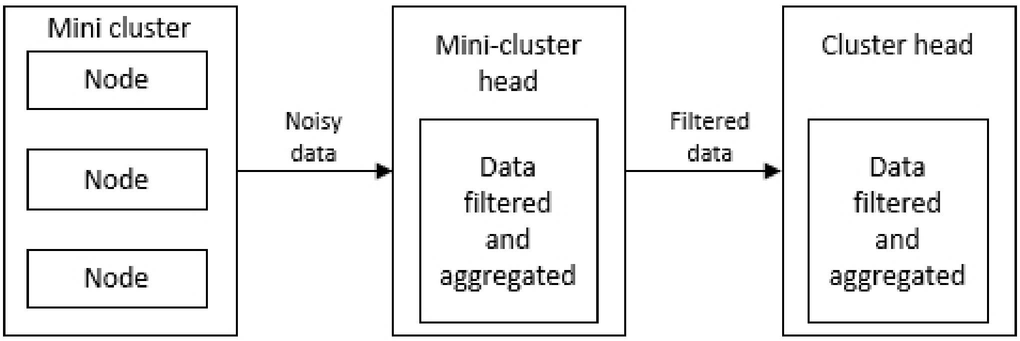

FIGURE 3.5: HHCM data transfer sequence

Keeping the stages of tracking mentioned above in mind, the Algorithm 1 can be visualised as:

Algorithm 1: Mini-cluster formation mechanism for next state prediction

Object observed;
if *closest node* **then**
    form mini-cluster;
end
Observe field;
Send object location to mini-cluster head;
if *mini-cluster head* **then**
    aggregate data;
end
return Location data;

3.3.2 Next State Prediction

For next state prediction a mechanism in combination with a HHCM has been proposed. The aim of the next state prediction is to estimate the moving target's position x_k at time k, from the measurement history up to time k within a field monitored by a sensor network in order to activate the relevant nodes and collect the most relevant data. Assuming that all nodes know their own position, z_k is a vector that contains the measurements of the mini-cluster head and its neighbours at time instant k. Kalman Filter has been employed to compute the corresponding Bayesian estimates and target prediction. It approximates the belief by a set of particle streams or sensor readings x_k^m, m=1, 2, **.....**, M, and their weights W_k^m, which are normalized such that $\sum_{m=1}^{M} W_k^{(m)} = 1$.

Particles structure is:

$$x_k^{(m)} = [x_{1,k}^{(m)}, x_{2,k}^{(m)}, \ldots\ldots\ldots, x_{n,k}^{(m)}] \tag{3.1}$$

1. Initialization: States x_k^m, m=1, 2, ..., M, are initially drawn from a prior distribution, and the weights are set to 1/M.

2. New state generation: A new set of states, x_k^m, m=1, 2, ..., M, is computed according to the posterior distribution

$$p(x_k|x_{k-1}) \tag{3.2}$$

$$x_k^{(m)} = \int (x_{k-1}^{(m)}) + n_k \tag{3.3}$$

   where n_k is a zero-mean Gaussian noise component with variance, which injects the randomness of the states observed by multiple sensors keeping the data true to the observations.

3. Weights update: Weights are updated by means of the likelihood, more relevant information is awarded higher weight

$$W_k^{(m)} = W_{k-1}^{(m)} p(z_k|x_k^{(m)}) \tag{3.4}$$

4. Position estimate: Based on the information received the target position is estimated as,

$$x_k \approx \sum_{m=1}^{M} W_k^{(m)} x_k^{(m)} \tag{3.5}$$

   Note that weights are normalized to sum up to 1.

5. Re-sampling: It takes samples with replacement from the set

$$(x_k^{(m)})_{m=1}^{M} \tag{3.6}$$

   Where the reason for taking sample m is $W_k^{(m)}$. Hence, states with the largest weights ($W_k^{(m)}$) are retained and those with the smallest ones are removed.

6. If the object is still within the observed field, reset time to latest and repeat the process from step 2.

While reducing the number of nodes involved in tracking process significantly reduces the data collected, however, a significant number of nodes must still be involved to increase data accuracy. In order to make sure that the nodes involved in object tracking do not collect excessive data a robust sampling mechanism is required so that nodes can be placed in a sleep mode to conserve energy while still maintaining a data collection rate that can be used to accurately detect and predict the object behavior. This thesis presents the following sampling mechanism, which dynamically adjusts the sampling rate to collect the least amount of data.

3.3.3 The Dynamic Sampling Mechanism

As the tracked object moves through the mini-clusters, it is important that the object's accurate location is observed. However, the limited energy reserves of a sensor node necessitate that the object is not observed in a naive continuous manner. There can be certain constraints when observing the object as it moves through the mini-cluster, like, level of accuracy needed, speed of the object, direction of the object and more. The objective is to meet the constraints by minimizing the average sampling rate. Each node must adjust its sampling rate to observe the object. Traditionally centralised optimisation is used to adjust the sampling rate, however, that requires an excessive number of control transmissions which in itself can lead to energy depletion in a sensor node. The alternative is a distributed approach where the sampling rates are adjusted locally while making sure that it meets the requirements of the constraints.

After carrying out extensive simulations it is determined that the sensing rate should be determined by the required accuracy of data in comparison with the least number of sensing acts performed. Sampling rate is a trade-off between data accuracy and number of sensing sessions performed. Data accuracy can be examined by calculating the average variation from the actual location of the object. In order to meet the accuracy of data piece wise fitness function has been integrated with HHCM.

Lets assume that Q_a is the observed accuracy of the sensor data and if it is within the required range of accuracy R_a then sampling rate satisfies data accuracy.

$$F = \begin{cases} \sum_1^N (S_i)^2 & : Q_a = R_a \\ S_i++ & : Q_a < R_a \\ S_i-- & : Q_a > R_a \end{cases} \tag{3.7}$$

Algorithm 2: Sampling Mechanism

```
if Q_a = R_a then
    Maintain sampling rate S_i;
end
if Q_a < R_a then
    Increase sampling rate S_i++;
else
    Reduce sampling rate S_i--;
end
```

However, as in Algorithm 2, if Q_a is lower than R_a then a sampling rate is increased, however, if the Q_a is higher than R_a then the sampling rate is decreased, where S_i is the sampling rate per period of time for the mini-cluster.

This sampling mechanism dynamically increases and decreases the sampling rate of any given mini-cluster based on the requirements, such as, the level to data accuracy required, or the amount of energy available to name a few.

3.4 Discussion

In this chapter, a novel tracking mechanism based on HHCM and a sampling mechanism to decrease the sensing energy consumption of the network during object tracking have been proposed. This mechanism, takes advantage of a robust network architecture combined with an aggregation mechanism that not only proposes to reduce the energy consumption of the WSN but also collects the most relevant data about the objects movement through the sensed environment. The object tracking mechanism also predicts the next location of the target and wakes up the sensor nodes at the next predicted location to form the next mini-cluster and to detect the target. it has been argued that this reduction of the energy consumption provides more alive sensor nodes to locate the target as the network ages. Hence, the object tracking mechanism based on HHCM not only prolongs the life time of the network but it also collects the least amount of data required for accurate object tracking based on the constraints. Also, a sampling mechanism that dynamically changes the rate of sampling based on the movement of the object has been proposed.

In the following chapter the task of object tracking is further enhanced by building on HHCM mechanism to predict the destination of the object. HHCM defines the underlying network and the next state prediction mechanism where as PAD predicts the probability of the destination of the object and the path it would take towards that destination.

Chapter 4

Prediction-Based Algorithm for Destination-Estimation (PAD)

In recent years significant research has taken place to on the subject of object tracking using sensor networks and that research has been aimed to several different fields, such as, health, traffic management, animal migration trends. Several of those researches [2, 23, 24, 48, 49] aim to not only track the object as it moves through a sensed field but also to ascertain the destination or an exit point of the tracked object from the sensor field. Most of these research areas have focused on building models that track the object over a period of time and then use the information collected as the basis for future predictions. However, when we talk about the security and intrusion detection systems that prior information about the movement of the object is not only unavailable but also every tracking scenario within those areas can be vastly diverse and hence cannot be used as prior information.

In this chapter a Prediction-based Algorithm for Destination-estimation (PAD) is being introduced that will not only predict the destination of the object but will also predict the paths the object would take to that destination without prior information about the movement of the object.

4.1 Requirements for Destination Estimation

Carrying on from the security scenario presented in Chapter 1, here, a scenario where the enemy is moving towards a specific destination is being discussed. As the enemy enter the observed field from origin O and move towards the Destination $D3$ on the map there are several paths available to the enemy. While they move within the observed field they are being tracked by HHCM described in Chapter 3. The choice of destination is based on a set of parameters that the enemy has, for example, $D3$ might be the location of an important installation within the city or a hiding place from security. Although, the objectives that the enemy is unknown, however, a profile of the possible locations towards which the enemy is moving can be deduced. As the enemy moves within the fieldthe movement pattern is analysed and based on the possible importance of the locations, the final destination of the enemy can be predicted. Once the final destination has been ascertained the system would then predict the path or paths that the enemy could take to that particular destination.

4.2 A Prediction-based Algorithm for Destination-estimation (PAD)

To achieve the objectives mentioned above within the problem area and track the object inspiration has been drawn from Origin-Destination (OD) Estimation model from traffic management systems [18, 48, 100]. In OD traffic flows are considered time-independent and an average OD demand is determined for long-time transportation planning and road design purpose. For this purpose OD needs prior knowledge like prior traffic counts and prior destinations of the objects. It estimates the destination of the traffic by observing the traffic within segments of the paths known as traffic links, it then infuses the current traffic data with prior knowledge to determine where the traffic is headed. This model is used for road traffic management and future needs prediction.

4.2.1 The Proposed Mechanism

Within a security environment, prior information about the object is not available and hence, Kalman Filter has been employed to determine the time+1 prediction and to also predict the movement even if the object moves into a zone with no sensors or unreliably low number of sensors as explained in Chapter 3. The principles of Origin-Destination (OD) estimation have been inherited to create a set of trajectories that a particular object would follow which is then used as the basis to form the topology of sensor nodes for object tracking. It is assumed that every object being tracked would move about in the observed field with a purpose hence it would move along a certain path. These paths are made up of segments. These predicted paths can be more than one at any given time. To calculate the importance of one path over the other assumptions binomial distribution has been implemented to predict the most likely destination of the object based on the information available about the destinations, as it moves along the segments within a particular path. The whole process can be explained within the following stages:

1. Object is observed within the sensor field at an entry point.

2. Next state of the object or time+1 state of the object is calculated using HHCM as explained in Chapter 3.

3. Mini-clusters are formed over the observed field in the regions where the object has access.

4. Probable trajectory of the moving object is calculated by stitching together the mini-clusters within an observed field.

5. These mini-clusters would then observe the field and transfer the observation information to the mini-cluster head.

6. Mini-cluster heads aggregate the data and pass the most relevant data to the cluster head.

7. The cluster head further aggregates the data received from multiple mini-clusters then calculate the destination of the moving object based on the tracking information and required percentage chance of binomial distribution.

Figure 4.1 shows the flow of information as an object is observed.

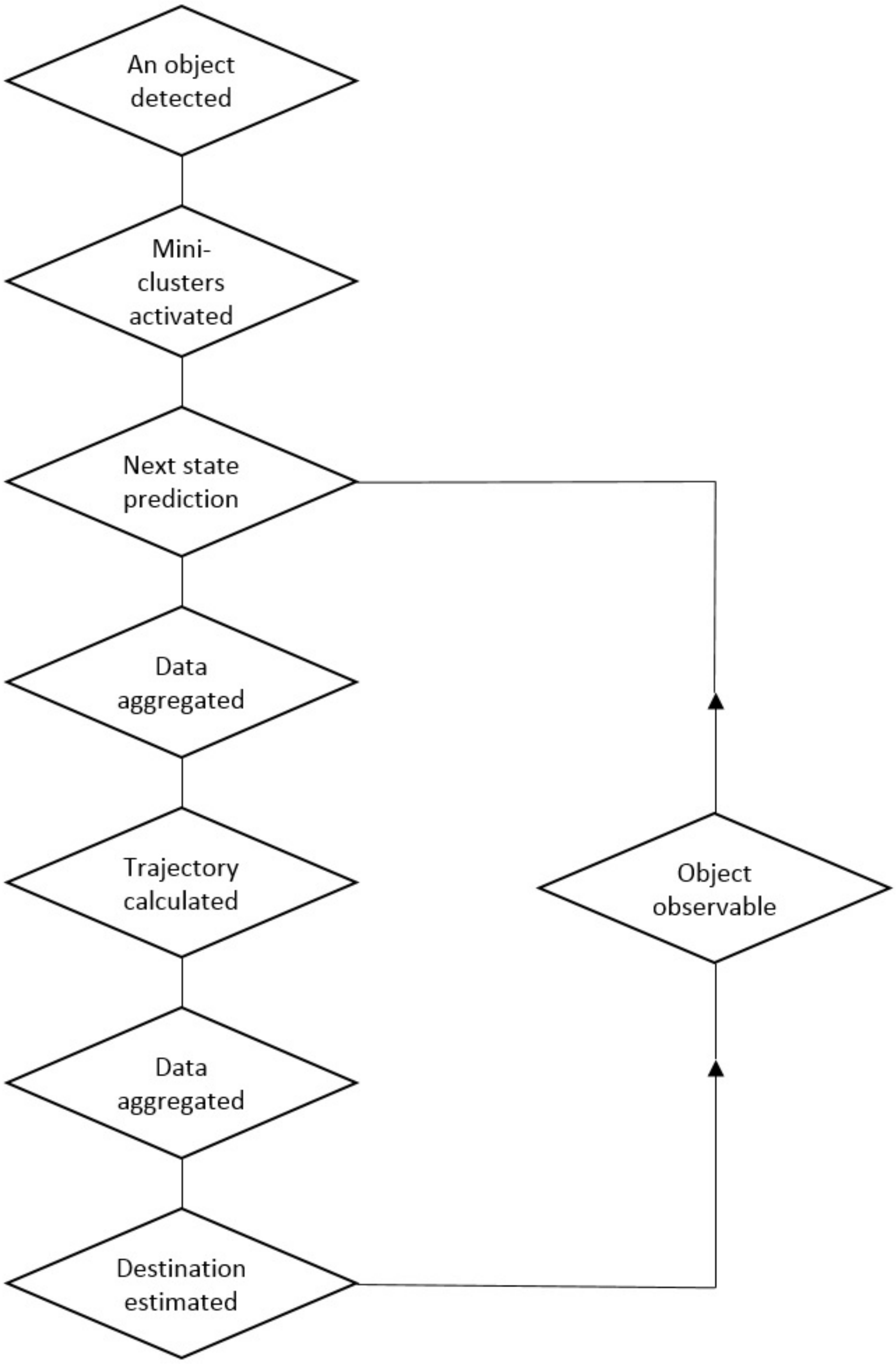

Figure 4.1: Information flow during object tracking

Algorithm 3: Prediction-based Algorithm for Destination-estimation

An object detected;
Activate mini-cluster;
loop:
Observe field;
Send object location to mini-cluster head;
if *mini-cluster head* **then**
  Calculate position estimate: x_k;
  Aggregate data;
end
Send aggregated date to cluster head;
if *Cluster head* **then**
  Collect information from all mini-clusters;
  Aggregate data;
end
if *Destination path unknown* **then**
  Calculate Q;
  (Path from an origin O to destination D)
else
  Update $P(Q)$;
  (Probability of the path being a true path) Update $P(D)$;
  (Probability of the object moving to the destination D)
end
goto *loop*

Algorithm 3 explains the general flow of PAD as the object is first observed and the and its location estimate $x_k$is calculated, that location data is then aggregated by the mini-cluster and sent to the cluster head which aggregates data from multiple mini-clusters and transmits the data to the sink. At sink the paths Q are computed based on each origin O and destination D. Once the paths that an object can take are computed the probability $P(D)$ of the object going to a particular destination D is calculated.

