Title	An Energy Efficient Recovery Mechanism for Tracking Systems
Author(s)	Seide, Germine
Citation	
Issue Date	2008-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4760
Rights	
Description	Supervisor:TAN Yasuo, 情報科学研究科, 修士

An Energy Efficient Recovery Mechanism for Tracking Systems

By Seide Germine

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Tan Yasuo

September, 2008

An Energy Efficient Recovery Mechanism for Tracking Systems

By Seide Germine (610203)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Tan Yasuo

and approved by
Professor Tan Yasuo
Professor Shinoda Yoichi
Associate Professor Defago Xavier

August, 2008 (Submitted)

Préface

The document is about my master thesis for obtaining the degree of master in computer science at Japan Advanced Institute of Science and Technology (JAIST). The thesis modeled a tracking system for monitoring mobile objects. This system can be applied to different type of applications such as tracking intruders in a field (a pets in firm or human being in a certain area) or following a moving car.

The model used in this document lays its foundation on previous work and inspired from current work in wireless sensor network. In this document the tracking scenario is based on prediction to select the future object location, and on a multi-layer architecture to ensure scalability and to achieve recovery in case of lost objects. By investigating the previous algorithms this work leads to a system in which the accuracy of tracking has been improved while showing a good performance of energy usage for object tracking in wireless sensor network.

I would like to warmly thank, first, my friends and ”Senpai” Amr Mostafa Mohamed Ashmawy from Tanaka Lab, Takashi Okada from Tan Lab, Peng Chao, and Nakata Junya from NICT Japan. The knowledge and know-how you taught me will remain valuable for me. For your time, your patience and kindness, many Thanks.

To my supervisor Professor Tan Yasuo, I would like to express my deep gratitude for his guidance and support. I would like to thank also Professors, Defago Xavier and Shinoda Yoichi for answering my questions and their help with my thesis. Special thanks go to the Japanese Government Scholarship (MONBUKAGAKUSHO) for supporting my study and living in Japan.

To my Heavenly father and my family: ”my ever lasting love”.

SEIDE Germine
JAIST
August 2008

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Ever since computers have emerged from laboratory to personal use, technological advances in microprocessor as well as micro-electro-mechanics and wireless communication made them more pervasive and thus more dependant. Scientists and engineers come up with applications feed by the needs to revolutionize living, work and even to interact with the physical environment. To meet such a goal, there is a demand for devices to be small enough to be embedded and cost-efficient in order to be widely deployed, programmed, used and maintained.

The advent of sensors nodes makes this possible. They allow a combination of sensing, data processing and computation capabilities into a single tiny device. Those sensor nodes can therefore measure physical quantity and return a signal that can be reported wirelessly. They can be battery-powered and they are capable to work for a long period of time. Because of those characteristics and their ability to self-organize themselves, gather, treat and send data relating to their located area, they are increasingly being used in applications in different areas.

In this thesis we study the problem of object tracking recently, applied in Wireless Sensor Network. Object tracking is used for application such as battlefield surveillance for military operations [4], Habitat monitoring [5], medical surveillance (Code Blue) [6] and application for the elderly people [7]. They are densely and randomly deployed and must report information to base station (Sinks) based on a certain frequency. This procedure will allow nodes to communicate to each other frequently and collaboratively accomplish their sensing tasks.

How to manage the data routed within the network, insure accurate detection and location when the object is moving randomly and its speed is changing along the way? Those questions raise an energy saving issue for networks used in such kind of applications.

1.2 Problem Statement

As we shown in previous paragraphs many tentative are being done in order to apply Wireless Sensor Networks application in real-world environment. In a near future more likewise applications are expected to provide access and information sharing in different places by using wireless devices. For that purpose sensor networks are meant to be widely deployed. However this new step of implementation cannot be done without any consequences. Indeed, beside some environment elements interference such as noise and weather, some constraints, directly related to its limitation such as power, synchronization, deployment and data routing, can be harmful in the implementation of such kind of system.

Among all the constraints the energy is one of the most important because this system is made to be applied in environment until now unaccessible. A lot has been written about finding a way to efficiently solve the energy saving issue in object tracking using Wireless Sensor Networks. Prior researches devoted to that problem implemented algorithms that focus on the two principal operations which are the monitoring and the reporting. Some in contrast based their interests on the communication cost while others proposed many topologies to approach problems like the object speed, data fusion, computing components, and so on.

For the last few years prediction based approach is being widely discussed. It used the information of the moving object to predict its next location. This method in order to save energy schedule the states (sensing / sleep) of the sensor nodes based on the object position in the sensing region covered by the deployed Wireless Sensor Network. This approach can be applied to any architecture. However, as the movement of the mobile object can be based on certain underlying events and/or randomness, and also some factors can influence the system, there is a need to provide a structure that can ensure an efficient energy saving sensing task without sacrificing the tracking accuracy of a mobile object.

Clustering, nevertheless, has been shown as a very efficient grouping technique to reduce the energy consumption problem in large scale wireless sensor network applications. **However there is not enough concern about methods or mechanism to approach the recovery phase of a tracking application when the object is out of the scope or range of the sensor**.

1.3 Research Objective

Baring that in mind, the main target of this thesis becomes obvious. It is to:

"Investigate the energy saving problem when object tracking application is being deployed on a large scale Wireless Sensor Networks".

1.4 Research Contributions

Implementing a mobile tracking application that can scale in case of architecture changes is not an easy task. In this thesis we will make use of an architecture that will serve as a backbone for a prediction-based tracking application over a large scale Wireless Sensor Networks. Thus the thesis has two main contributions, one is based on the architecture and the second one is about a prediction-based tracking application and a energy efficient recovery mechanism used in case of miss prediction. Thus, the proposed system will :

1. **Address the large scale deployment issue**

   The system is based on a multi-layer cluster based architecture in which all **level-(m-1)** clusters are grouped into a single **level-m** cluster. The system results into a structure of nested clusters as shown in chapter 4.

   We will show how this architecture will help reporting data from moving object while balancing the load and the main energy consumption as much as possible. We will prove that the proposed architecture can scale in case of adding nodes and handle the mobile object position with a margin of error adequate.

2. **Address the energy saving issue of the tracking system**

   As in previous approaches enumerated in chapter 3, this application uses the prediction-based mechanism to keep tracking the mobile object route. Moreover our approach is different from them in two ways:

   - In case of miss prediction we provide a recovery mechanism based on the proposed architecture and operate by the base station (sink node).
   - As frequent transmission is one of the key element when talking about energy consumption in object tracking application, we introduce a model named "sleep/predict/awake" between internal and boundary nodes. This model is expected to regulate the amount of data flow and the reporting frequency.

1.5 Thesis Organization

The thesis will be as follow:

- Chapter 2 will give a brief resume of wireless sensor network. It describes how the object tracking application is applied and enumerates examples for such kind of application. Finally it presents a list of terms that are frequently used in the document.

- Chapter 3 describes the prediction based method used in object tracking application to overcome handover problem. It surveys different works realized in that field and compares every energy scheme use for monitoring.

- Chapter 4 presents our application. It explains each steps followed to implement the proposed system through algorithms and flow charts. At last it presents the all system via a states diagram.

- Chapter 5 describes the experiment, which was carried out to evaluate the proposed approach. and explains each step of the simulation by using data charts. It includes also the design, scenarios, model, metrics, environment and tools used in the simulation environment.

- Chapter 6 provides graphs and explanation about the results obtained from the simulation. It includes also discussions and comparisons based on the experiment outputs.

- Chapter 7, finally draws an overall conclusion of this research and plans for future work.

Chapter 2

Object Tracking In Wireless Sensor Network

This section largely accentuates on the formal context of the topic: Object Tracking Application in Wireless Sensor Network.

2.1 State Of The Art

The projection[1] made by Gordon E. Moore that, "**by the year of 2002 more than billion wireless communication devices will be in used**", meets its expectation. Currently, the need for devices to be more accessible brings the wireless communication to a level until now unreachable. Meanwhile, the continued miniaturization of mobile computing devices and the extraordinary rise of processing power available, make wireless an integral part of residential, commercial and military computing applications.

Among other applications, many are meant to report information about a specific environment. This new evolutionary step in wireless communication requires devices to be smart enough to collect information in a real world sensing process. Sensors are then used. They can be self-organized and therefore offer the advantage to be densely deployed. Because of their ability to interact with the physical phenomena in real time and their pervasive attribute they can are being used in smart environments such as buildings, home, transportation system as illustrated in figure 2.1.

[1]Publication made in 19 April, 1965 "Cramming more components onto integrated circuits" in Electronics Magazine and coined as "Moore's law" around 1970 by the Caltech professor, VLSI pioneer, and entrepreneur Carver Mead.

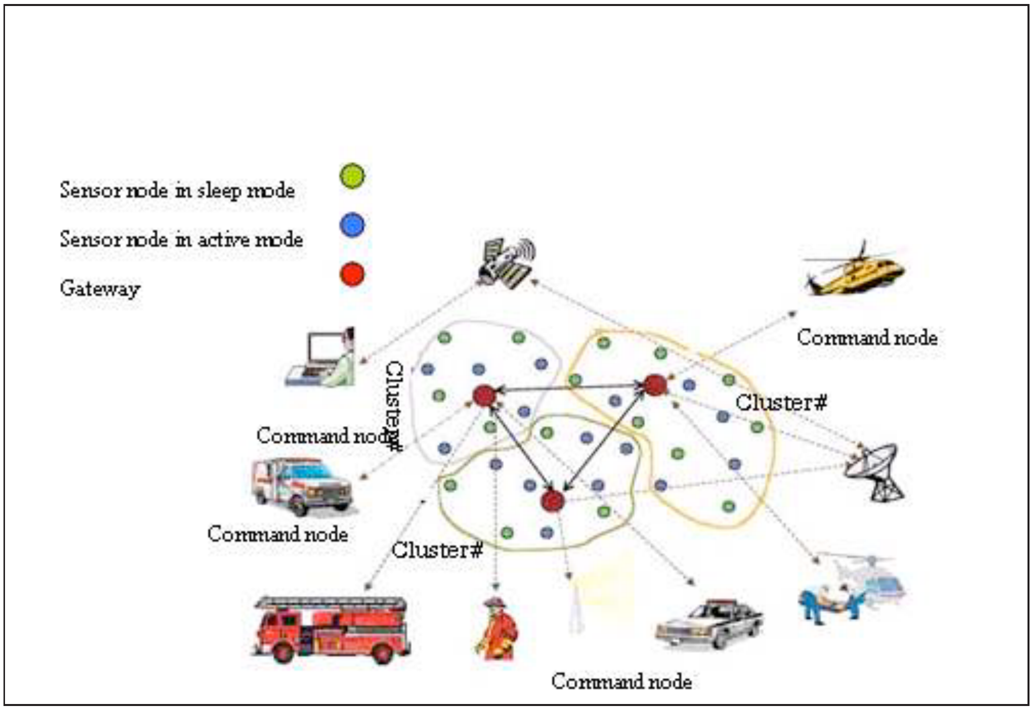


Figure 2.1: A Wireless Sensor Network

Some researches group them into three categories such as:

1. Event detection application in which the source detects an event and sends messages to the sink. An example of this category is an application that monitors the humidity rate in a room. Once the rate reach the threshold humidity, an alert is sent to the manager of the system for further assistance.

2. Applications that require periodic measurements. They describe any application that needs to send periodical data to the sink. Applications like CodeBlue that provide a wireless sensors network as a support for Medical Care, building automation, belong to this category.

3. Tracking application on which the thesis is based on, is an application in which the event producer, in this case the object, is mobile. As examples we can cite container tracking in the global supply chain and transportation application realized by the the Crossbow technology[2] team.

[2]Crossbow is a leading supplier of wireless sensor technology and inertial MEMS sensors for navigation and control.

2.2 Object Tracking

The problem of Object Tracking has raised many interests and has been the subject of many researches for the last few years. Given a mobile object to track, the system consists of detecting the precise object, locating its position and reporting the retrieved data from real world, for further assistance as illustrated in Figure 2.2.

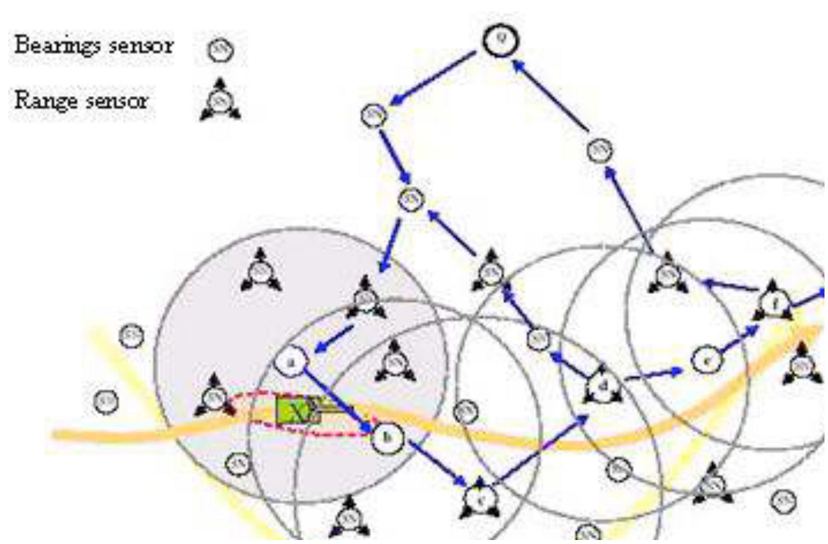


Figure 2.2: Object Tracking Sensor Networks Reproduced from [1]

To obtain interested information (such as location, speed, direction, size and shape) of the object, different sensors need to work in a collaborative manner. Those nodes must share their information. They must manage and appoint which nodes should sense, communicate, and receive information. They also have to decide how often this will need to be done. In the other side specific nodes, called base stations or gateways must act as an interface between the network and applications to issue command to the network and collect information from the field.

2.2.1 Tracking Scenario

As an object X is moving along the sensor field a number of activities are to be reported in the network.

- Discovery: a sensor node A detects the object X and initializes the tracking
- Monitoring: node A estimates the target location for a time **ti**. The position estimation of the object X can be done by triangulation method, grid or by bayesian estimation.
- Reporting: node reports data collected from the monitored region based on a certain frequency. Depend on the application, the reporting can be done with or without delay.
- Location: finding nodes that can handle future movements of the mobile object.

2.2.2 Approaches

In the literature, mainly 3 different methods have approached the problem of object tracking.

- A Tree-based approach that improves data collection and aggregation as described in [5-6].
- The authors of [7-8] come up with Cluster-based that uses multiple nodes instead of single one to get more precision. It allows to reduce duplicated messages
- Researchers in [7-8-9-11] propose a Prediction-based method. It can minimize the number of nodes participating in the tracking. By using this strategy, the network alerts appropriate sensor nodes to locate the target in the next tracking period.

2.2.3 Object Tracking Application

Tracking a mobile object is a very important task because it enables several important applications for:

Military such as

1. Security
   - Tracking enemy vehicles
   - Track and detect enemy intrusion over a battlefield instead using landmine.

2. Surveillance
   - Locate fire in a building
   - Detecting illegal border crossings.

Civilian such as

1. Health cares in which sensor nodes make less invasive patient monitoring and health care possible. [3]
2. Utilities such as electricity grid streetlights, and water municipals, wireless sensors offer a lower-cost method for collecting system health data to reduce energy usage and better manage resources.
3. Track current available doctor on a floor in case of an urgent intervention.
4. Retail space instrumentation to report current location of animal, object.
5. Traffic management to analyze flow, to detect accidents.

Hierarchical Architecture apply to Object Tracking

The figure 2.3 gives a preliminary vision of the proposed system. The dashed circles represent the range of each sensor. Sensors are grouped into a clusters lead by cluster heads. The hierarchical update is made by the cluster heads within the network until they report to the sink where the observer can plan for problem requiring drastic action.

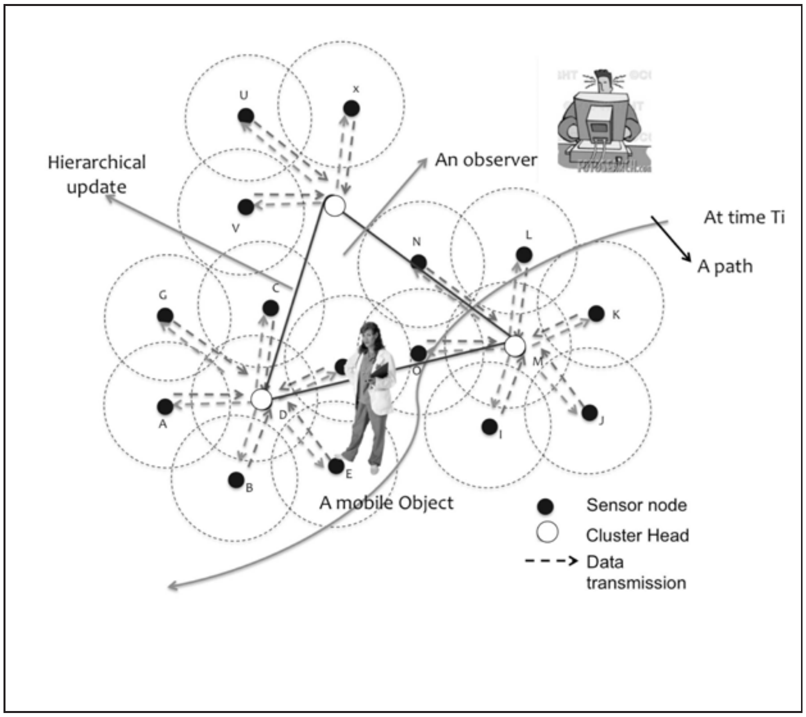


Figure 2.3: Object Tracking Sensor Networks based on a hierarchical architecture

2.3 Terminology

In the following section we define some of the key terms encountered in this document, the definitions below will be used in many parts of the document when describing object-tracking system.

- Sensor Node.
  A sensor node is a node in a wireless sensor network. It contains low power-paging channel in the physic layer and support the four di erent radio modes (transmit, receive, idle and sleep). This channel keeps the node running at fully duty, allows sensor node to communicate and awake each other. When a group is spread out over a sensing area they are responsible to monitor the immediate environment and report their data.
- Wireless sensor network
  Large number of heterogeneous sensor devices spread over a large field to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at di erent locations.
- Cluster head or gateway

They act on behalf the sensing nodes. They collect the data sent by each node in their cluster, compress it and transmit the aggregated data to the base station.