Keeping the stages of tracking mentioned above in mind, the destination estimation approach is divided into three levels, dynamic clustering, next state prediction and path prediction.

Assume that a network of sensors monitoring the movement of an object in a scenario based on the Figure 1.1 from Chapter 1 is deployed. The observed field has entry point referred to as the origin, and exit points denoted as the destinations. A particular destination of the object is predicted by identifying the paths which connect the sectors beginning at the origin O and ending at a destination D. The number of intermediate sectors linked to obtain these paths depends on the density of the sensor network. Each sector of O is observed at an entrances into the observed region, and a destination track in D is observed at one of the many destinations. A group of interlinked sectors S obtained by sensing the environment. A set of paths Q, where each path q is a subset of Q are ascertained and are represented as a set of sectors, $(o_q,S_q;\ d_q)$, with $o_q$is a subset of O and d_q is a subset of D representing the origin and destination sectors of the Path. Similarly, $S_q = (s_q^{(1)},......s_q^{(n)})$ is an ordered set of intermediate segments which are linked to form the trajectory. These segments are ordered by the time of initiation.

$$Q = max_Q P(S|Q)P(Q) \tag{4.1}$$

Where $P(S|Q)$ is the probability of the segments in S being true segments.

$$P(S|Q) = \prod_{q \epsilon Q} \prod_{s \epsilon S_q} \frac{P_{tp}(s)}{P_{fp}(s)} \tag{4.2}$$

Where $P_{tp}(s)$ and $P_{fp}(s)$ are probabilities of the segment being a true positive and false positive respectively. It is assumed that a Markov-chain model connects intermediate segment $s_q^{(i)}$ in a trajectory Q, to the subsequent $s_q^{(i+1)}$ with a probability given by

$$P(s^{(i+1)} s_q^{(i)}) \tag{4.3}$$

This leads to probability of the trajectory as

$$P(Q) = \prod_{q \epsilon Q} P(q) \tag{4.4}$$

Once the next state prediction is incorporated within the segments of the path generated, a binomial distribution of the probability of the object moving towards any particular destination can be calculated. If the probability has to be determined that the moving object would move towards a destination then certain information relating to the importance of a particular destination for a tracked object has to be known. In case of a mall,a person who travelled to the mall by car would prefer a destination that leads to the car park through a lift. This information is combined together by creating a matrix M which takes into account the importance of a certain destination or why a certain destination could be preferred over any other.

So to calculate the chance percentage b that the tracked object would chose a certain destination from a point in the observed location while following a certain probable path.

$$P(D) = \sum_{i=b} (b)\, M^b (1 - M)^b \tag{4.5}$$

Where b is the percentage of chance we want to calculate, like if there is a 80% chance that the object would move to a particular destination, M is the matrix of importance.

4.2.2 Importance Matrix for Destination Estimation

In order to calculate the binomial distribution prediction of the the enemy movement within the observed field a comprehensive knowledge of the observed field is obtained. As explained the previous sections that PAD assumes that the object moving within the observed field has a purpose behind it and without any prior information about the movement of the object,the destination could then be determined based on how important a certain destination is to the object. For that purpose a matrix of importance has been developed for any certain destination called the M Matrix. It can be represented as:

TABLE 4.1: M Matrix representation

	D1	D2	D3	D4	D5
A1	m_{D1A1}	m_{D2A1}	m_{D3A1}	m_{D4A1}	m_{D5A1}
A2	m_{D1A2}	m_{D2A2}	m_{D3A2}	m_{D4A2}	m_{D5A2}
A3	m_{D1A3}	m_{D2A3}	m_{D3A3}	m_{D4A3}	m_{D5A3}
A4	m_{D1A4}	m_{D2A4}	m_{D3A4}	m_{D4A4}	m_{D5A4}

This matrix contains information about the destination and why a certain destination is more desirable than others. Based on the location of the objects and its movement within the field a preference value can be attached to any destination. For example, in Figure 1.1, as the enemy move towards a destination D from the origin O the value of the M matrix would increase or decrease based on the movement of the object in combination with other conditions A, such as, time of day, importance of a destination, level of security, weather and more. These conditions mentioned here are not fully inclusive and depending on the scenario different conditions could effect the probability calculations. A sample data can be shown in the following form:

TABLE 4.2: M Matrix sample data

	D1	D2	D3	D4	D5
Vulnerability	0.25	0.5	0.75	0.25	0.25
Weather	0.75	0.75	0.5	0.75	0.6
Time of day	1	1	0.5	0.75	0.25
Importance	1	0.5	0.75	0.25	0.25

As the above Table 4.2 shows that the values of any condition can be different for any destination. For example, in the scenario described in Chapter 1, it can be assumed that the enemy is moving towards $D1$ because of the importance of the location, which can be a more likely destination for the enemy than a military base which would be heavily protected by armed security. The importance of that destination is then combined together with other factors like time of the day, weather conditions among others. Although the enemy may be going towards $D1$ but during afternoon hours on a clear sunny day it would be difficult to move in a city undetected so the enemy would most likely want to go to a destination that has more cover for them to hide till a more suitable time.

In order to calculate the value of the M matrix, first and foremost, the level of vulnerability of each destination must be known. Several research works have focused on calculating the vulnerability of any particular location [101, 102]. Once the threat of an enemy is identified, a vulnerability assessment must be performed. The vulnerability assessment considers the potential impact of loss from a successful attack as well as the vulnerability of the destination to an attack. Impact of loss is the degree to which the working of a destination is impaired by a successful attack. A key component of the vulnerability assessment is properly defining the ratings for impact of loss and vulnerability [101, 102]. These definitions may vary greatly from destination to destination. Vulnerability of any destination would be a combination of the attractiveness of a destination as a target and the level of defence provided by the existing security. Target attractiveness can be a measure of a destination in the eyes of an enemy and is influenced by the function or the symbolic importance of the facility. For example, in a military base an ammunition storage facility would be an attractive target for the enemy, so as to disable the defences but on the other hand attacking the command centre within the base can be a moral success for the enemy.

Once a destination's vulnerability has been identified, other factors are then taken into account, for example, the time of day, weather conditions, distance from a particular destination, etc. Time of day can have an impact on which destination is more accessible, for example, during afternoon hours on a sunny day a structure like an Air Hanger would be difficult to access, however, day time hours might be more desirable for a target like a busy market place. Likewise, as an object moves within an observed field its direction of movement can also be an indicator to its potential destination so the distance in combination with direction of movement can also be an important factor in determining the M matrix value of any destination from a given point during the tracking process. The conditions that would make any given destination a target for the enemy can be numerous and would depend on the a particular scenario. Suppose there are 5 destination $D1, D2, ...D5$, in order to calculate the probable target for the enemy, sum of m values for each pair of destination D and condition A.

$$m_a = \sum_{i=1}^{n} m_{D_j A_i} \tag{4.6}$$

This M is then normalised to 1 to manage the variations is data

$$M = \frac{m_a - min(m)}{max(m) - min(m)} \tag{4.7}$$

Where $min(m)$ would be 0 but the value of $max(m)$ would depend on the number of conditions A that have been employed for calculation in any given scenario. PAD would then assign the normalised value of M to every single mini-cluster for every possible destination.

As the object then moves through the observed field its destination is estimated while making adjustments to the estimates based on further movements. However, any object tracking system can suffer from missed prediction leading to missing the object as tracking of the object is dependent on the destination estimation and the next state predictions. In that case the object has to be rediscovered, for that purpose we have developed an energy efficient Multi-level Recovery Mechanism.

4.2.3 A Multi-level Object Recovery Mechanism

As the object moves through the observed field any number of problems can lead to loss of object. This could occur because of many reasons including but not limited to:

1. Localization faults: Localisation is a complicated procedure and any fault can result in the estimated location being the wrong one. These faults may have a collective effect on the overall mechanism of object tracking.

2. Network Failure: The WSN may fail because of communication breakdown, computational overload, environmental factors and etc.

3. Node Failures: WSNs have limited basis and nodes have limited battery. Node defeat may happens because of hardware defeat, battery discharge, enemy parasite, etc.

4. Prediction Errors: The mini-cluster heads are activated as the object moves into its range. This activation is initiated by the static cluster head. If activation of the mini-cluster is affected for any reason, communication failure, inaccurate state prediction, etc., and cluster activation is delayed then that can lead to loss of object.

There can be more reasons, such as, but not limited to, object trying to evade detection, object being lost behind an obstacle, etc. for the loss of object. In order to regain tracking of the object recovery process is then initiated. The recovery mechanism is initiated when a mini-cluster cluster head and its nodes cannot detect the object. This loss of object is then reported to the Cluster head by the mini-cluster head. If the mini-cluster head does not regain tracking of the object within a stipulated interval of time, the cluster head would then initiate the recovery procedure. The recovery process is based on four stages:

1. **Local Search:** As the object loss is reported further mini-clusters are formed by the cluster head and instructed to sense the environment around them to recover the object. The mini-clusters formed in all four directions of the last known location of the object which then initiate search. If the object is detected, it is reported to the cluster head and the location estimates of the object tracking are updated. In Figure 4.2 as the object is lost from cluster $C1$ subsequent mini-clusters are formed ($C2, C3, C4$ and $C5$) and they sense the environment around them to recover the object. Depending on the speed of the object's movement, most objects lost at a certain point within the observed field should lie within the proximity of the previous known location and hence forming additional mini-clusters should lead to object recovery. Local search mechanism keeps the activated nodes to the minimum while trying to recover the object. Object recovery is a redundant task that can lead to energy inefficiency, hence, activating the smallest number of nodes for recovery maintains energy efficiency.

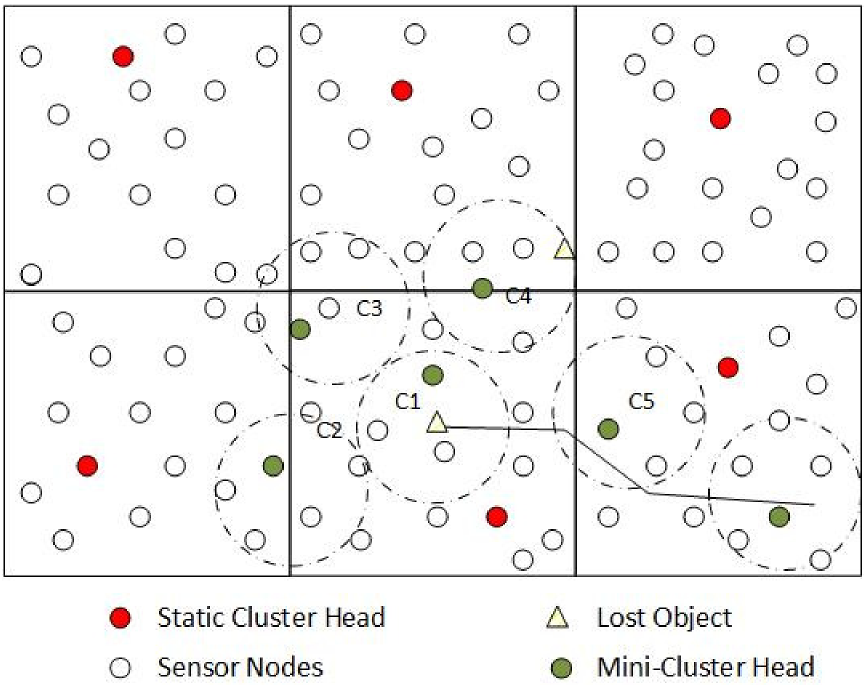

FIGURE 4.2: Local search recovery mechanism

2. **Cluster Search:** If the object is not detected by the local search mechanism within a fixed amount of time the mini-clusters are dissolved and a larger cluster wide search is initiated, where every node in the static cluster then senses the environment around it in an attempt to detect the object and reports directly to the cluster head. For example in Figure 4.3 the whole static cluster has been activated to track the object. Upon object detection a mini-cluster is formed around the object and location estimates are updated. Cluster search mechanism contains a larger number of nodes and hence can lead to more energy consumption during the recovery process. This method is less efficient as compared to Local search but in a security scenario object recovery and detection has a higher priority than energy conservation if the object is lost.

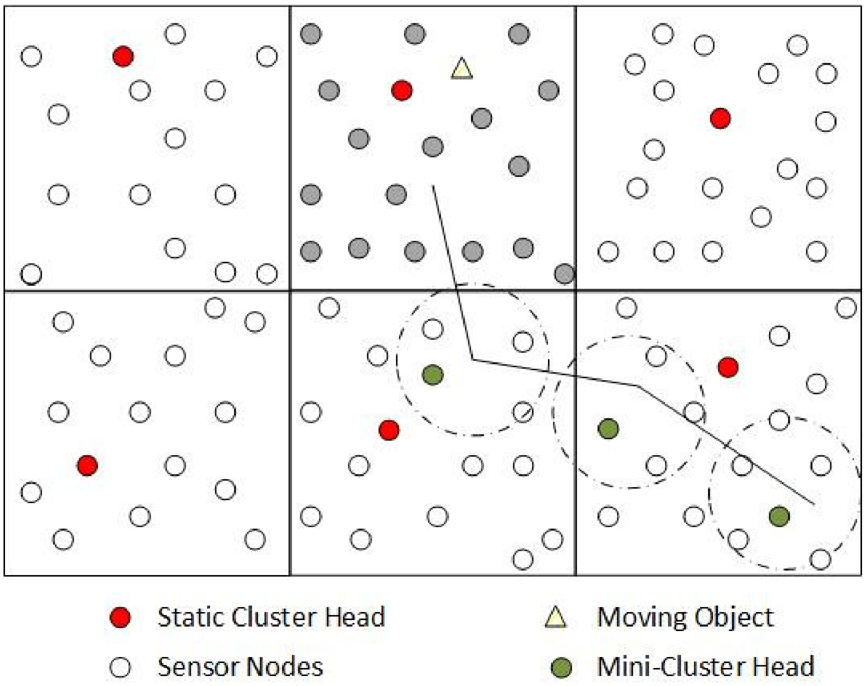

FIGURE 4.3: Cluster search recovery mechanism

3. **Multi-Cluster Search:** If the cluster fails to recover the object being tracked then a distress message is sent to all accompanying static cluster heads. These static clusters then initiate the search for the object by activating all nodes and observing the environment as shown in Figure 4.4. If the object is detected the state estimates are updated after forming mini-cluster around the object. Multi-cluster search enhances the redundancy of the recovery mechanism. A large number of nodes are activated, which leads to not only increase in the sensing cost of the network but also the transmission and computational costs as the data is transmitted to the cluster head which could be several hops away from the farthest nodes of the cluster.

4. **General Search:** In the event that multi-cluster search fails a general distress message is transmitted through-out the network and a network wide search is initiated, where all nodes in the network are activated. This is the least desired recovery mechanism and a matter of last resort. If the object is still within the observed field is would be detected by at least one node in the network. However, this method reduced the network to a naive tracking approach, where energy efficiency is not desired in favour of locating the object. General search

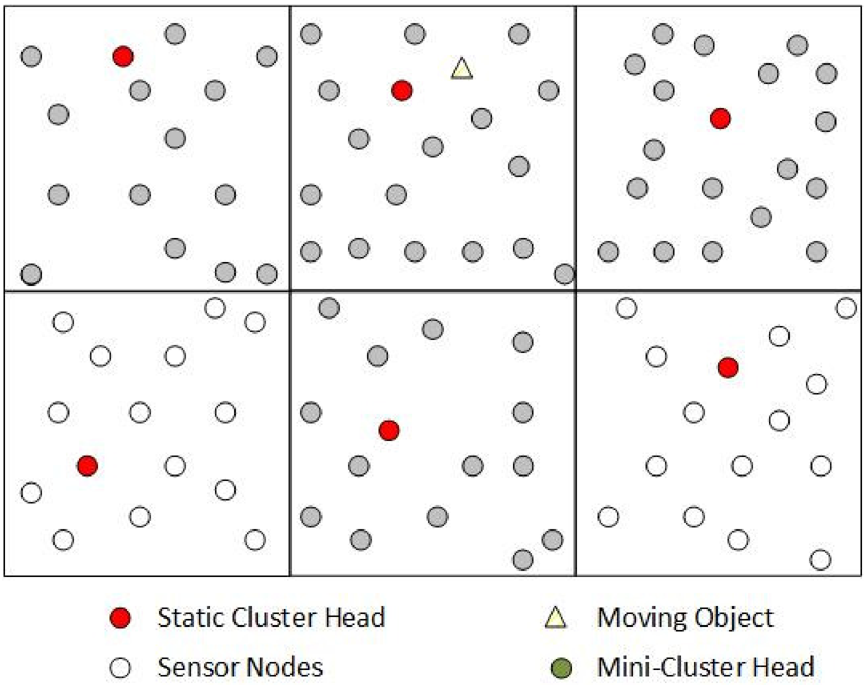

FIGURE 4.4: Multi-cluster search recovery mechanism

would lead excessive overhead communication and redundant data collection and transmission.

4.3 Discussion

In this chapter a novel object tracking mechanism called Prediction-Based Algorithm for Destination-Estimation (PAD) has been proposed. PAD calculates the destination of the object based on the reasons why the object might want to move within an observed field. That reasoning is calculated based on the importance matrix called the M Matrix value of a certain destination. M matrix becomes the importance parameter in the binomial distribution which predicts that the destination of the object. In order to calculate the path of the object to any particular destination, the mini-clusters that have been created for object tracking are stitched together to form paths from any point within the field to any possible destination. The information about the possible destination and the paths that are available to an object within the observed field, can then be, in a security scenario, used to intercept the moving object.

Chapter 5

Experimental Evaluation

In this chapter, the performance of HHCM and PAD are evaluated using extensive simulations. Tracking parameters and assumptions are defined which then leads to the comparisons of the HHCM and PAD against the existing approaches.

In order to set-up our experiments, first the existing hardware and software environments that are available for wireless sensor networks for object tracking are briefly discussed.

5.1 Software and Hardware Environments

It is important to be aware of the software and hardware environment that are available for sensor networks to build an experimental set-up that complements the available technologies. Here, software and hardware systems are discussed.

5.1.1 Software Environments

Several software environments have been researched and developed in the last two decades. Some of the popular Operating Systems and their main features are discussed in this section.

TinyOS

TinyOS is a open source operating system with flexible component based design. It has very low memory requirements with only 400bytes for the core system. It also contains an extensive library of network protocols and drivers for different sensors. As a component based system, it can be modified based on application requirements [103]. More recent versions of TinyOS provide multi-threading support along side an event driven programming model. TinyOS also provides a wide range of communication protocols including but not limited to TDMA, Z-MAC, B-MAC and DIP. With scarce memory availability it is designed to operate one application at a time. Other features include support for simulation environments, communication security and C programming language through NesC.

Contiki

Contiki is another light weight open source operating system. Major features include a multitasking kernel, pre-emptive multi-threading, TCP/IP networking, a personal web server through a simple telnet client [104]. Contiki provides access to a wide array of communication protocols including TCP/IP, UDP, ICMP. However, contiki does not provide support for real-time applications and also lacks inherent support for communication security.

MANTIS

The MultimodAl system for NeTworks of In-situ wireless Sensors (MANTIS) is a multi-threaded operating system for sensor network hardware [105]. MANTIS does provide support for real-time applications albeit to a limited levels. However, it provides robust simulation support through AVRORA with C programming language. LiteC++ is used for programming applications.

LiteOS

LiteOS is a University of Illinois developed operating system similar to Unix [106]. LiteOS provides support for multi-threading and can run multiple applications at the same time. LiteOS like MANTIS also supports AVRORA for simulation support.

5.1.2 Hardware Environments

For object tracking in an security and surveillance systems the choice of sensor hardware is limited due to very specific requirements, such as low power, wide capture area, low processing time, and high data reliability. Different sensor types that fit this set-up are:

Passive Infra-red Sensors (PIR)

PIR is a thermal sensor that detects the heat in the infra-red spectrum [107]. As the wavelength of normal light is a lot less than that of infra-red, which means it is not visible to naked eye, however, a pro-electric sensor can detect it through its crystalline materials that produce a charge when exposed of infra-red radiation. PIR can be very precise with the distance measurements and provide a wide range of 30 degrees of detection angle.

Ultrasound Sensors

Ultrasound sensors are some of the most simple sensors for object tracking with low computation overheads [108]. Recent researches have shown that ultrasound sensors can be used to precisely determine the shape of an object while it moves at high speeds. It emits sound waves that bounce back from the object and can then be used to determine the distance and shape based on the returning sound waves. Ultrasound sensors can have large detention range but data precision is effected at larger distances.

Optical Sensors

Optical sensors unlike PIR capture the visible light spectrum [109]. They capture the entire field of view and are some of the most commonly used sensors in security and surveillance environments. However, they pose some exciting research opportunities due to the recording of information in frames that can then be used to detect the movement of an object as it changes position from one frame to the other. They tend to generate a large amount of data and can be computationally heavy for an object tracking environment.

5.2 Experimental Set-up

A large number of nodes are deployed over a geographical area. Although, nodes are deployed in a specific area for object detection, however, any specific node placement is not assumed and the nodes are mostly deployed randomly. This random deployment does, however, makes sure that the nodes are distributed evenly and there are enough nodes available in any sub region for object tracking. It also assumes that there is a localisation service that can provide absolute location to a certain number of nodes from where other nodes can determine their relative location.

The medium of communication between these nodes is wireless and broadcast is the basic communication method. These nodes have limited communication range and hence, multi-hop communication may be required to transmit data from one location to the other. For the purposes of reporting results of the tracking activity, one or more node may be attached to a relay which can transmit to a base station or sink.

Each node in the network has a unique identity and consists of a processor, memory, transceiver, battery, and a sensing unit. Some nodes also contain a localization unit such as a GPS system. These nodes have limited energy resources and hence perpetual active state is not advisable for network longevity. A single node has limited processing power, memory, and hence, computationally intensive processes are not advisable at any single node.

In order for sensor nodes to localise an object being tracked, they must know their own location. For that purpose it is assumed that small percentage of nodes are capable of determining their own location with the help of GPS units built in to the node. Although, GPS units tend to have a margin of error, the GPS signal in space has a global average user range error of <7.8 meters (25.6 feet), with 95% probability [1]. For the purpose of determining the accuracy of the proposed protocols it is assume that the location determined from the GPS is absolute and the rest of the network can also determine the absolute location of every node. In the experiments the GPS margin of

[1] Official U.S. government information about the Global Positioning System (GPS): http://www.gps.gov/systems/gps/performance/accuracy/

error has not been taken into account when the difference between the actual location and the location is computed by our protocols.

Based on the experimental setup and node design the following assumptions are made.

5.3 Assumptions

For the algorithm proposed in this research, the following assumptions are used regarding the sensor nodes and the underlying network:

- The WSN consists of one sink and a large number of immobile sensor nodes. The sensor nodes are randomly deployed over a 2D square area to monitor the environment.
- All nodes have the same radio, battery, memory, and processing capabilities at the start of the experiment to make sure that any node in the experiment can assume the role of mini-cluster head without being at any disadvantage.
- Each node is aware of its location's coordinates and the coordinates of its neighbouring sensor nodes.
- The communication links between sensors are bi-directional.
- Sensor nodes are capable of multi-hop communication to relay data.
- The sensor are not affected when exposed to the environmental elements, such as, water, air temperature, humidity, direct sunlight, etc.

Based on the above assumptions, the basic network set-up can be viewed as:

5.4 Evaluation Metrics

In order to evaluate the robustness and efficiency of PAD, HHCM and the sampling mechanism we have established the following evaluation metrics:

Table 5.1: Simulation Parameters

Parameters	Values
Node density	$1/6m^2$
Number of nodes	270
Total area	$100m^2$
Distribution of nodes	Random
Sensing range	$15m$
Communication range	$60m$
Average sampling duration	0.5 seconds

1. Energy Consumption

   Energy consumption is one of the most important evaluation metric because of the limited energy reserves that a sensor has at its disposal. Energy consumption, however, can be expanded for different tasks. For that reason the energy consumption during communication, sampling the environment and computation are calculated. Excessive sampling can lead to energy depletion due to sensing of the environment, computation of the collected data and transmission of that data.

2. Amount of Data Generated

   Another important aspect of sensor network is the amount of data generated. In the scope of this research data refers to the localisation information of the moving object. Date is generated by individual nodes and mini-clusters. This data is then used to predict the future location and eventually the destination that the object is moving towards. Excessive data generation can lead to transmission costs and sensing costs that reduce the life-time of the network.

3. Number of Nodes Used

   The importance of this evaluation metric is that if an excessive number of nodes are used for data collection, then the activity of all those would result in not only excessive sensing costs but also the increase the communication costs for that data. Within the proposed mechanisms nodes become part of the mini-clusters based on their location.

4. Duration of Node Activity

   Although, nodes are activated and deactivated during the course of the tracking process, it is an important statistic to know the the duration that any single node stays in active state. This metric would be used to prove that the load of tracking process is more evenly distributed among the nodes and no single node or group of nodes is taxed more than others.

5. Localisation Accuracy

   The localisation accuracy is calculated as the difference between the real location of the object and the predicted location of the object as determined by the proposed approach. When tracking an object, the accuracy of information is important, especially in security environment so that the appropriate security measures can be taken. Within the scope of this research, accuracy of tracking is required to be at the confidence level of 80% that the object would be moving to a certain destination. To achieve that level of accuracy the sampling rate of the environment changes dynamically. Sampling rate can have a significant impact on the localisation accuracy of the object as a conservative sampling rate can lead to object being missed or faulty data generated which reduces the accuracy of the whole tracking process. This sampling rate is dependent on the speed of movement of the tracked object and the required tracking accuracy.

5.5 Simulation Results and Analysis

This section explains the simulation conducted and the scenarios considered to evaluate the efficiency of HHCM, PAD and the Sampling mechanism. Based on the parameters in Table 5.1 and the evaluation metrics following experiments to evaluate PAD, HHCM and the sampling mechanism have been devised.

During the simulation it was observed that on average of just above four (4.09) nodes formed a mini-cluster, however, each mini-cluster had different number of nodes with the largest cluster containing eight nodes and the smallest cluster containing 2 nodes. Table 5.2 shows the nodes in any given mini-cluster during one of our test runs:

Table 5.2: Node Distribution in all Mini-clusters

Mini-cluster Number	Nodes in mini-cluster
1	4
2	3
3	7
4	3
5	8
6	4
7	5
8	4
9	2
10	2
11	3

Based on the information mentioned above it can also be highlighted that as these mini-clusters are formed within and across static clusters, the number of nodes in any mini-cluster is not dependent on the size of the static cluster. Hence, the mini-clustering structure of HHCM is independent of the underlying static structure.

5.5.1 Experimental rationale

Before the experiments were carried-out to compare the performance of HHCM and PAD against existing approaches, it was determined that the proposed framework has certain unique features which pose difficulty in comparing the proposed protocols directly against existing approaches. In order to make sure that we carried out comparison to highlight the efficiency gains, it was decided to compare parts of the proposed protocols against the existing approaches.

HHCM is compared against the naive activation approach [110] and zonal activation approach [111] of object tracking. Naive approach gives us a comparison point where all nodes are activated. Naive activation leads to high energy consumption and hence, provides a point of reference. Zonal activation is closely matched to HHCM where only a small number of nodes are activated within a zone and hence, provides a comparison when only a small number of nodes track an object and they are activated and deactivated as the object moves through the field. Several zonal approaches have been proposed in literature [112–114].

PAD is compared against two major protocols CODA and PES. Although, they do not predict the destination of the object, however, both these approaches have been known to provide wide ranging energy efficiency. CODA [12, 23, 24] is based on hybrid clustering model, where, dynamic clusters are formed to track the object continuously within dynamic clusters and provides a point of comparison for the proposed hybrid structure. PES [11, 44] provides a robust energy efficiency mechanism that is based on collecting the most relevant information from the network and discarding the irrelevant information which is one of the basis over which PAD has been based. PES also implements a wake-up mechanism that activates and deactivates the nodes based on the position of the object and accuracy requirements.