- Sink or Base Station
  A wireless communication station installed at a fixed location and used to communicate to the nodes. They support requests and collect data reported by the sensor nodes from the monitored region.

- Mobile object
  Entity that generates various stimuli for sensor nodes.

- Object Tracking
  Track an object is to locate the object to some accuracy in each tracking period.

- Prediction
  Anticipation of future movements of object based on initial information. Thus, only the sensor node expected to discover the object will be activated.

- Current node
  It is a node that contains the mobile object in its range.

- Boundary node
  Sensor nodes that are located in the boundary of its cluster.

- Internal Node **Sensor nodes that are located inside its cluster.**

- Scalability
  In wireless sensor application the network should scale from ten to thousands or millions of sensor nodes. This needs automatic-configuration, maintenance, upgrading of individual devices.

Chapter 3

Prediction-Based Method

This part of the document describes the prediction method. We will present the main steps of this method and we will explain related works that have been already done in that field.

3.1 The Context

Prediction method is being used for a long time from now in mobile computing environments to improve system performance. Among others examples we can cite the user search space prediction that reduces the paging overhead in cellular environment [6].

3.2 The Prediction Method in Object tracking

In object tracking application each position of the mobile object is to be monitored and recorded. Like many wireless sensor network applications, object tracking must be reliable, accurate and precise, but unlike them, each task requires a lot of energy to be consumed for both transmission, computation and also location. Thus the prediction method whom steps are in figure 3.1 will help reducing the number of awakened sensor nodes to participate in the tracking application. By predicting each future movement of the mobile object, it can be feasible to reduce the energy consumption, increase the efficiency, provide a robust and reliable system.

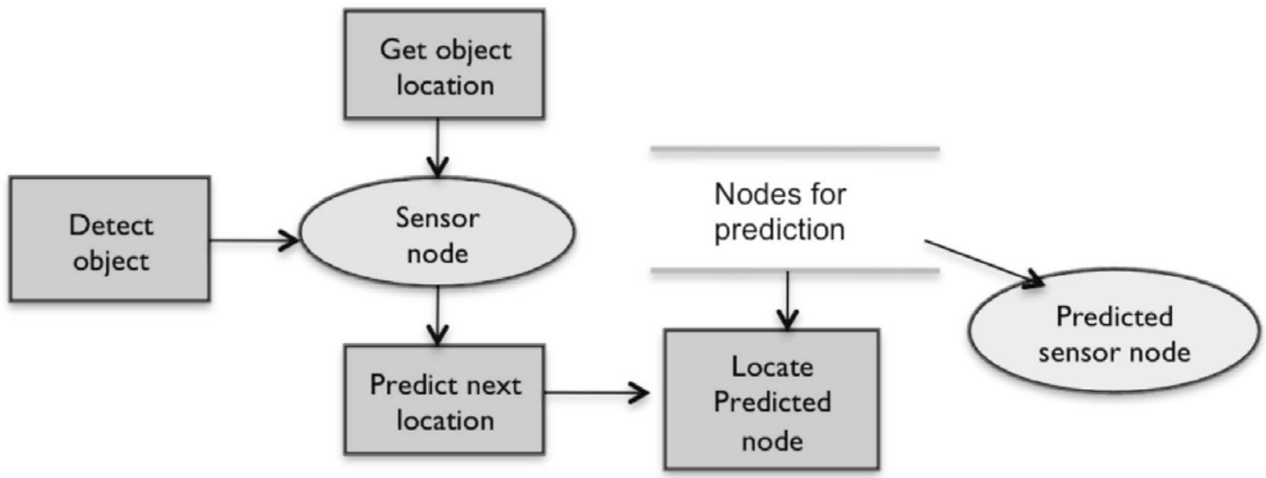


Figure 3.1: Object Tracking Flow Chart

3.3 Prediction-Method Phases

3.3.1 Phase 1 - Monitoring

In the monitoring phase nodes detect the mobile object and get information about its location. To address the energy optimization issue in object tracking system, some monitoring schemes have been discussed in previous literature. Authors in [10] describe three basic energy saving schemes as described in table 4.3.2.

1. Naïve monitoring in which all the sensors are active to monitor the sensing area all the time. This scheme is not tunable to the application requirements for energy saving.

2. Scheduling monitoring addresses the fact that the network does not need to report information all the time. It is assumed that all, the nodes and base station are well synchronized. All the nodes can turn to sleep and wake up only when it is time to monitor their detection areas and report sensed results. In this network all the nodes need to be activated for **X** second then go to sleep for a period of **T-X** sceonds. This monitoring will last for the entire network operation period. The advantage of this method is to keep more node in sleeping mode as long as they can hence save energy.

3. Continuous monitoring only activates sensor node that has the object in its range. It will monitor the object until it enters its neighboring cell. The handoff happens when an object reaches the detection area boundary. It then wakes up the destination node **W** seconds before it enters its range thus, it can handle the next location of the moving object. This scheme involves only **one** sensor node to monitor the object while the others are turn to sleep ans save energy. To ensure no missing report, the active sensor has to stay awake while the object is in its detection area.

Schemes	**Nodes Involved**	**Continuous?**	**Energy Consumption**
Naïve	All($= S$)	Yes	$E_{wake}\ \ TS\ \ S$
Scheduling monitoring(SM	All($= S$)	No	$(E_{wake}\ \ X + E_{sleep}\ \ (T - S))\ \ \frac{TS}{T}\ \ S$
Continuous Monitoring(CM)	One for each object	Yes	$E_{wake}\ \ TS + X + E_{sleep}\ \ TS\ \ (S - 1)$
Ideal	One for each object	No	$E_{wake}\ \ \frac{TS}{T}\ \ X + E_{sleep}\ \ (TS\ \ S - \frac{TS}{T})\ \ X$

Table 3.1: Analytical evaluation for energy saving schemes

In Table [3.1] we summarize the studies made on monitoring schemes. The table shows an Analytical evaluation for energy saving schemes in which:

- E_{wake} denotes the energy consumption rate per second at a sensor node.
- E_{sleep} denotes the energy consumption rate in sleeping mode.
- T represents the running time for the application.
- X denotes the number of sensor nodes participating in the tracking.
- S denotes number of sensor nodes in the networks.
- $\frac{TS}{T}$ Number of time transmission happens.

3.3.2 Phase 2 - Prediction

When the target enters the field its location is being reported to the sink node. Based on that information, prediction values can be calculated. Many methods can be used for prediction. Recent approach brings to light methods such as **particle filter, movement history in which the past states recorded**.

3.3.3 Phase 3 - Localization

Once the predicted value is issued the cluster head need to locate the node that will report the next location of the target. To select the next node to wake up a particle filter can be used as well as geometric (coordinate) and symbolic model (grid, sensor cell and triangle) as described in [6,7].

3.4 Pro & Cons about Prediction-based Method

1. It can minimize the number of nodes participating in the tracking.
2. Trades computation for communication
3. Different prediction models, wake up mechanisms and recovery mechanisms will affect the system performance
4. Works well if one can tolerate
   - small amount of errors in predictions.
   - some latency in generating prediction models.

3.5 Related Work

3.5.1 Prediction-Based Energy Saving (PES)

This work relates strategies for saving energy in application such as Object Tracking. Its objective is to achieve high-energy efficiency by minimizing the energy dissipation in the micro-controller unit (MCU) and sensors components.

PES [6] approaches the ideal scheme and minimizes both the sampling frequency and the number of nodes involved in the tracking. The scheme consists of a prediction model, a wake up mechanism and recovery mechanism. The current node[1] performs sensing for **x** seconds and report to the base station. Before going back to sleep it predicts the object movement for the next period and informs the nodes target [2] located in the predicted area. After sleeping for **T-X** seconds all, the target nodes and the current node, wake up together to track the object. The new current node will then repeat the same process while the other nodes go to sleep.

3.5.2 Dual Prediction-Based Reporting (DPR)

This algorithm investigates the prediction-based approaches for performing energy efficient reporting in object tracking. It used a dual prediction reporting approach to reduce the energy consumption of radio components. In doing so it can minimize the number of long distance transmission between sensor and the base station with a reasonable overhead.

The DPR[7] has two main components a localization model and a prediction model that analyzes the history of the moving object and estimates its next location. In this mechanism, the prediction model is deployed at both the sensor nodes and the base station. They both used the same historical data and make the same prediction. If their predictions match the sensor nodes can avoid transmitting to the base station. Otherwise the sensor nodes have to correct the base station by sending the real object movement. Thus the sensor nodes can make intelligent decisions about whether or not to send updates of objects movement states to the base station and thus can save energy.

To allow the sensor nodes which never saw the object to make predictions, the movement history has to be passed between sensor nodes as the object moves along the sensing area. This method makes a trade off between two concepts which are: multi-hop/long range transmission, between sensor node and base station and one-hop/short range communication among neighbor sensor node.

3.5.3 Prediction-Based Monitoring in Sensor Networks(PREMON)

This work[14] mainly focuses on the reporting aspect in the tracking system. Its goal is to prevent a sensor node from unnecessarily transmitting all the readings that can

[1] the node that has the object in its range
[2] Target nodes used to locate the next location of the object

be successfully predicted by the base station, thereby saving energy.

Sensor nodes detect the mobile object inside their detection regions only monitor its state and report the readings to the base station. This reporting process is done based on a reporting frequency specified by the application. Nodes that do not have mobile object in their detection area do not report to the base station. Node does not make any prediction, thus, no exchange of historical data is necessary.

Once received the data, the base station makes prediction and then transmits it to the corresponding sensor node. If the prediction is not correct the sensor nodes do not need to report their readings, otherwise they have to update the correct readings to the base station.

3.5.4 Distributed Prediction Tracking(DPT)

DPT[16] is a distributed predictive tracking algorithm that can accurately track mobile object. This algorithm aim is to minimize the communication between sensor nodes and control overheads while using cluster based approach to achieve scalability.

DPT distinguishes two kinds of nodes. The border nodes are required to locate the object when it enters the sensing field. The non-border nodes are keeping in hibernation mode until it is asked to start the sensing task.

When the object is in detected the algorithm requires tree sensors nodes to sense jointly the relative position of the moving object. Each of them reports its data to the cluster head. It will aggregate this information and get the present location of the object. The cluster will then estimate the next location of the mobile object after a period of time. It is then inform the most likely downstream cluster heads to keep tracking the object.

The protocol to be robust against node failures presents a recovery scheme. This scheme uses the high beam and the normal beam available for a sensor. It allows sensors around the predicted area to switch from one to another in order to locate the missing object. This scenario helps to quickly recover from node failures with the used of little additional energy.

3.6 Fallbacks from the related work

Obviously these approaches bring a new network paradigm for mobile object tracking in wireless sensor networks; however, some analysis show that

Prediction

1. In PES and DPR, because
   - One sensor node detects and reports its own discovery to its neighbor.
   - Node makes its own prediction and localization from its history.

   More energy is to be consumed to transmit that history from one node to another when the target moves in the range of the latter.

2. Another key problem to point out is the fact that information (history) is sent to predicted node whether or not it can detect the mobile object. This latter comes up also with an extra energy waste problem.
3. No consideration for the case of failure nodes before reporting.

Recovery

1. In PES the recovery mechanism involves too many nodes which makes it not energy efficient enough.
2. As for the recovery process in DPT we consider the procedure excessive because it uses the maximum communication range of the sensor to overcome missing object which consumes too much energy for only a single task.

Approaches Comparison

Name	Monitoring Scheme	Approaches	Recovery
Dual Prediction(DPT)	CM	Prediction is made in both sensor and sink nodes	Dual prediction to prevent missed object
Prediction-based energy saving (PES)	Ideal	Prediction is made by sensor nodes	Wake up all sensor nodes in the network to locate missed object
Prediction-based monitoring in sensor networks (PREMON)	SM	Prediction is made cluster Head	Wake up all the neighbor nodes of the predicted location to locate missed object
Distributed prediction tracking(DPT)	Triangulation method	Prediction is made by cluster head	Use of the extreme limit of sensor node range in case of missed object
The proposed Tracking System	SM done by cluster group	Prediction is made by cluster Head and Sink Node	Use of the upper layer to locate missed object

Table 3.2: Approaches comparison

Chapter 4

Proposed Energy Efficient Tracking System

This part of the document presents the main objective of the thesis. It describes the components of the proposed system and the steps used to evaluate the approach it is based on.

4.1 Object Tracking Application

The main purpose of implementing an Object tracking application using wireless sensor networks is to monitor the mobility of an object i.e. be able to detect, track its positions, and possibly its speed and direction. In this application information about the mobile object motion is being reported from the sensing field.

4.1.1 Assumptions

To realize the Object tracking application we pointed out some assumptions

- Sensor nodes are statically and randomly distributed to cover all the sensing area.
- The approximate geographical boundary of the monitored field is known to the application
- Each node can determine its location.

- The base station has full knowledge about the network topology.
- Sensor nodes are enabled for sensing and communication by the sensor components and the RF radio component respectively.
- Both the cluster heads and the sink (being special sensor nodes) are additionally enabled for computation by the micro-controller unit (MCU).

4.1.2 The Mobile Object

The mobile object can be any object that can enter the sensing area at a random boundary and time. It will move continuously until it goes out of the sensing area. In that application we focus on two kind of objects; first humans who can execute a pre-defined route with, more or less a fixed speed and second vehicles, whose speed can change at a undetermined period of time. For the purpose in hand, we assume that the objects are identifiable[1], so that sensor nodes (see Fig 4.1.) are able to store mobile objects' history, to be used later to predict its future movement.

4.1.3 The Sensor Node

This section gives a brief description of a simple example of a sensor node. Thus figure 4.1 represents a MICAz mote from Crossbow group. On the market, Crossbow Technology[23] comes as a leader in supplying sensor systems. Among all their products the Mote-KIT2400-Micaz as shown in fig 4.1 can be used to implement the system in a real world environment. The choice of this kit is based on the fact that it supports the basics functions of the system and provide the important initial raw data for evaluation and manipulation.

[1]Electronic tags or a pre-embedded object code table defined in the sensor nodes are some techniques that can be used to identify the tracked objects.

Figure 4.1: MPR2400-MICAz (Crossbow, 2005)

4.1.4 Topologies

Considering the case that the object may vary its direction we still have two topologies to consider.

1. Straight lines

   This case is simple based on the fact there is no deviation in the movement of the object. It follows a line in which each next point is directly deduced from the prediction method itself.

2. None straight line

   This method is more complicated because it is based on some deviations from the mobile object current direction. We consider the fact that the mobile object does not change its path abruptly then its next position can be predicted with an error margin by the linear predictor. This case is actually a composition of two or more paths of the first case. Both cases are showed in figure 4.2.

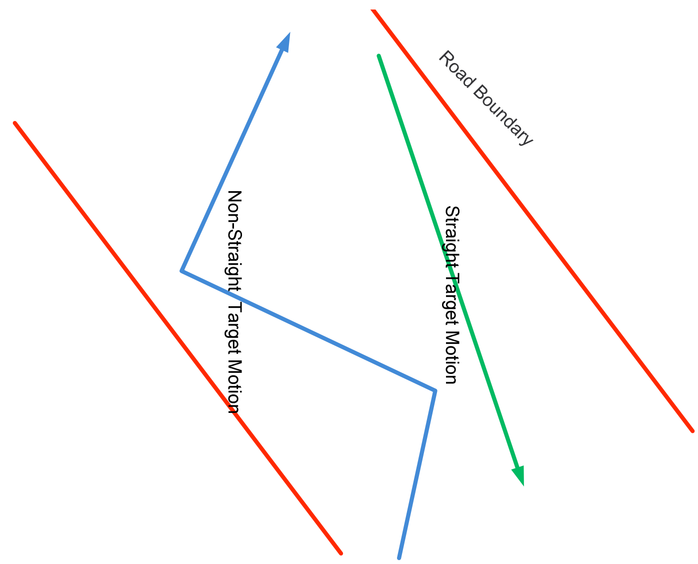


Figure 4.2: Topologies

4.2 Problem Statement

Tracking application as other applications in wireless sensor network are meant to be deployed on a large scale, therefore some issues such as scalability, accuracy and power consumption need to be considered. Addressing those issues for object tracking in wireless sensor network is theoretically twofold:

1. How to expand the network over a large area?
   - to handle the increased nodes number
   - to protect the system against nodes failure
2. How to handle the tracking application in such conditions?
   - to manage the communication through the network
   - to minimize the energy consumption during communications between nodes

4.3 Proposed Solution

To resolve the problem described above we need to define first an architecture that can support any network change and second a tracking mechanism that can achieve, based on that architecture, low energy for the application.

4.3.1 Network Architecture

Settings

- Sensor initialization and identification
  The network is organized so that each node knows its ID and its location in the sensing field. At this step all nodes belong to **level-0** in the architecture.

- Clustering
  After the deployment of sensor nodes they will be grouped into cluster.

- Election
  Nodes which are in the same range elect a cluster head[2] and restarts clustering in higher level. This process will continue until it reaches the sink node, thus all **level-(m-1)** clusters are grouped into a single **level-m** cluster. In this architecture each level has its own tasks and communicates to others through its elected leader (cluster head) it is shown in figure 4.3.

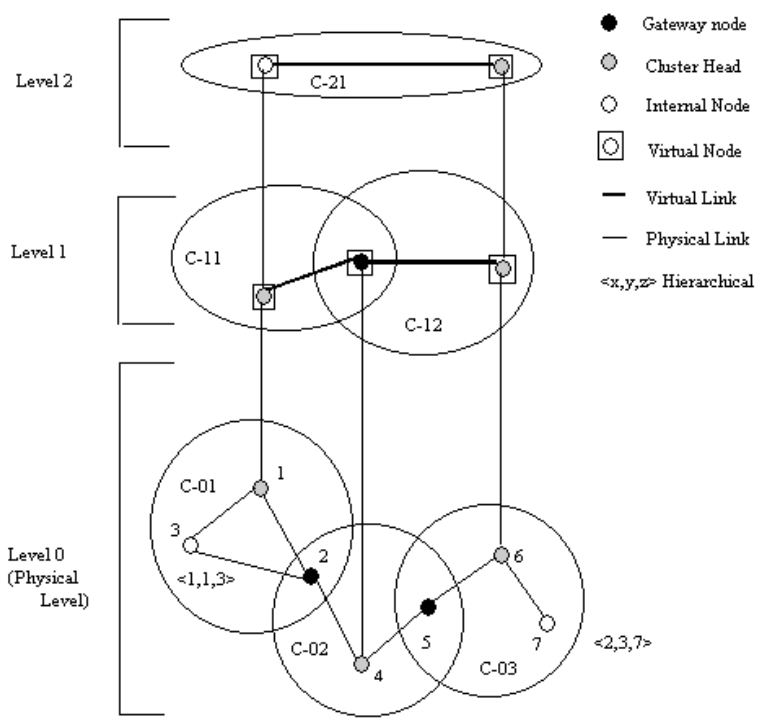


Figure 4.3: Multi-tier architecture from [2]

[2]Determine the cluster head is out of the scope of the thesis.