By comparing the proposed protocols against the above two approaches we aim to compare the basic features of this research. Results will prove that the proposed approach has performed better against the energy efficiency of PES when compared in conjunction with the proposed dynamic sampling mechanism. HHCM and PAD also provide tracking accuracy gains against the two existing approaches while maintaining energy efficiency.

Results of the experiments have also been compared under different conditions like change in node density, accuracy of probability in-terms of time and accuracy of data collected in-terms of its deviation from the actual location of the object. These results explain the scalability and robustness of the network and its ability to perform under different conditions.

5.5.2 Energy Consumption of mini-clusters at Different Movement Speeds

This experiment measures the energy consumption based on the speed of the object's movement. In this experiment the consumption of energy is computed by every single mini-cluster that is formed. It would prove that as the object moves through the observed field it enters mini-clusters are have varying number of nodes and that the individual nodes are not excessively used for the tracking of the object.

Energy Consumption at 1 m/sec

In this experiment, the object is moving at 1 meter per second, which is the average walking speed of a human. As Figure 5.1 shows, the proposed approach reduces the sensing and the cost associated with it has reduced considerably as compared to Continuous sensing and time-cycle sensing of 50%.

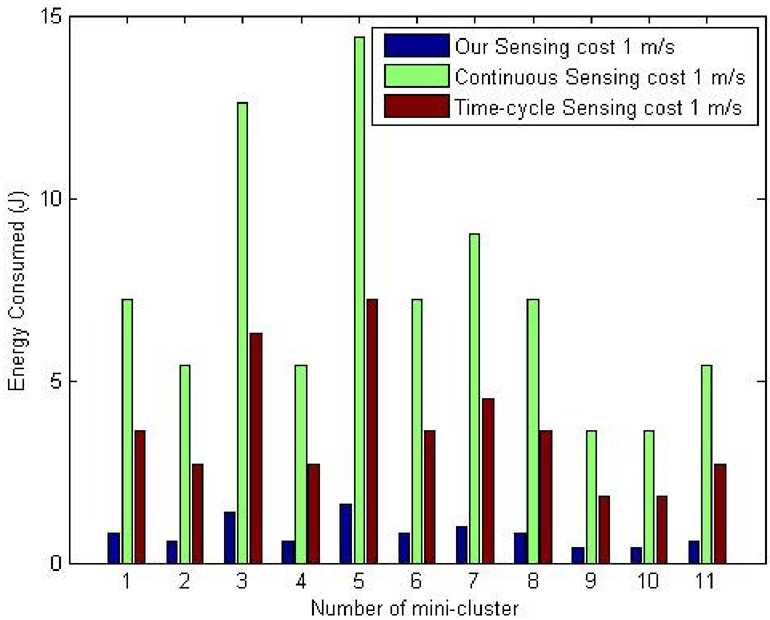

FIGURE 5.1: Comparison of energy consumption as the object moves at the speed of 1 m/sec

On average the proposed sensing scheme consumed 0.8 Joules of energy per cluster as compared to 7.3 Joules and 3.6 Joules by continuous and time-cycle mechanisms respectively. This reduction in energy consumption is due to the fact that the sampling rate of the environment within our mechanism is dynamically adjusted as the object moves through the environment. Hence, as the object moves slowly and remains at a steady pace the confidence level of the data accuracy remains stable and hence sampling rate is maintained at the minimum level leading to energy savings as compared with the static sampling rates of the other two approaches.

Energy Consumption at 5 m/sec

In this scenario, the object moves at 5 meters per second which the average cycling speed of a person.

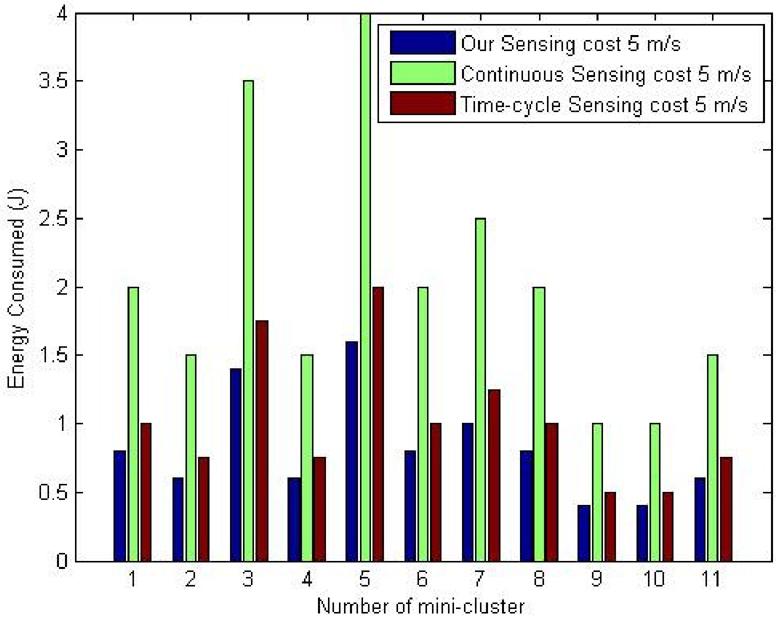

FIGURE 5.2: Comparison of energy consumption as the object moves at the speed of 5 m/sec

Figure 5.2 shows the energy consumed at each cluster as the object moves at 5 m/s. As the object moves faster it spends less time in each cluster and hence the energy consumed by the continuous sensing and time-cycle sensing is reduced but the energy consumed by our approach increases. This is due to the fact that the adaptive sampling mechanism of our approach increases the sampling rate to maintain data accuracy, where as the fixed sampling rate of the other two approaches would result in reduced data accuracy and increase the chances of object loss. However, our approach still consumes less energy than the other two approaches as it consumes 0.9J, but the continuous scheme consumes 2.04J and time cycle scheme consumes 1.02J.

Energy Consumption at 10 m/sec

As the object moves at the speed of a galloping horse, which is approximately 10 m/s it is observed that the energy consumption trends show a different picture.

As the object moves at 10 m/s (Figure 5.3) it is further observed that the time-cycle scheme consumes less energy than the proposed scheme this is because of the fact that the object spends only 2.5s in a cluster on average and that leaves it to only sense the space less times than our scheme. Thus, consuming less energy at 0.5 Joules per cluster as compared to 0.9J and 1.0J by our scheme and continuous scheme respectively.

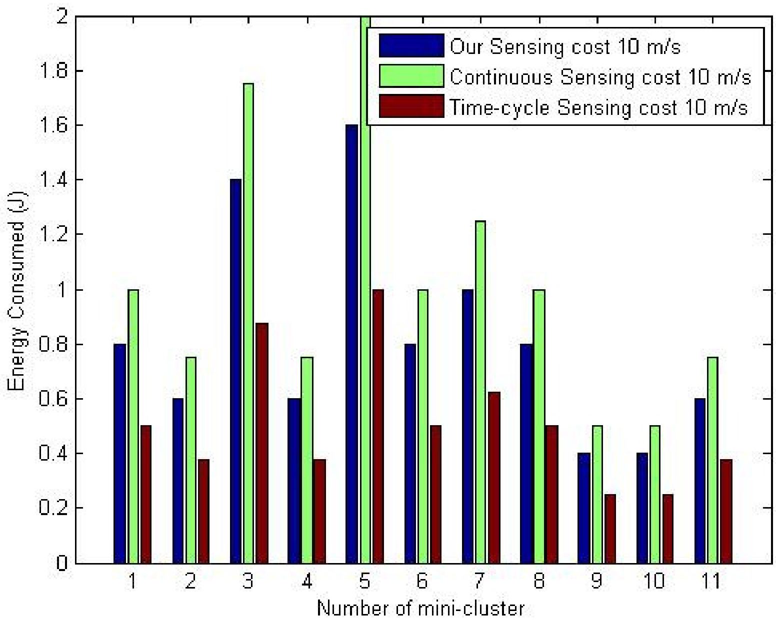

FIGURE 5.3: Comparison of energy consumption as the object moves at the speed of 10 m/sec

This result shows that although our approach does not bring the most energy savings, however, it does maintain the data accuracy. In object tracking data accuracy to a required threshold is more important than the energy savings as the loss of object due to reduced sampling could result in higher energy consumption of rediscovery of the object.

Energy Consumption at 13 m/sec

As the object moves at the speed of an average speed limit for motor vehicles in urban areas it is observed that the results have changed even more. As the object moves at higher speed the continuous sensing and the proposed approach consume the same amount of energy, as time-cycle scheme consumes less because at this speed in our scenarios the object spends only 1.5 seconds in a mini-cluster on average, which leaves the proposed approach and the continuous approach to only sense the region three times at the very most (Figure 5.4).

Even though the object spends less and less time in a cluster as it moves faster through the sensed field it can be seen that if the simulation is run at 13 m/sec speed, the required object detection accuracy of 70% to 80% (distance between the actual location

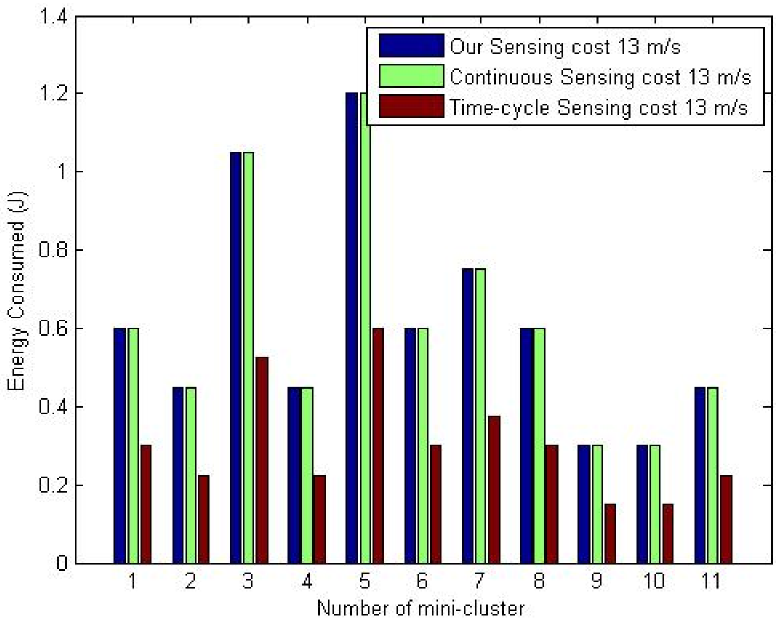

FIGURE 5.4: Comparison of energy consumption as the object moves at the speed of 13 m/sec

and predicted location), remains consistent even though the energy consumption is not the most economical.

5.5.3 Average Energy Consumption With PAD

In this scenario the proposed approach is compared against CODA. CODA is an energy efficient system for object tracking. The results of this experiments show that the PAD in combination with HHCM has a considerable low energy consumption.

Figure 5.5 shows that the energy consumed is considerably low with almost 20% difference on average energy consumed. Although, the energy consumption over 200 minutes does show varied results which can be explained by the fact that HHCM inherently does not have a consistent number of nodes within mini-clusters and this variation in node density and any sudden change in objects direction as the object moves results in variable energy consumption. However, the the average energy consumption remains low.

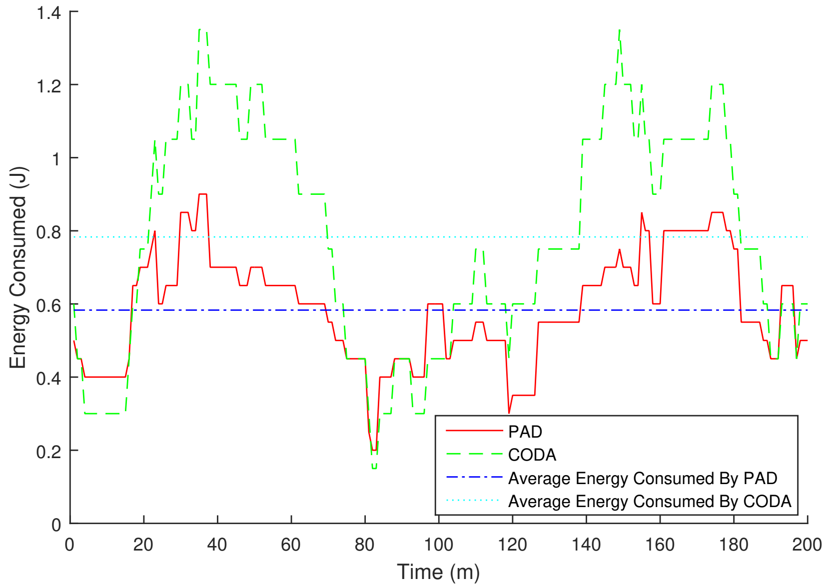

FIGURE 5.5: Comparison of average energy consumption of PAD against CODA

5.5.4 Energy Consumption Over Time With PAD

Figure 5.6 shows the comparison between PAD and PES in terms of average energy consumed over time as the object moves at a consistent speed. This comparison explains that the as time progresses PAD remains efficient and average energy consumed over a period of time is less than PES. This comparison also shows that our network uses less energy over time and hence, remains available for a lot longer.

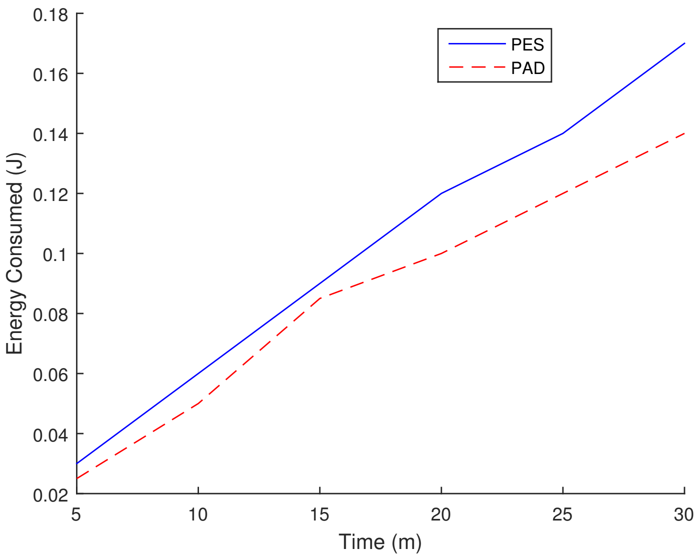

FIGURE 5.6: Comparison of energy consumption of PAD against PES over time

5.5.5 Energy Consumption at Different Movement Speeds With PAD

Figure 5.7 demonstrates the comparison between PAD and PES where the object moves at different speeds. As an object moves through the sensed field nodes are activated and deactivated to detect its location, however, as the object increases its speed a higher rate of sampling is required to make sure that the sensing nodes do not lose the object.