Advantages

This multi-tier architecture is proposed to solve the problem of large scale deployment. In this network, nodes will report their data to the cluster head while the mobile object is in their range otherwise keep silent. By using this architecture the transmission will be done within one hop to reduce the number of packets to be transferred. Therefore less energy will be consumed to execute the tasks. This architecture although it facilitates the expansion of the network, it will permit the recovery process in case of node failure and keep the reporting rate low and only restricted within the cluster.

4.3.2 Prediction Strategy

Step 1: Object detection and monitoring

The monitoring phase is scheduled after the network configuration. In this thesis we will consider the case in which the object is detected among a cluster range. The node that detects first the object will report its discovery to its cluster head within one hop and during a period of time **x**.

Figure 4.4 illustrates the step 1 of the proposed method. It indicates the mobile object current position and the range of the sensor that initiates the detection. The dashed circle represents the cluster that will handle the next operations for the tracking application.

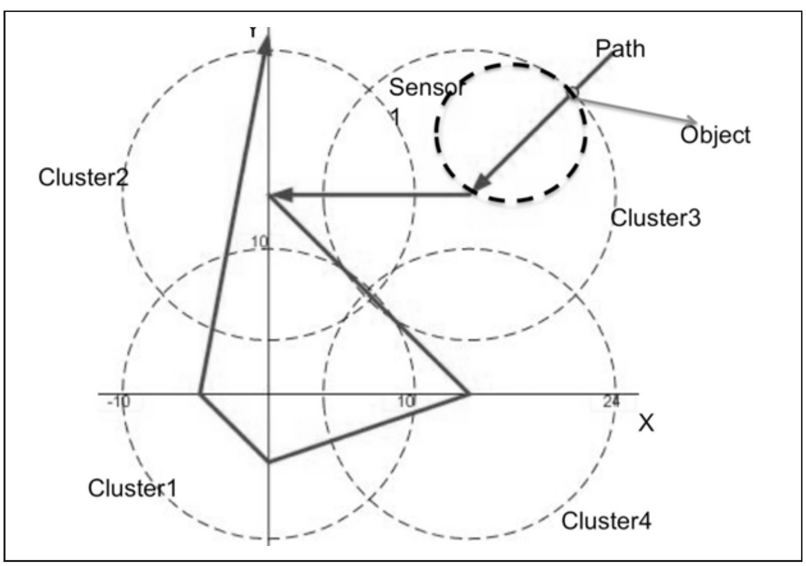


Figure 4.4: Detection

Algorithm for Sensor Node

```
loop
    Monitoring
    if Object Detected in range then
        Report
    else
        if sensor is boundary then
            Go to sleep mode
        end if
    end if
    Delay
end loop
```

Table 4.1: Sensor Node Algorithm

Step 2: Prediction

The cluster head will keep a history of all previous location of the object while it is in its range. When it approaches the boundary of the cluster range the prediction value is then calculated. The next section presents the formulas used to compute the prediction value. Based on obtained predictive values figure 4.5 indicates the predicting area of the node that will handle the next tracking.

Linear Predictor

The next location of the object is computed by using the linear predictor. The linear predictor uses only the previous two positions to linearly predict the next location by using the following equations. We can estimate the speed, the direction and the next location of the object by using the following equations.

- Speed of the moving object

$$v = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \tag{4.1}$$

- Direction of the mobile object

$$\theta = \arccos \frac{t_i - t_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \tag{4.2}$$

- Next location of the mobile object

$$x_{i+1} = x_i + vt\cos(\theta) \tag{4.3}$$

$$y_{i+1} = y_i + vt\sin(\theta) \tag{4.4}$$

X	Position of the mobile object on X-axis
Y	Position of the mobile object on Y-axis
t	Time of motion detection

Table 4.2: Linear Predictor

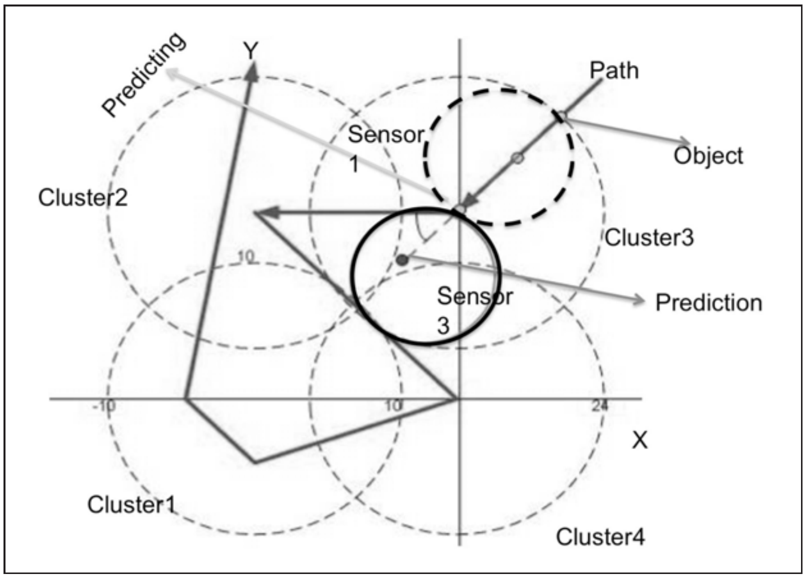


Figure 4.5: Prediction

Algorithm for Cluster Head

```
listen till timeout or receive
if receive (CurrentPosition, SenderNode)//NewThread then
    if LastPosition is defined then
        PredictedPosition   2(CurrentPosition) − LastPosition
        acitveSN[ ]   0
        cancel previously waiting Threads
        Locate
        if in range then
            send wake up
        else
            send recover
        end if
    end if
    LastPosition   CurrentPosition
    acitveSN[SenderNode]   1
    start timer
    continue to listen
else
    Timeout
    PredictedPosition   2(PredictedPosition) − LastPosition
    acitveSN[ ]   0
    cancel previously waiting Threads
    Locate
    if in range then
        send wake up
    else
        send recover
    end if
    PredictedPosition   undefined
end if
```

Table 4.3: Cluster Head Algorithm

4.3.3 Synchronization

The synchronization problem can occur in three different situations all shown in figure 4.6.

1. Single report

   One sensor reports its first detection and stops. In this case no prediction can be done until the other sensors start reporting.

2. Unordered report

   - One sensor reports the first detection of position x_1 at time t_1
   - Another sensor reports its detection of position x_2 at time t_2
   - cluster head receives the second report before the first.
   - cluster head assumes reverse motion from position x_2 to x_1 such that $t_1 < t_2$ and $x_1 = x_2$.

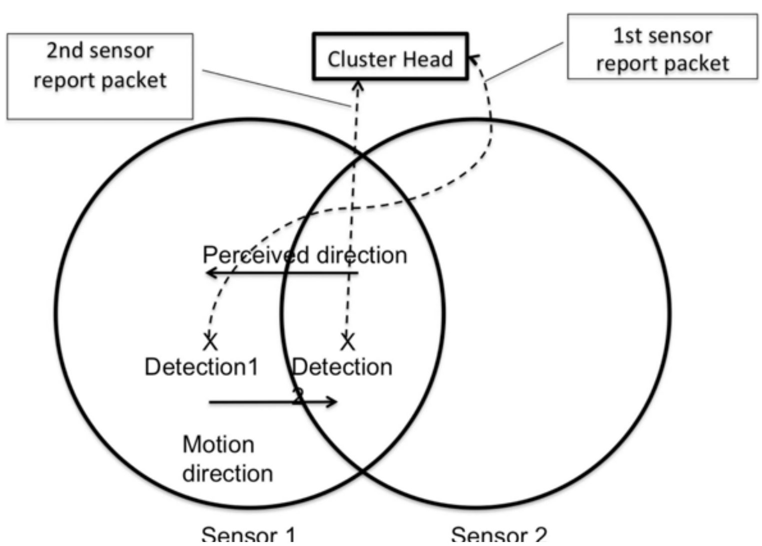


Figure 4.6: Synchronisation

3. Multiple reports

   Two or three different sensors report simultaneously within a certain time frame.

To resolve this problem, we propose a simple synchronization mechanism inspired from the consensus problem for out of order reporting from nodes. For the last situation their data averaging can be useful.

1. Averaging can be used to provide an easy solution for the last situation with minimum accuracy loss.

2. For the second situation delay the direction calculation to give opportunity for any of the reporting sensors to send two consecutive(making use of the guaranteed FIFO ordered channel) reports, thus ignoring other in between reports.

4.3.4 Recovery Mechanism

Miss Prediction

Like any other prediction, the prediction we proposed is associated with miss prediction and error. Figure 4.7 shows the recovery phase. The recovery mechanism is initiated when the cluster head cannot locate the next position of the mobile object whether because sensor fails to wake up, a bad prediction has been made or predicted outside the range. From figure 4.7

- the current position of the mobile object is identified
- the circle represents the sensor range of the detected sensor node
- the dashed circle represents the cluster range.
- the predicted position is shown in cluster 2 while the object is leaving the first sensor.
- the position where the sensor failed to wake up causing cluster head to initiate recovery is also indicated.
- the higher level predicted position is shown in cluster 2.

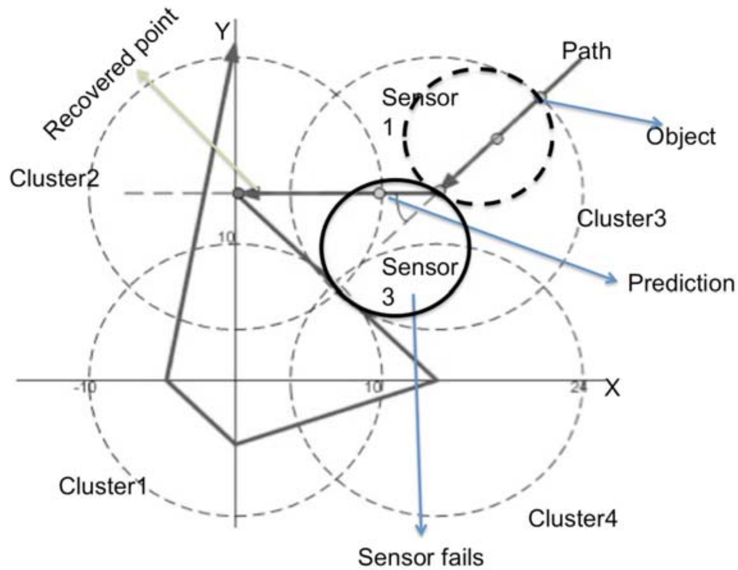


Figure 4.7: Recovery

Solution

In either of the pre-cited cases, the cluster head times out, informs the sink node to start the recovery by estimating the next location based on the reported history and by taking advantages of its additional info in its higher level in the hierarchy. The sink predicts faster target motion along the same direction and awake the next node(s). This mechanism is a backup for miss prediction without sacrificing much energy consumption and providing resoanble accuracy.