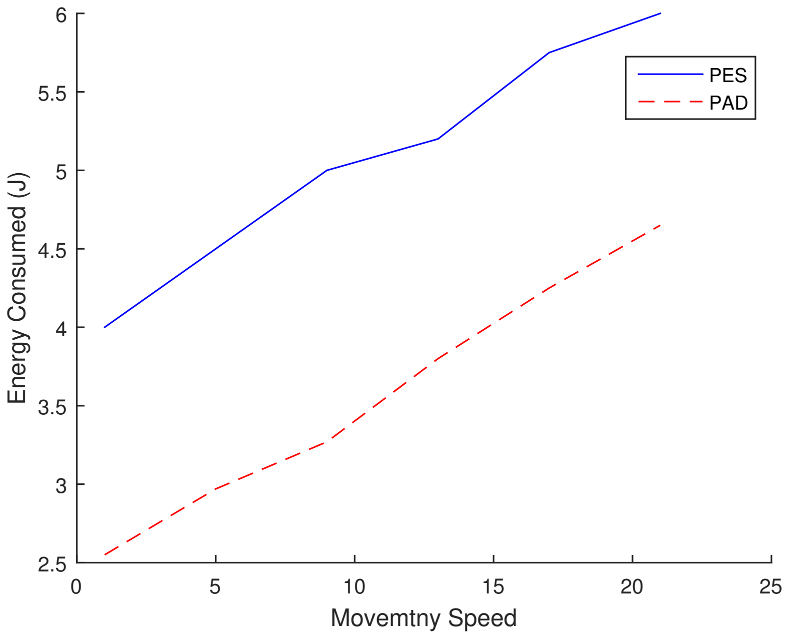

FIGURE 5.7: Comparison of energy consumption by PAD and PES at different movement speeds

Figure 5.7 shows that as the object's speed increases PAD manages to keep the energy consumption to a reduced rate. As the speed of the moving object increases PES can be seen as experiencing sudden rise in energy consumption levels, however, due to the adaptive sampling mechanism of PAD the energy consumption remains steady. The increase in energy consumption is due to the increase in the sampling rate which not only results in higher sensing cost, but also increased communication and computational costs in term of energy consumed.

5.5.6 Amount of Data Generated

A very significant aspect of object tacking in real time is that sensor networks can generate a large amount of data, which result in computational delays and higher transmission energy consumption.

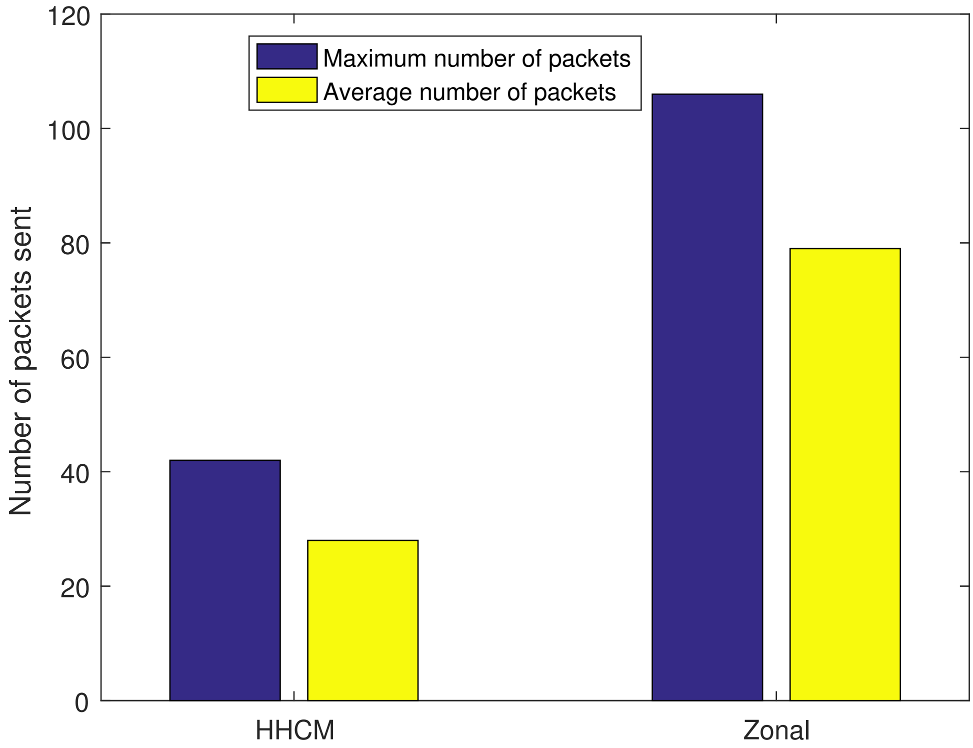

FIGURE 5.8: Average amount of Data generated by a zone/cluster during the duration of the simulation

Figure 5.8 shows that with our approach the number of packets generated for tracking an object are kept to a minimum. As the nodes are distributed randomly no two mini-clusters or zones are alike and hence each zone or cluster generates different amount of data. However with our approach the largest mini-cluster , while tracking the object, generated considerably less amount of data i.e. 42 packets while in zonal approach that largest zone generated 110 packets of data. On average our approach generates less than 25 packets while zonal approach generates more than 75 packets per zone. This reduced data generation is due to lesser number of nodes becoming part of a mini-cluster while a larger number of nodes become part of a zone and hence generate more data. Less data generation leads to lesser computational load at the mini-cluster head and lesser communication costs, both these aspects reduce energy consumption.

5.5.7 The Number of Nodes Used During Object Tracking

One of the major factors in for efficient tracking of an object is to make sure that enough nodes are available for tracking an object while making sure that redundancy is kept to a minimum. This approach has two major advantages, (a) less number of nodes utilised mean that more of the network is kept in a sleep state which minimized energy consumption and (a) also means that the amount of data generated by the network is kept to a minimum which in turn reduces delay due to computation and also reduces the energy consumed during transmission.

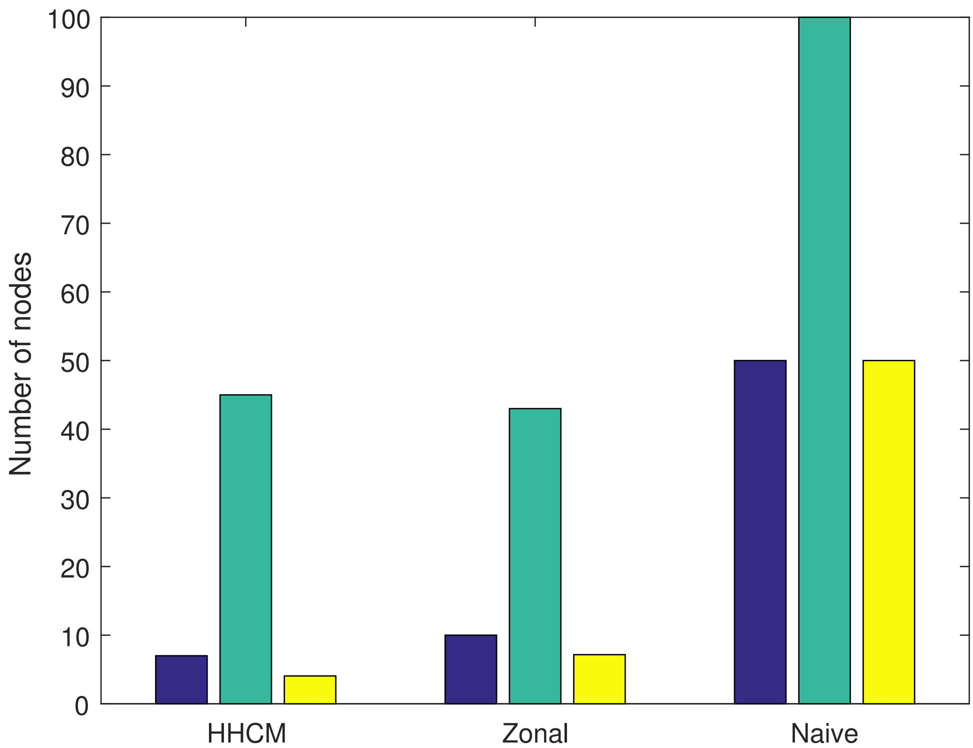

FIGURE 5.9: The Number of nodes used during object tracking

Figure 5.9 shows that with our approach the number of nodes associated with tracking at any given time are less than that of zonal and the naïve approach. The maximum number of nodes within a cluster always remains lower than in the other two approaches showing that at any given time our approach uses less number of nodes. Although, the total number of nodes utilized during the course of the simulation is higher than the zonal approach but it shows that the load is more evenly distributed among nodes during the course of the simulation. This even distribution of load among a larger number of nodes mean that no single node is utilised in excess and hence reducing the change of node failure. Within the naive and zonal approaches a smaller number of nodes are utilised putting more pressure on a smaller group of nodes. This result demonstrates that by even distribution of load not only energy consumption during tracking is split across a larger number of nodes but also that the mini-cluster head in HHCM has a small number of nodes to receive data from for computation and hence does not get taxed excessively for computation.

5.5.8 Average Duration of Node Activity

One of the major factors for a sensor network is that no one node should be required to do more work than the other nodes and the work load of the network should be divided almost equally among all nodes. If one single node or a group of nodes is allocated too many responsibilities then that node or group is more likely to run out of energy quicker than the rest of the network thus potentially creating holes in the sensing field.

Figure 5.10 shows that with our approach a single mini-cluster spends considerably less time in the awake state as compared to a zonal approach. Which highlights that no one mini-cluster stays awake for a long period of time and thus the nodes within that mini-cluster stay less time in sensing state and as the object moves out of mini-cluster that particular mini-cluster is dissolved and thus all nodes retreat to the sleep mode. In comparison with the zonal approach this result shows that no single group is awake for an extended period of time reducing the energy consumption of individual nodes and also distributing the tracking responsibility more evenly than the zonal approach. Less time in awake means that the energy consumption while sensing environment is reduced making the network more resistant to node failure due to energy depletion.

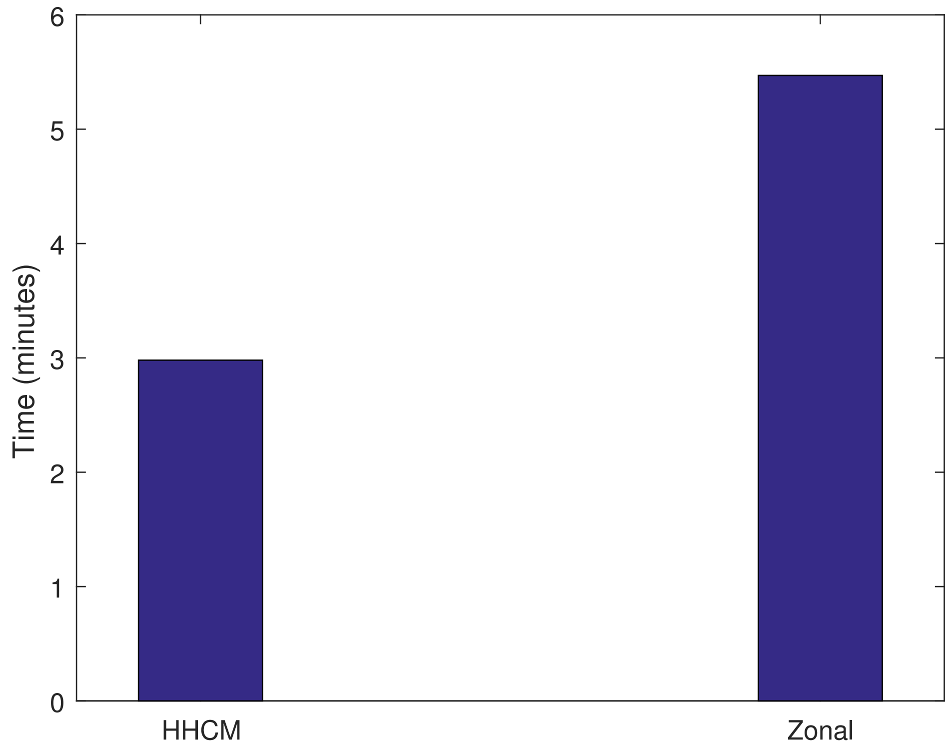

FIGURE 5.10: Average awake time for a zone/cluster

5.5.9 Localisation Accuracy

The main objective of any tracking technique is to track an object with a high certainty of accuracy while consuming the least amount of resources. It has already been observed in the previous sections that the proposed approach utilises less resources than other approaches. In this scenario, the path of the object can be observed as the object is tracked and the observed path of the object by the sensor network.

Figure 5.11 shows the path the object took while it was being tracked and two different angles of change of path. As the object moves in a circular path during the initial stages of the simulation HHCM computes the path and records that information and as the object changes its path in an unexpected way HHCM manages to adjust the readings and keep track of the object.

Figure 5.12 shows that the standard deviation of the difference between the actual location and the detected location is 1.1 with an average of 2.4. This means that the moving object is detected to within a standard deviation of 1.1 meters of the actual

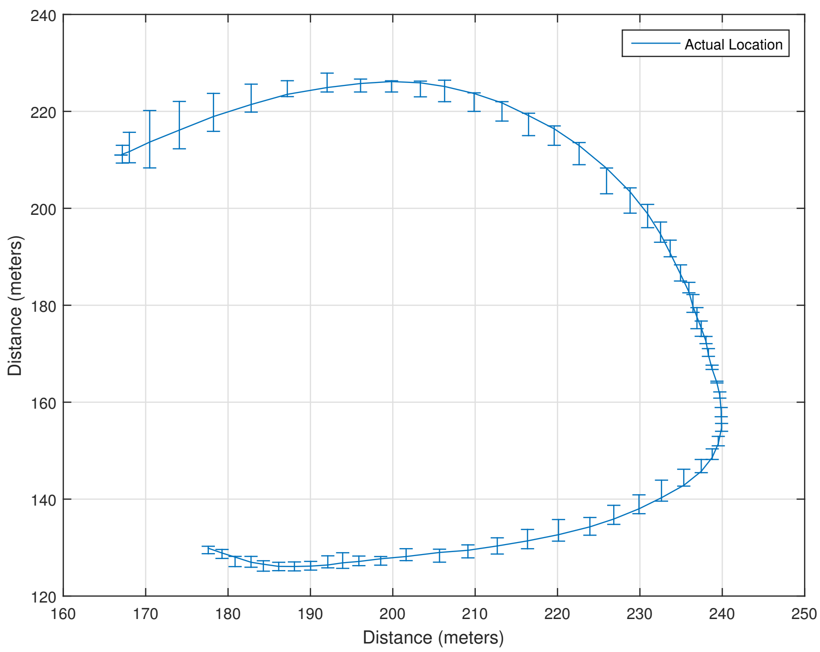

FIGURE 5.11: Accuracy of predicted location against the actual location

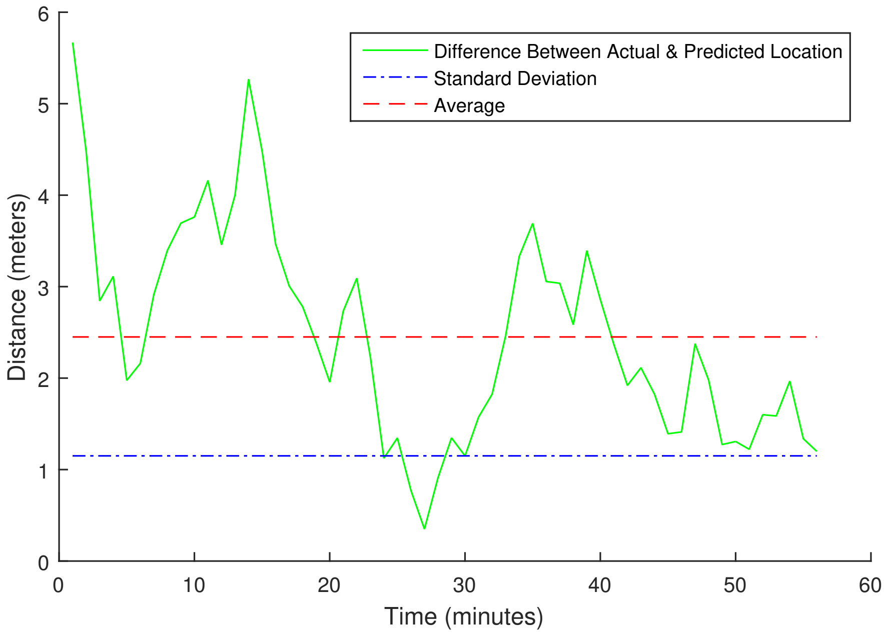

FIGURE 5.12: Accuracy of tracking data through the duration of the experiment

location of the object. Another aspect to be noted is that with our approach not all mini-clusters are of the same size because of the random distribution of nodes. The smallest mini-cluster formed during the course of the simulation consisted of only two nodes and yet even with that small number of nodes the tracking accuracy of the network is not compromised. Although there are spikes in the data which are caused by the varying number of nodes in a given mini-cluster and also due to the movement pattern of the object. With the highest difference between the actual and predicted location being close to 6 meters, however, the variation in the accuracy remains within a standard deviation of 1.1 meters.

5.5.10 Localisation Accuracy Over Time

In this scenario the accuracy of the location data is observed. Data accuracy is an important aspect of object tracking as in a surveillance or security environments data latency or inaccurate state predictions can lead to security breaches making the system useless.

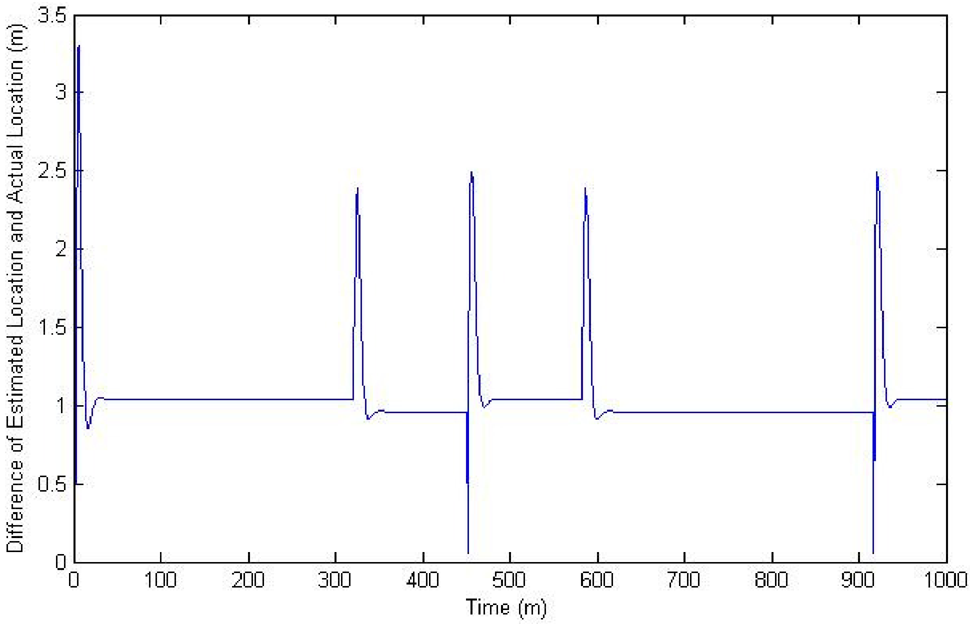

FIGURE 5.13: Localisation accuracy Over the course of a journey

Figure 5.13 show that the predicted location data of the object remains around 1 meter of the actual location of the data through most of the journey. Although, the data accuracy remains high during the course of the experiment, however, there are some spikes in the data showing reduced accuracy. This reduced accuracy is caused by any sudden change in the direction of the object's movement. In this experiment the object changed direction by 90 degrees more than once. The most spike in data comes at the initial stage of the tracking when due to lack of data about the object results in lower accuracy but as the object moves through the observed field the location estimation becomes stable and accurate.

5.5.11 Kernel Density Estimation

In this scenario the Kernel Density Estimation of the observed data is calculated. Kernel density estimation is a non-parametric way to estimate the probability density function of random data. Kernel density estimation smoothes random data to show data grouping, closer the data is grouped more accurate the original data is perceived to be.

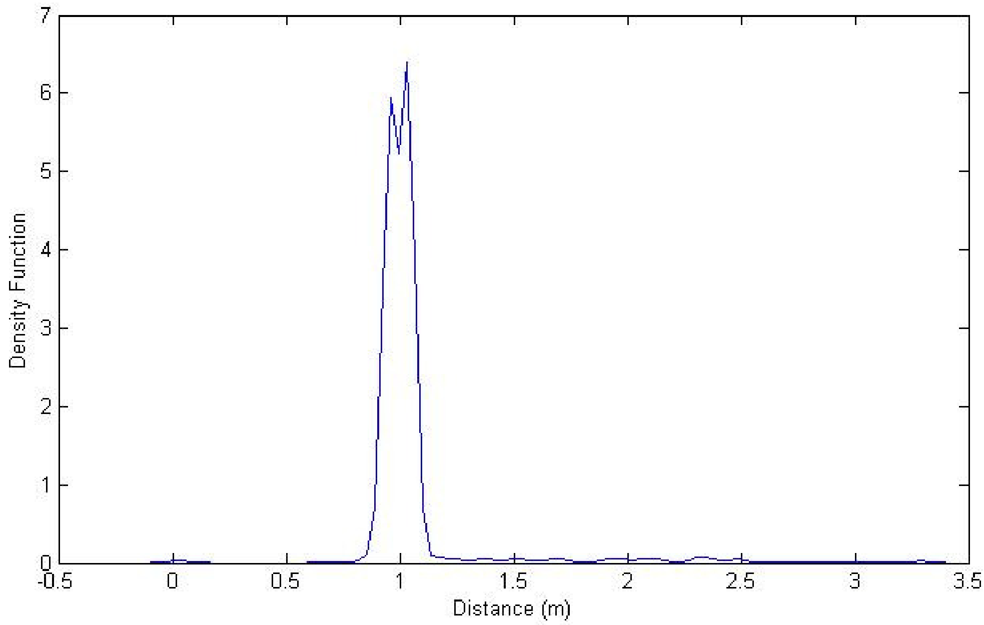

Figure 5.14: Kernel density estimate of the accuracy of collected location data

Figure 5.14 shows that the data is mostly distributed around the mark of 1 meter. This further confirms the data accuracy as explained in the previous experiment. Although, there are instances of reduced accuracy where the data suggest reduced accuracy, however, that can be explained as the sudden change in objects direction. This reduced accuracy is corrected quickly, hence, most data residing around the 1 meter mark.

5.5.12 Destination Estimation Accuracy

PAD has been designed to specifically determine the eventual destination of the object as it moves through the observed field. As the object moves through the field its behaviour is observed and then based on a matrix of destination selection the eventual destination of the object is estimated. In this scenario the object moves within the observed field the estimate of its eventual destination variates, however, as more data is gathered about the object the estimates are revised to show the final destination of the object.

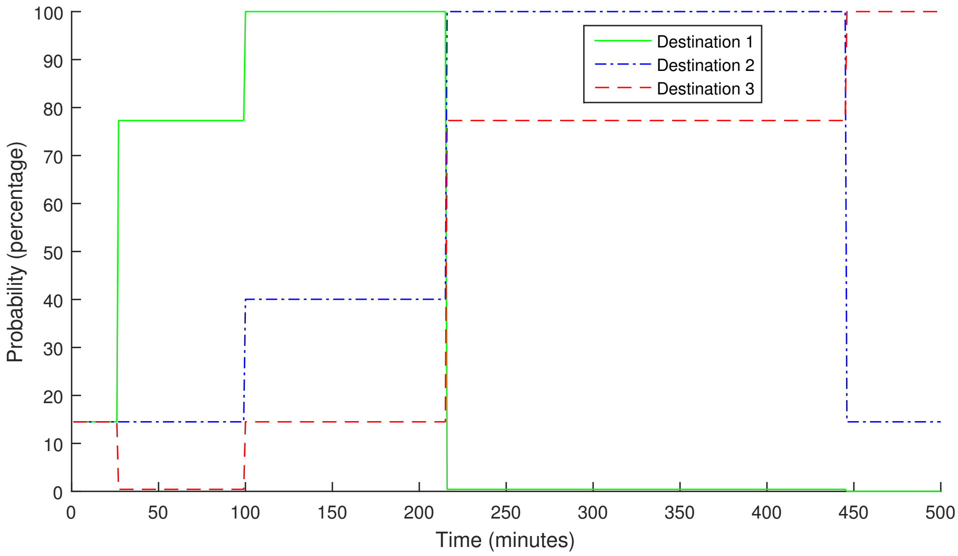

FIGURE 5.15: Accuracy of the destination estimation mechanism of PAD through the duration of the experiment

As the object moves through the observed field it take the object 500 minutes to reach its destination as shown in Figure 5.15. However, after 220 minutes of its movement within the field up to 80% certainty can be achieved about the future destination of the object and eventually 100% certainty can be achieved about the object's particular destination once it has made 80% of its journey.

5.5.13 Tracking Accuracy And Energy Consumed With Different Node Density

In this scenario we look at the accuracy of data when we variate the density of nodes in the environment. Data accuracy at different node density is tested to prove that the mechanism is robust and scalable and can deal with different node densities and also to show what effect change in node density would have on the accuracy of data. We tested this with three different node densities at 1 nodes per 3 square meters, 1 nodes per 6 square meters and 1 nodes per 9 square meters.

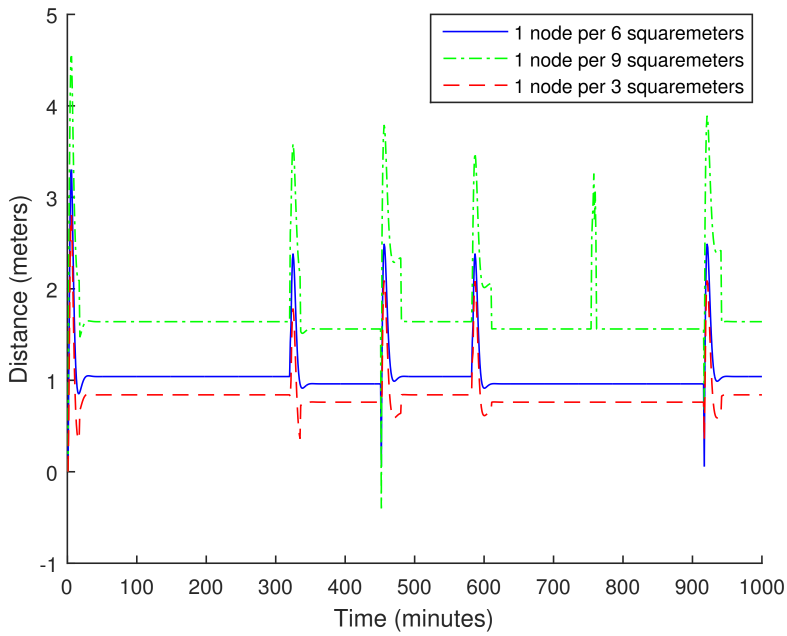

FIGURE 5.16: Comparison of the tracking accuracy with different node density

Figure 5.16 show that the predicted location data of the object remains at an average of 1.15 meters of the actual location of the data through most of the journey when the node density is at 1 node per 6 square meters. If the number of nodes is reduced in the same observed environment to 1 node in 9 square meters the data becomes more erratic and the average data accuracy is reduced to 1.8 meters from the actual location of the object. Likewise, if the data density is increased to 1 node in 3 square meters data accuracy is increased to 0.9 meters from the actual location. It must be observed that as the data accuracy does increase it does not bring much improvement to the overall accuracy with an increase of 0.25 meters from the actual location, however, when the node density is reduced the data accuracy goes down by 0.65 meters from the actual location. This reduction in accuracy is more significant than the gains made at higher density. This proves that the data accuracy discussed in Figure 5.13 being of high significance with an optimum number of nodes.

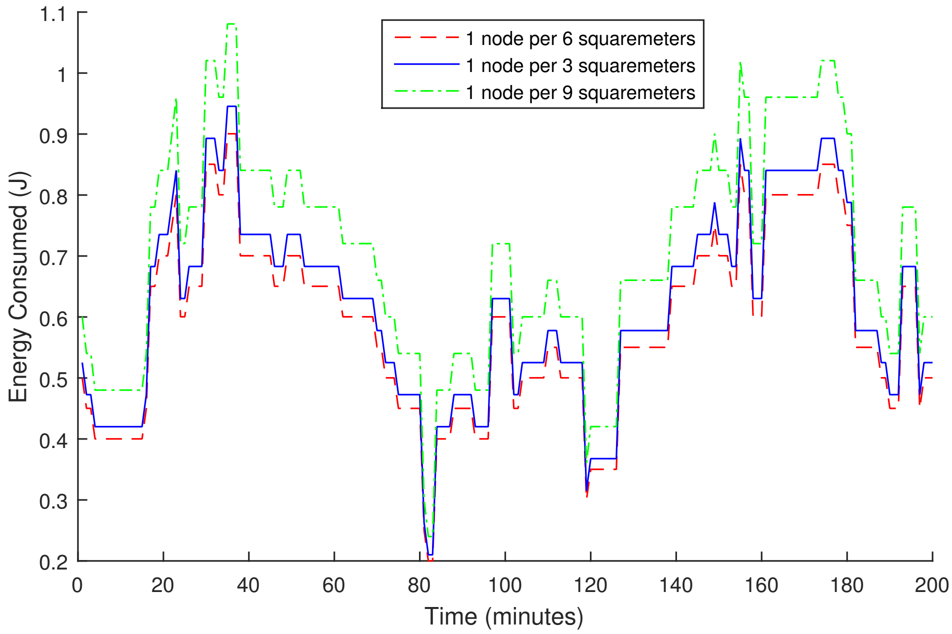

FIGURE 5.17: Comparison of energy consumption at different node density

When the energy consumed is analysed at different node density (Figure 5.17), the results show that the energy consumption increases when the node density is increased or decreased. When the node density is increased the energy consumption in increased due to excessive communication and computational costs, and when the node density is reduces the energy consumption in increased due to the excessive sampling of the environment to maintain data accuracy. This result can be used to determine the optimum node density at which the data accuracy and energy consumption are both at an acceptable levels.