4.4 System flow chart

4.4.1 Mobile object

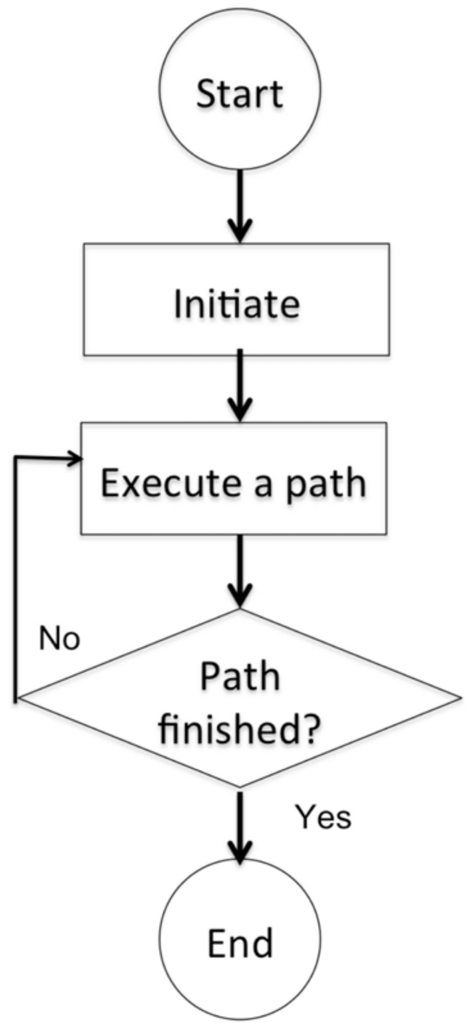


Figure 4.8: The object flow chart

4.4.2 Sensor Node

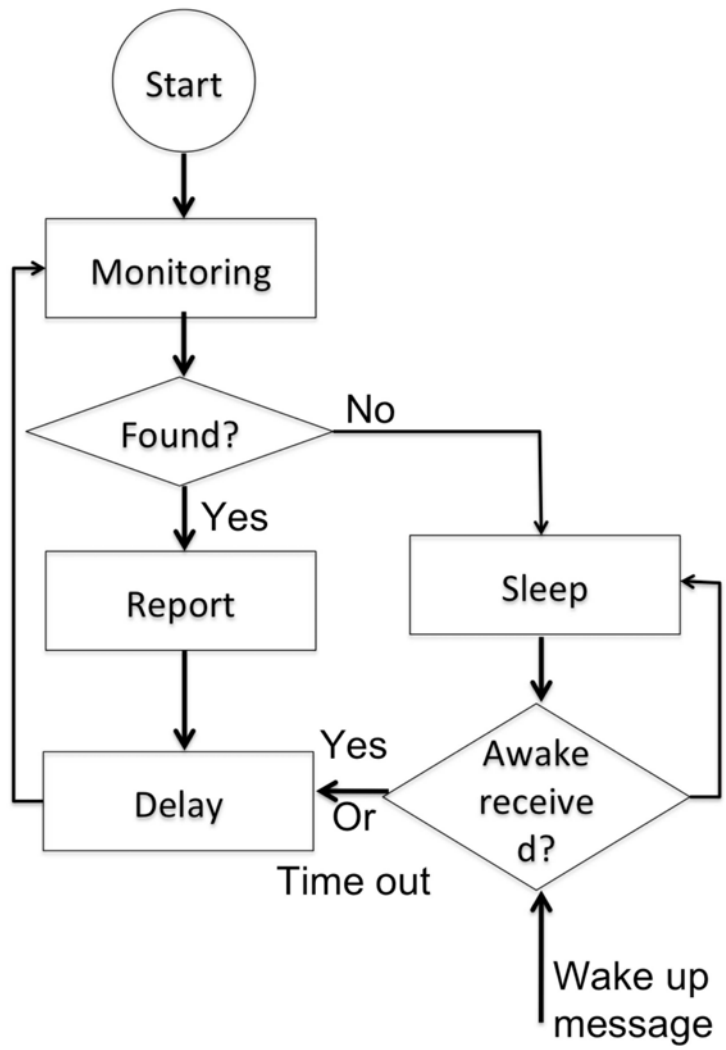


Figure 4.9: Sensor node flow chart

4.4.3 Cluster head

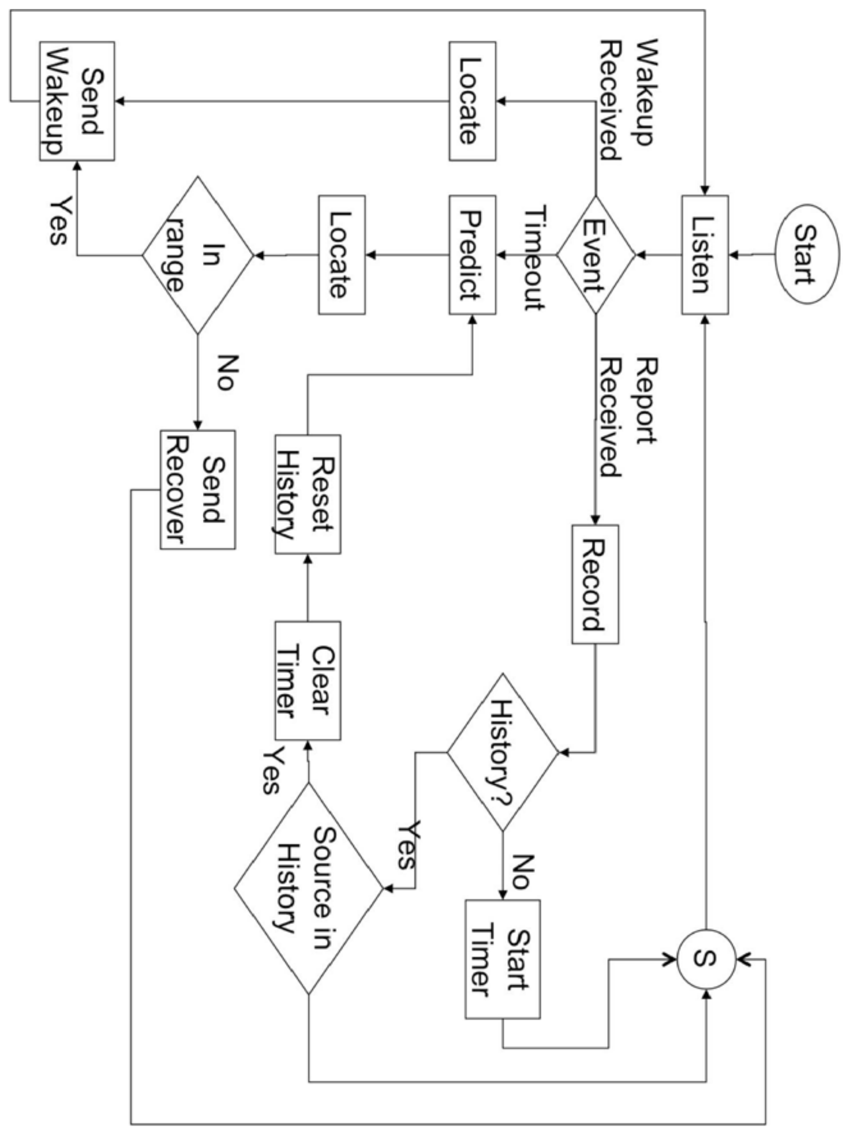


Figure 4.10: Cluster Head flow chart

4.4.4 Sink node

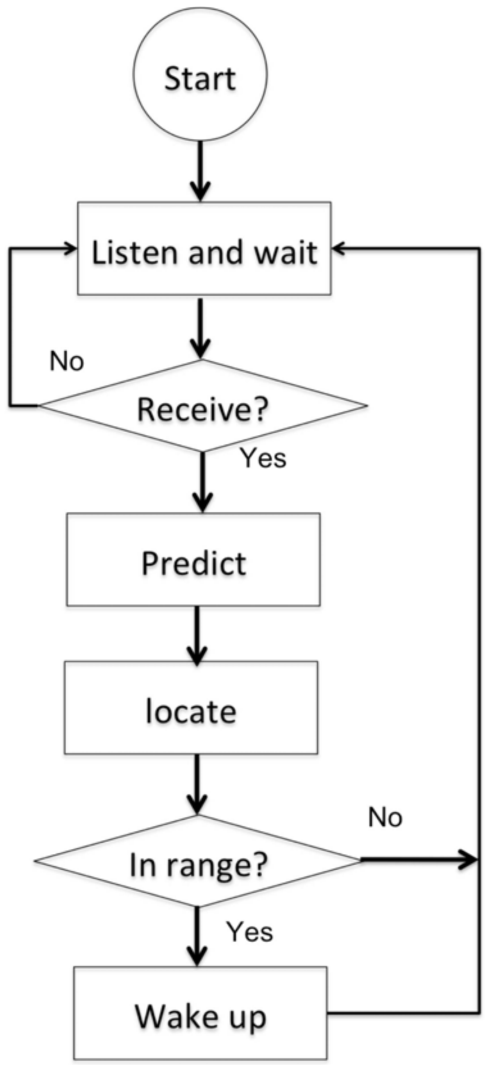


Figure 4.11: Sink node flow chart

4.4.5 The System States Diagram

- Inner nodes keep sleeping until they receive a wake up message based on the predicted position.
- If any node fails to wake up the recovery process will be initiated after a time out by the cluster head.
- The sink based on its full information about the network topology will estimate its own prediction and locate the appropriate cluster head within which the predicted position lies.
- That cluster head becomes the new leader and will wake up the appropriate local node to handle the next tracking.

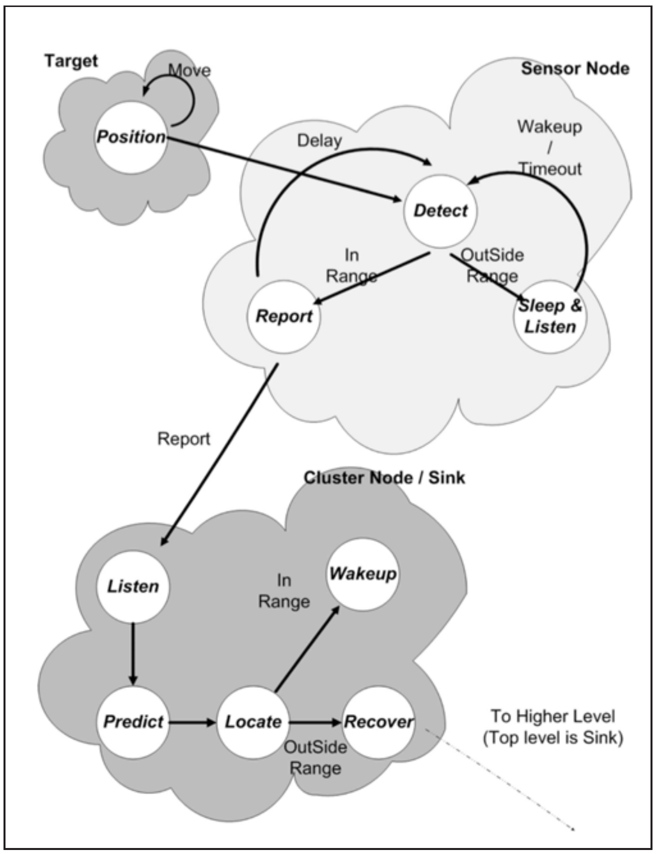


Figure 4.12: States Diagram of the application

Chapter 5

Experiment

It is di cult to directly deduce the e ciency of any approach without performing fair experiments using varying simulation parameters. This chapter presents the experiment conducted to determine whether or not the approach can improve the energy saving issue. It describes the tools, modules and metrics used for the simulation.

5.1 Environment

There exists severals environments that can be targeted to simulate an object tracking application. In our case we will consider a 2-D flat area in which the sensors nodes are logically grouped in a hierarchy tree. The area is a planar field without hills or deeps. Over the sensing area, sensors and clusters are symmetrically, regularly centered in circles without any gaps between them see figure 5.1.

Accordingly all points in the monitored field can be categorized into 4 different kinds (see figure 5.1) depending on the number of sensors covering.

- 0 sensor: this case refers to a position that is outside the range of all the local sensor nodes but still inside the range of a boundary cluster. It is represented by the "circle" in the figure.
- 1 sensor: it is located in the middle of the sensor range away from its neighbors overlap. It is represented by the "rectangle" in the figure.
- 2 sensors: exactly two sensors overlap covering that the point. It is represented by the "star" in the figure.

- 4 sensors: this is only the point of intersection of four adjacent sensor ranges. It is not region as the previous cases. The sensors can belong to the same cluster or each to a different cluster or some 2 sensors from one cluster and the other from the adjacent cluster. It is represented by an "x".

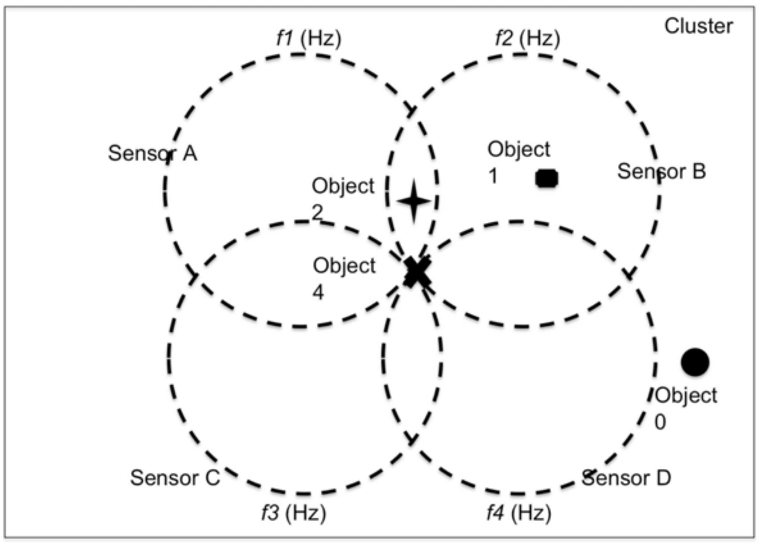


Figure 5.1: Method to cover the sensing area

5.2 Experiment Setup

We implemented a simulator that generates the workload data of the object tracking application. It is an event-driven application, implemented in C language over a set of Embedded Computers AR2000 Series with the features summed up in table 5.1.

Name	Description
PC	Embedded Computers AR2000 Series Model 110
Operating System	FreeBSD 6.0
Processor	Intel Pentium M
Card Supported	CompactFlash, CardBus and PC Card Standard
Connectors	Audio, USB 2.0, IEEE 1394
LAN	2xLAN (10/100BASE-T), PXE enabled
Power Supply	AC & DC (16V to 24V)are supported

Table 5.1: Functions and Features for Embedded Computer AR2000 Series

5.3 Path for the moving object

To evaluate the tracking method we define a path containing an example of the latter topology described in chapter 4. The mobile object while executing the path can be detected by any sensor in the network. We fixed a start point at which the object must start moving. Then the object keep moving along the path crossing specified points. The motion direction does not change in between the points but only at each point when reaching it. In the other hand the speed between points can change randomly.

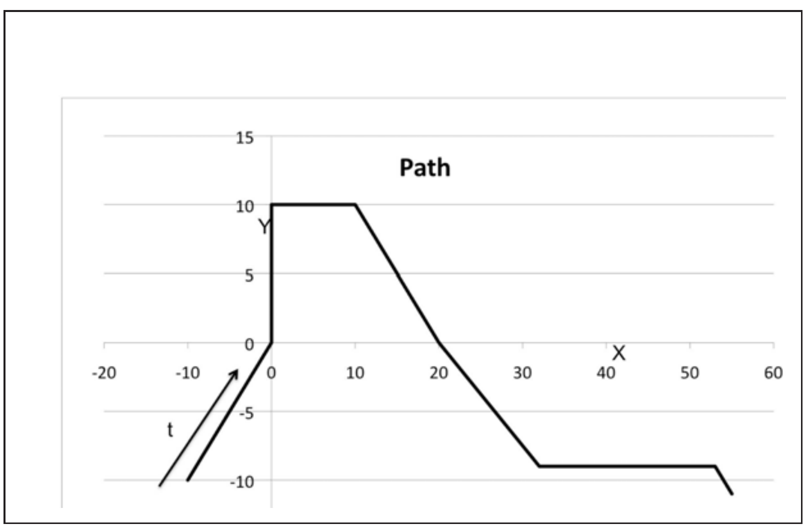


Figure 5.2: Path for a mobile object

5.4 Network Topology

Once deployed in the sensing field the sensor nodes, as described in chapter 4, will be clustered and arranged into an hierarchy. The result network shown in figure 5.4) is a 3 layer architecture in which each embedded machine is emulated as a cluster and each cluster is a group of 4 sensor nodes.