5.6 Discussion

This chapter performance of PAD, HHCM and the sampling mechanism have been explored. Three effective parts of the algorithms are examined in a wide range using simulations to find the most appropriate values as explained in the Evaluation Matrices.

The proposed approaches have shown to prolong the network life time and increased the efficiency of the network while tracking the target. In addition, the impact of the deployed sensor nodes density on data accuracy and energy consumption has also been explored. Results have also shown that the energy efficiency of the HHCM and PAD have considerable improvement over the existing approaches. Experiments have also shown that the accuracy of data can be increased by effective management of sampling mechanism and results demonstrate that the improvements are significant.

While results of the experiments discussed in this chapter have highlighted the energy efficiency, data accuracy and future location estimation accuracy, there are certain aspects of the experiments that can have an adverse effect on the results. One of the aspects in the M matrix, although, it is assumed in this research that the data about the observed field is accurate and can be rusted, however, in case of faulty or missing data from M matrix the results of the destination estimation can be effected. This aspect of the research, however, was not within the scope of the project and future work on the project can incorporate this scenario.

Chapter 6

Conclusion and Future Work

In recent years wireless sensor networks have been deployed for many real world applications like disaster monitoring, security, surveillance, animal migration, health systems and so on. Sensor networks have become more useful for object tracking and several modern cities have deployed sensors, such as, optical and acoustic sensors for monitoring and crime prevention. However, these systems have several drawbacks, such as, the monitoring of the system still remains predominantly and human activity. In this thesis the use of wireless sensor networks for object tracking and destination estimation within a security scenario has been explored. Object tracking has been researched extensively in recent years, however, the task of destination estimation has been limited to certain domains within object tracking like traffic management, vehicular networks, etc., where destination estimation has been conducted with the aid of prior data about traffic and vehicle journeys. Destination estimation of a tracked object within security and surveillance domains has been limited due to lack of prior information. Every security scenario can be unique and hence, prior information of security breaches cannot be used to determine future incursions.

6.1 Summary

Within this thesis, the application of wireless sensor networks and important implementation techniques and protocols have been discussed. The wireless sensor networks, theoretical characteristics and system constraints, and currently available networking architectures and deployment topologies have also been introduced.Also, some of the related protocols and algorithms for object tracking have been discussed.

In this thesis a novel and comprehensive framework for object tracking using sensor networks has been proposed where not only the object is tracked in real time but also the destination of the object is determined. In this work, several goals have been successfully achieved, which are:

1. *Accurately predict destination of object without any prior information about the object's movement.*
2. *Develop an efficient network architecture to reduce the number of nodes involved in tracking process.*
3. *Obtain the most relevant information from the network by reducing the amount of data collected to reduce the communication and computational costs while maintaining the required level of data accuracy.*

For this purpose the following approaches have been developed:

1. A novel Prediction-based Algorithm for Destination-estimation (PAD). PAD not only predicts the destination that an object would take to a particular destination but also predicts the path(s) that the object could take to that destination. A traffic management system called Origin-Destination (OD) Estimation has been the inspiration of this project. OD estimation predicts the destination of the object with the help of prior data about traffic journeys. However, in a security environment the intruder would only make a decision to go to a particular destination because of its importance. As in security environment the luxury of prior information about a intrusion is not available, a Matrix of Importance for any given destination has been proposed. With the help of this matrix and the movement pattern of the object in the observed space the destination of the object and the path the would take to that particular destination is computed.

Experiments have been conducted to determine the accuracy with which PAD calculates the destination of the object. These experiments have been discussed in detail in this Chapter 5. Also experiments have been conducted to determine the energy efficiency of PAD and compared them against the existing approaches of object tracking. PAD not only predicts the destination with a high degree of energy efficiency but also maintains a high level of data accuracy. Results of the experiments have shown that the proposed approach can predict the destination of the object with 80% accuracy when the object has travelled less than half on its total journey and with 100% certainty that the object is moving towards a particular destination once it has made 80% of its journey, maintaining a high level of energy efficiency.

PAD operates on a new Hierarchical Hybrid Clustering Mechanism (HHCM) that has been developed for object tracking with wireless sensor networks.

2. A New Hierarchical Hybrid Clustering Mechanism (HHCM). HHCM has been developed as an underlying network architecture to minimise energy consumption of the sensor network while it accurately predicts the object's future location. The factors affecting energy efficiency include the amount of data collected and the number of nodes active at any given time. Excessive data collection leads to higher processing and transmission costs in terms of energy and high number of active nodes lead to excessive data collection and sensing energy costs. Sensor nodes inherently have limited energy supplies and these energy supplies are not only required for sensing the environment but also to compute the data collected and transmission of that data. So as a network the energy consumption increases if more nodes are in active state or if they are collecting excessive amounts of data through sensing operation.

   HHCM is designed to form dynamic mini-clusters within and across static clusters along the predicted future location of the object that is being tracked and a mini-cluster head then collect the information from its neighbouring nodes. HHCM is split across triple level hierarchy, where ordinary nodes form part of the dynamic mini-cluster which are managed through a mini-cluster head. This mini-cluster head itself reports to the cluster head of the larger static cluster. Mini-clusters perform a dual role in HHCM, not only do mini-clusters help to reduce the number of nodes made active at any given time to perform tracking which reduces the data collection and processing costs in terms of energy but also carry out part of the computation to reduce amount of data that is transmitted by filtering out the redundant data. This data is then sent to the static cluster head which not only receives the most concise data from the mini-cluster but also the most useful data, hence, reducing the computational load from the cluster head.

To test the efficiency of this architecture, experiment have been conducted which measured the number of nodes used during the object tracking, amount of data generated from within each mini-cluster, average time a particular mini-cluster spends in active state. In all these experiments the efficiency of the architecture is measured by comparing it against the naive and zonal node activation mechanisms. The results show that by employing this architecture not only the number of nodes are reduced which limits the energy consumption of the network during sensing operations but also spreads the tracking duties amoung various mini-clusters. This distribution of duty leads to larger collaboration and reduces the load from any particular node. Reduced amount of data generation from within a mini-cluster leads to reduced energy consumption during data transmission. Experimental results show that this architecture has significant energy savings.

3. A dynamic sampling and Bayesian filtering mechanism for extraction of most relevant data and prediction of next state location.

   For an efficient object tracking sensor network, it needs to collect the most relevant data with a mechanism to discard any redundant or excessive data. For this purpose a sampling and next state prediction mechanism has been developed that can not only observe the environment selectively but also selects the most relevant data out of the collected samples. The sampling rate of the observed environment is dynamically changed to maintain a predetermined level of data accuracy. This mechanism also predicts the next state of the object which has multi-fold advantages, not only it provides an estimate of when to observe the environment again, it also predicts the movement pattern of the object. This observation of the movement pattern can then be combined with PAD to generate more accurate transition predictions, while at the same time filtering the data for any unwanted information. For this next state prediction we have employed Kalman filtering which is a Bayesian filtering mechanism. The Kalman Filter is an algorithm that uses a series of measurements observed over time, containing random variations and other inaccuracies, and produces estimates of unknown variables. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state. This sampling and prediction mechanism has been built on top of the HHCM to engage the smallest number of nodes possible for data reduction and then further curtailing the data with this robust sampling mechanism. By further reducing the data with the aid of the Kalman filtering we reduce the communication costs in terms of energy and by performing this at mini-cluster head we distribute the computation cost among several nodes.

To analyse the performance of the proposed sampling and next state prediction mechanism experiments have been conducted to calculate the energy efficiency and the data accuracy of the next state prediction. The actual location of the data against the predicted location is analysed, average energy consumed during this process, and all this while the movement speeds of the object variate to test the dynamic and adaptive nature of the sampling mechanism. Experimental results show that the gains in energy saving and at variable movement speeds of the object are considerable, while maintaining the desired level of data accuracy, which in our case shows that the location estimates stood at a standard deviation of 1.1 meters from the actual location.

6.2 Future Work

This research project was set out to develop an object tracking mechanism that can predict the future location and the destination of the object without any prior information about the object, while maintaining energy efficiency to enhance the life of the network. Existing approaches make use of the prior or historical date when trying to determine the destination of an object, however, within security scenarios prior information is almost never available. During the course of this research this aim of the research has been successfully achieved as has been demonstrated through the experimental evaluation. However, this research project has highlighted additional avenues of research that would enhance the working of this mechanism.

1. **Firstly**, this project has been developed in a modular form where mini-clusters can be formed and dissolved as desired by the tracking application. This modular form not only is capable of tracking a single object but should also be able to track multiple objects. Multiple object tracking has not been explored as part of this mechanism and hence provides an opportunity to further develop this mechanism to incorporate multi-object tracking.

2. **Secondly**, the M matrix which is proposed within this research project could be further developed to incorporate the a significant margin of error due to missing, incomplete or faulty data about the environment. Within the scope of this research the errors within the M matrix were not explored as that would have deviated this project from the intended aim and objectives. Further research within this domain could reduce the number of assumptions about the environment and help to make this tracking mechanism more robust and less prone to any errors.

3. **Thirdly**, within this research project the random node deployment has been used. Further research into different deployment mechanism could be conducted to determine if their are any advantages to deploy nodes in any particular topology or form. Random node deployment is easier and quicker in real life and hence has been the choice for this research, however, quick deployment is not always required. Hence, research into a comparative study of different node deployment mechanism could highlight if further gains can be made in tracking accuracy and energy conservation.

Bibliography

[1] W. Zhang and G. Cao, "DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks," *IEEE Transactions on wireless communications*, vol. 3, no. 5, pp. 1689 – 1701, 2004.

[2] H.-J. Chang and G.-T. Park, "A study on traffic signal control at signalized intersections in vehicular ad hoc networks," *Ad Hoc Networks*, vol. 11, no. 7, pp. 2115 – 2124, 2013.

[3] B. W. Peyton, A. Pothen, and X. Yuan, "Partitioning a chordal graph into transitive subgraphs for parallel sparse triangular solution," *Linear algebra and its applications*, vol. 192, pp. 329 – 353, 1993.

[4] H. Yang and B. Sikdar, "A protocol for tracking mobile targets using sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pp. 71 – 81, IEEE, 2003.

[5] S. Bhatti, J. Xu, and M. Memon, "Clustering and fault tolerance for target tracking using wireless sensor networks," *IET wireless sensor systems*, vol. 1, no. 2, pp. 66 – 73, 2011.

[6] S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris, "Energy efficient routing in wireless sensor networks through balanced clustering," *Algorithms*, vol. 6, no. 1, pp. 29 – 42, 2013.

[7] D. Gao, W. Zhu, X. Xu, and H.-C. Chao, "A hybrid localization and tracking system in camera sensor networks," *International Journal of Communication Systems*, vol. 27, no. 4, pp. 606 – 622, 2014.

[8] H. Chan, M. Luk, and A. Perrig, "Using clustering information for sensor network localization," in *International Conference on Distributed Computing in Sensor Systems*, pp. 109 – 125, Springer, 2005.

[9] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61 – 72, 2002.

[10] S. Phoha, J. Koch, E. Grele, C. Griffin, and B. Madan, "Space-time coordinated distributed sensing algorithms for resource efficient narrowband target localization and tracking," *International Journal of Distributed Sensor Networks*, vol. 1, no. 1, pp. 81 – 99, 2005.

[11] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293 – 313, 2002.

[12] H. Yang and B. Sikador, "A protocol for tracking mobile targets using sensor network, sensor network protocols and applications," *IEEE International workshop on sensor network protocols and applications*, pp. 71 – 81, 2003.

[13] L. Song and D. Hatzinakos, "A cross-layer architecture of wireless sensor networks for target tracking," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 1, pp. 145 – 158, 2007.

[14] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 70, IEEE Press, 2005.

[15] J. Li and Y. Zhou, *Target tracking in wireless sensor networks.* INTECH Open Access Publisher, 2010.

[16] H.-T. Kung and D. Vlah, *Efficient location tracking using sensor networks*, vol. 3. 2003.

[17] Z. Wang, W. Lou, Z. Wang, J. Ma, and H. Chen, *A novel mobility management scheme for target tracking in cluster-based sensor networks.* 2010.

[18] Y. Shen, K. T. Kim, J. C. Park, and H. Y. Youn, "Object tracking based on the prediction of trajectory in wireless sensor networks," in *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*, pp. 2317 – 2324, IEEE, 2013.

[19] S. Oh and S. Sastry, "Tracking on a graph," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 26, IEEE Press, 2005.

[20] H. Lin, J. A. Rushing, S. J. Graves, S. Tanner, and E. Criswell, "Real time target tracking with binary sensor networks and parallel computing.," in *GrC*, pp. 112 – 117, 2006.

[21] V. S. Tseng and E. H.-C. Lu, "Energy-efficient real-time object tracking in multi-level sensor networks by mining and predicting movement patterns," *Journal of Systems and Software*, vol. 82, no. 4, pp. 697 – 706, 2009.

[22] V. S. Tseng and E. H.-C. Lu, "Energy-efficient real-time object tracking in multi-level sensor networks by mining and predicting movement patterns," *Journal of Systems and Software*, vol. 82, no. 4, pp. 697 – 706, 2009.

[23] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: Taking lessons from MPEG," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82 – 98, 2001.

[24] Y. Xu, J. Winter, and W.-C. Lee, "Dual prediction-based reporting for object tracking sensor networks," in *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pp. 154 – 163, IEEE, 2004.

[25] L. Song and D. Hatzinakos, "A cross-layer architecture of wireless sensor networks for target tracking," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 1, pp. 145 – 158, 2007.

[26] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE signal processing magazine*, vol. 19, no. 2, pp. 17 – 29, 2002.

[27] K. A. Darabkh, S. S. Ismail, M. Al-Shurman, I. F. Jafar, E. Alkhader, and M. F. Al-Mistarihi, "Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 2068 – 2080, 2012.

[28] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE transactions on mobile computing*, vol. 3, no. 3, pp. 258 – 271, 2004.

[29] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 270 – 283, ACM, 2004.

[30] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, *et al.*, "VigilNet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 1 – 38, 2006.

[31] C. Gui and P. 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*, pp. 129 – 143, ACM, 2004.

[32] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 12, pp. 1696 – 1710, 2006.

[33] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple target tracking with RF sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1787 – 1800, 2014.

[34] A. Kose and E. Masazade, "Adaptive sampling with sensor selection for target tracking in wireless sensor networks," in *2014 48Th Asilomar Conference on Signals, Systems and Computers*, pp. 909 – 913, November 2014.

[35] W. Zhang and G. Cao, "DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks," *IEEE Transactions on wireless communications*, vol. 3, no. 5, pp. 1689 – 1701, 2004.

[36] H. Zha, J. J. Metzner, G. Kesidis, *et al.*, "Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks," in *Communications, 2004 IEEE International Conference on*, vol. 7, pp. 3807 – 3811, IEEE, 2004.

[37] Z. Wang, H. Li, X. Shen, X. Sun, and Z. Wang, "Tracking and predicting moving targets in hierarchical sensor networks," in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pp. 1169 – 1173, IEEE, 2008.

[38] J. Guo, H. Zhang, Y. Sun, and R. Bie, "Square-root unscented Kalman filtering-based localization and tracking in the internet of things," *Personal and ubiquitous computing*, vol. 18, no. 4, pp. 987 – 996, 2014.

[39] A. Kose and E. Masazade, "Adaptive sampling with sensor selection for target tracking in wireless sensor networks," in *2014 48Th Asilomar Conference on Signals, Systems and Computers*, pp. 909 – 913, November 2014.

[40] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processing*, vol. 25, pp. 1 – 16, 2014.

[41] A. Bechar, S. Y. Nof, and J. P. Wachs, "A review and framework of laser-based collaboration support," *Annual Reviews in Control*, vol. 39, pp. 30 – 45, 2015.

[42] A. Rai, S. Deswal, and P. Singh, "MAC Protocols in Wireless Sensor Network: A Survey," *International Journal of New Innovations in Engineering and Technology*, vol. 5, no. 1, pp. 95 – 101, 2016.

[43] B. Krishnamachari and S. Iyengar, “Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241 – 250, 2004.

[44] K.-W. Fan and P. Sinha, “Distributed online data aggregation for large scale sensor networks,” in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5Th IEEE International Conference on*, pp. 153 – 162, IEEE, 2008.

[45] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, “Distributed target classification and tracking in sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163 – 1171, 2003.

[46] S. Zhang, C. Wang, S.-C. Chan, X. Wei, and C.-H. Ho, “New object detection, tracking, and recognition approaches for video surveillance over camera network,” *IEEE Sens. J.*, vol. 15, no. 5, pp. 2679 – 2691, 2015.

[47] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, “Hierarchical and networked vehicle surveillance in ITS: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 557 – 580, 2015.

[48] P. Ye and D. Wen, “Optimal Traffic Sensor Location for Origin-Destination Estimation Using a Compressed Sensing Framework,” *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[49] B. M. Sanandaji and P. Varaiya, “Compressive origin-destination estimation,” *Transportation Letters*, vol. 8, no. 3, pp. 148 – 157, 2016.

[50] M. J. Saeed, L. Han, and M. K. Muyeba, “An energy efficient and resource preserving target tracking approach for wireless sensor networks,” in *2014 9Th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP)*, pp. 232 – 237, July 2014.

[51] M. J. Saeed, L. Han, and M. Hammoudeh, “Energy Efficient Sampling Mechanism for Object Tracking with Wireless Sensor Networks,” WorldS4 2017, 2017.

[52] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, “Mobile object tracking in wireless sensor networks,” *Computer communications*, vol. 30, no. 8, pp. 1811 – 1825, 2007.

[53] W.-R. Chang, H.-T. Lin, and Z.-Z. Cheng, "CODA: A continuous object detection and tracking algorithm for wireless ad hoc sensor networks," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5Th IEEE*, pp. 168 – 174, IEEE, 2008.

[54] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, "Mobile object tracking in wireless sensor networks," *Computer communications*, vol. 30, no. 8, pp. 1811 – 1825, 2007.

[55] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, no. 2, 2007.

[56] L. Xiangqian, Z. Gang, and M. Xiaoli, "Target localization and tracking in noisy binary sensor networks with known spatial topology," *Wireless Communications and Mobile Computing*, vol. 9, no. 8, pp. 1028 – 1039, 2009.

[57] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *Decision and Control, 2004. CDC. 43Rd IEEE Conference on*, vol. 1, pp. 735 – 742, IEEE, 2004.

[58] L. Montero, M. Pacheco, J. Barceló, S. Homoceanu, and J. Casanovas, "Case Study on Cooperative Car Data for Estimating Traffic States in an Urban Network," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2594, pp. 127 – 137, 2016.

[59] A. Oracevic and S. Ozdemir, "A survey of secure target tracking algorithms for wireless sensor networks," in *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pp. 1 – 6, January 2014.

[60] M. Akter, M. O. Rahman, M. N. Islam, and M. A. Habib, "Incremental clustering-based object tracking in wireless sensor networks," in *Networking Systems and Security (NSysS), 2015 International Conference on*, pp. 1 – 6, IEEE, 2015.

[61] S. Li, Z. Qin, L.-X. Shan, R. Zhang, and X. Yang, "A Survey on Target Tracking in Well-deployed Wireless Sensor Networks.," *JSW*, vol. 9, no. 5, pp. 1255 – 1262, 2014.

[62] S. Cho, L. Han, B. Joo, and S. Han, "P-LEACH: An efficient cluster-based technique to track mobile sinks in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[63] M. K. Farahani, I. Arak, and J. A. Torkestani, "A Learning Automata Based Prediction Mechanism for Target Tracking in Wireless Sensor Networks," 2015.

[64] J. Li and J. Cao, "Survey of Object Tracking in Wireless Sensor Networks.," *Adhoc & Sensor Wireless Networks*, vol. 25, 2015.

[65] C. Kim, H. Cho, S. Kim, T. Yang, and S.-H. Kim, "Sink Mobility Support Scheme for Continuous Object Tracking in Wireless Sensor Networks," in *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*, pp. 452 – 457, IEEE, 2016.

[66] T. Winkler and B. Rinner, "Security and privacy protection in visual sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 2, 2014.

[67] T. Ngo-Quynh, V. Tran-Quang, and Q. Nguyen-Trung, "A low-latency communication protocol for target tracking in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1 – 15, 2016.

[68] T. Ngo-Quynh, V. Tran-Quang, and Q. Nguyen-Trung, "A low-latency communication protocol for target tracking in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1 – 15, 2016.

[69] M. Akter, M. O. Rahman, M. N. Islam, and M. A. Habib, "Incremental clustering-based object tracking in wireless sensor networks," in *Networking Systems and Security (NSysS), 2015 International Conference on*, pp. 1 – 6, IEEE, 2015.

[70] Y. Shen, K. T. Kim, J. C. Park, and H. Y. Youn, "Object tracking based on the prediction of trajectory in wireless sensor networks," in *High Performance*

Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on, pp. 2317 – 2324, IEEE, 2013.

[71] A. L. Rodriguez and M. Stojanovic, "Adaptive object tracking in a sensor network," in *OCEANS 2015-Genova*, pp. 1 – 5, IEEE, 2015.

[72] R. Darman and N. Ithnin, "Object tracking methods in wireless sensor network: Network structure classification," in *IT Convergence and Security (ICITCS), 2014 International Conference on*, pp. 1 – 3, IEEE, 2014.

[73] O. Demigha, W.-K. Hidouci, and T. Ahmed, "On energy efficiency in collaborative target tracking in wireless sensor network: A review," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1210 – 1222, 2013.

[74] C. Sergiou, P. Antoniou, and V. Vassiliou, "A comprehensive survey of congestion control protocols in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1839 – 1859, 2014.

[75] Z. Wang, W. Lou, Z. Wang, J. Ma, and H. Chen, "A hybrid cluster-based target tracking protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, 2013.

[76] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel, "DTA: Deployment and Tracking Algorithm in Wireless Multimedia Sensor Networks.," *Ad hoc & sensor wireless networks*, vol. 28, no. 1 - 2, pp. 115 – 135, 2015.

[77] N. Sahu, S. Dubey, and T. D. Diwan, "Energy Efficient Communication Routing Protocols in Wireless Sensor Network Services," *International Research Journal of Engineering and Technology (IRJET) e-ISSN*, pp. 2395 – 0056, 2015.

[78] Q. Liu, Z. Wang, X. He, and D. H. Zhou, "Event-based recursive distributed filtering over wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2470 – 2475, 2015.

[79] D. Benferhat and J. F. Myoupo, "A Physical DPT and Regional CSP-Based Hybrid Algorithm for Energy Efficiency in Target Tracking in Wireless Sensor

Networks," in *Networks (ICN), 2010 Ninth International Conference on*, pp. 127 – 132, IEEE, 2010.

[80] A. T. Ihler, J. W. Fisher, and A. S. Willsky, "Particle filtering under communications constraints," in *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, pp. 89 – 94, IEEE, 2005.

[81] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," in *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, pp. 13 – 11, IET, 2001.

[82] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 181 – 188, IEEE, 2005.

[83] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 181 – 188, IEEE, 2005.

[84] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," vol. 1, pp. 735 – 742, IEEE, 2004.

[85] C. S. Jensen, H. Lu, and B. Yang, "Graph model based indoor tracking," in *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pp. 122 – 131, IEEE, 2009.

[86] H. Lin, J. A. Rushing, S. J. Graves, S. Tanner, and E. Criswell, "Real time target tracking with binary sensor networks and parallel computing.," in *GrC*, pp. 112 – 117, 2006.

[87] L. Xiangqian, Z. Gang, and M. Xiaoli, "Target localization and tracking in noisy binary sensor networks with known spatial topology," *Wireless Communications and Mobile Computing*, vol. 9, no. 8, pp. 1028 – 1039, 2009.

[88] K. Ren, K. Zeng, and W. Lou, "Secure and fault-tolerant event boundary detection in wireless sensor networks," *IEEE transactions on wireless communications*, vol. 7, no. 1, 2008.

[89] K. Ren, K. Zeng, and W. Lou, "Secure and fault-tolerant event boundary detection in wireless sensor networks," *IEEE transactions on wireless communications*, vol. 7, no. 1, 2008.

[90] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *Information processing in sensor networks*, pp. 80 – 95, Springer, 2003.

[91] N. L. Zhang and L. Yan, "Independence of causal influence and clique tree propagation," *International Journal of Approximate Reasoning*, vol. 19, no. 3 - 4, pp. 335 – 349, 1998.

[92] F. V. Jensen and F. Jensen, "Optimal junction trees," in *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pp. 360 – 366, Morgan Kaufmann Publishers Inc., 1994.

[93] R. Kindermann and L. Snell, *Markov random fields and their applications.* 1980.

[94] H. Kunsch, S. Geman, A. Kehagias, *et al.*, "Hidden Markov random fields," *The annals of applied probability*, vol. 5, no. 3, pp. 577 – 602, 1995.

[95] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE transactions on medical imaging*, vol. 20, no. 1, pp. 45 – 57, 2001.

[96] S. A. Barker and P. J. W. Rayner, "Unsupervised image segmentation using Markov random field models," *Pattern Recognition*, vol. 33, no. 4, pp. 587 – 602, 2000.

[97] S. Z. Li, "Modeling image analysis problems using Markov random fields," *Stochastic processes: modeling and simulation*, vol. 20, no. 5, pp. 1 – 43, 2000.

[98] T. Szirányi, J. Zerubia, L. Czúni, D. Geldreich, and Z. Kato, "Image segmentation using Markov random field model in fully parallel cellular network architectures," *Real-Time Imaging*, vol. 6, no. 3, pp. 195 – 211, 2000.

[99] S. Pattem, S. Poduri, and B. Krishnamachari, "Energy-quality tradeoffs for target tracking in wireless sensor networks," in *Information processing in sensor networks*, pp. 32 – 46, Springer, 2003.

[100] J. A. Stankovic, T. E. Abdelzaher, C. Lu, L. Sha, and J. C. Hou, "Real-time communication and coordination in embedded sensor networks," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1002 – 1022, 2003.

[101] A. Jøsang, B. AlFayyadh, T. Grandison, M. AlZomai, and J. McNamara, "Security usability principles for vulnerability analysis and risk assessment," in *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*, pp. 269 – 278, Ieee, 2007.

[102] A. Ebert, N. Kerle, and A. Stein, "Urban social vulnerability assessment with physical proxies and spatial metrics derived from air-and spaceborne imagery and GIS data," *Natural hazards*, vol. 48, no. 2, pp. 275 – 294, 2009.

[103] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, *et al.*, "TinyOS: An operating system for sensor networks," *Ambient intelligence*, vol. 35, pp. 115 – 148, 2005.

[104] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29Th Annual IEEE International Conference on*, pp. 455 – 462, IEEE, 2004.

[105] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 563 – 579, 2005.

[106] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, "The liteos operating system: Towards unix-like abstractions for wireless sensor networks," in *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference On*, pp. 233 – 244, IEEE, 2008.

[107] P. Shpater, "Passive infrared motion detector and method," 10 April 2001. US Patent 6,215,399.

[108] C.-Y. Chao, S. Ashkenazi, S.-W. Huang, M. O'Donnell, and L. J. Guo, "High-frequency ultrasound sensors using polymer microring resonators," *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 54, no. 5, 2007.

[109] R. Narayanaswamy and O. S. Wolfbeis, *Optical sensors: Industrial environmental and diagnostic applications*, vol. 1. Springer Science & Business Media, 2003.

[110] M. Sabokrou, M. Fathy, and M. Hoseini, "Idsa: Intelligent distributed sensor activation algorithm for target tracking with wireless sensor network," *arXiv preprint arXiv:1506.00122*, 2015.

[111] A. Hawbani, X. Wang, S. Karmoshi, H. Kuhlani, A. Ghannami, A. Abudukelimu, and R. Ghoul, "Glt: Grouping based location tracking for object tracking sensor networks," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

[112] P. H. Chou and C. Park, "Energy-efficient platform designs for real-world wireless sensing applications," in *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005.*, pp. 913–920, Nov 2005.

[113] S. Faisal, N. Javaid, A. Javaid, M. A. Khan, S. H. Bouk, and Z. Khan, "Z-sep: Zonal-stable election protocol for wireless sensor networks," *arXiv preprint arXiv:1303.5364*, 2013.

[114] T. Banka, G. Tandon, and A. P. Jayasumana, "Zonal rumor routing for wireless sensor networks," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 2, pp. 562–567, IEEE, 2005.