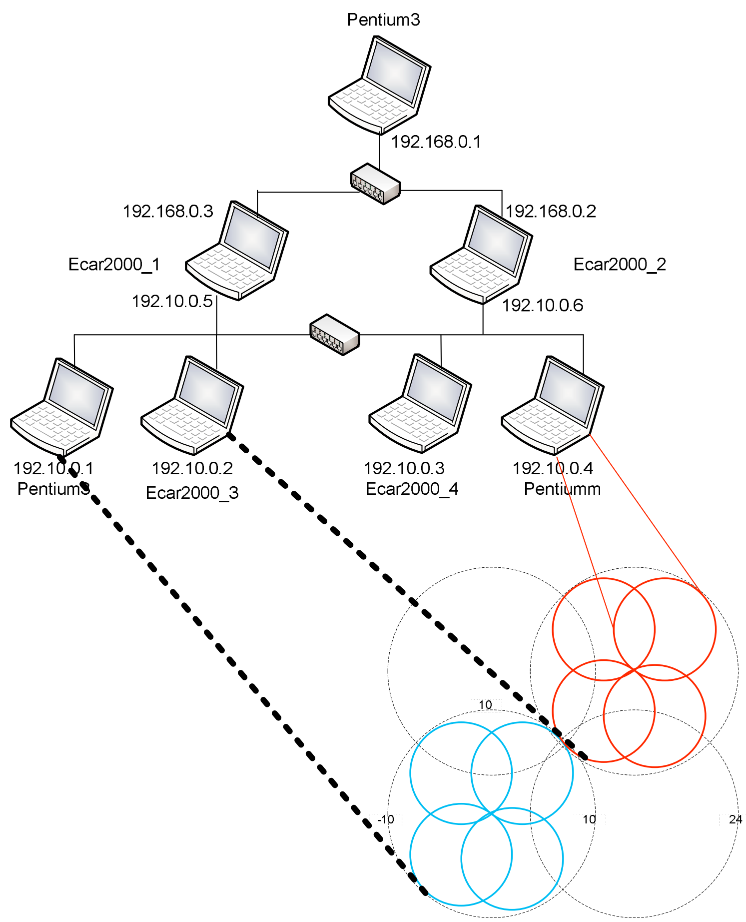


Figure 5.3: Sensor Network

In table 5.2 we described all the paramters used to setup the experiment eironment.

Name	Description	Values
SA	Sensing Area	200x200m
S	Sink node	1
SN	Number of sensor node	100
CH	Number of cluster head	25
SR	Sensor range	25cm
MO	The mobile object	1
T	tracing time	187mm
P	Path size	35m
O_s	Object steps	100

Table 5.2: Parameters for the simulation model

5.5 Programming

The Object tracking application has been written by using C language and compile with gcc. The choice of the language is based on the fact that it will be implemented on StartBED [1].

Using POSIX thread library each module is controlled by a thread.[2]. This procedure helps us to implement the application on a different number of machines by only changing configuration files. The simulation is event-driven. Using highly portable C code to run on the different testing OS and architectures (Mac, FreeBSD) with multiprogramming (across the modules discussed later) & multi-threading (within modules).

5.6 Modules

We defined 4 modules for the simulation.

Sensor Node

The sensor-node model is used to emulate a sensor node on embedded machines. Its basic task is to execute two functions: detection and reporting while going from on

[1]StartBED is a large scale testbed developped by Hokurikku Research center & JAIST

[2]It is mandatory to understand that the language and the procedure can change based on the type of devices that are using.

state to another; sleep, listen, wake.

Cluster Head

The cluster head model contains a set of functions that can received and save the data in history and predict the next location of the mobile object.

Sink node

Sink node is the module used to handle the system architecture and the recovery process.

Mobile Object

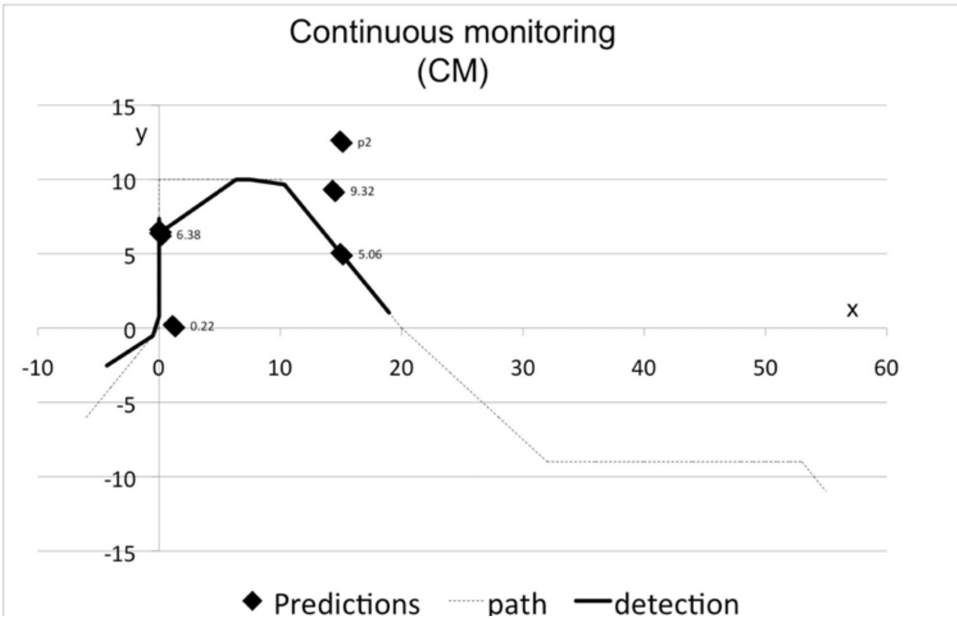
The object model represents the intruding object. It is meant to execute a path and generate some stimuli to the nodes.

5.7 Experiment Results

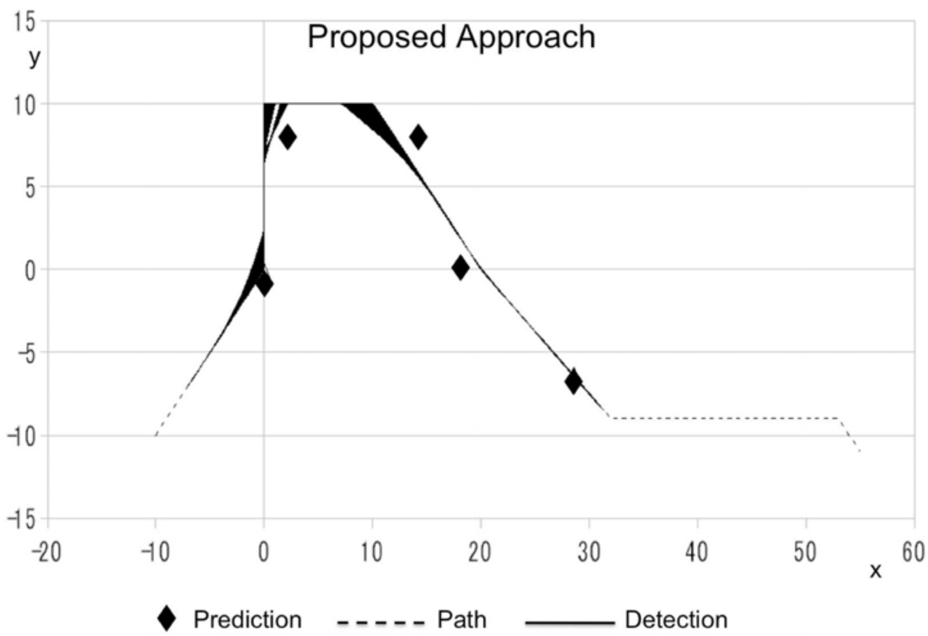
These cases show the motion of the object in the sensing area. The object starts moving from a random position, detected in sensors range and move continuously along the path.

Figure 5.4 shows the comparison between the proposed approach and the continuous monitoring scheme. In the proposed approach, the monitoring is done by cluster. Each sensor reports to its cluster and cluster heads compute the predictive value based on the order messages have been received. In the case continuous monitoring scheme, one sensor do both monitoring and reporting. It will predict the next node to handle the tracking before going to sleep.

On the Figures two information are revealed along the path: the detection and the prediction. The prediction is represented by diamonds and indicates at which point to initiate the recovery mechanism. In the case of the proposed approach figures **??** and 5.4(b) show an intense detection each time a variation of the direction occurs. This is due to the fact that the average position is computed by using the positions received from the sensor nodes. In each case the prediction is obtained based on that final coordinate.



(a) Continuous monitoring(CM)



(b) [Proposed approach

Figure 5.4: Number of sensor nodes from 0 to 100 and 1 sink

Chapter 6

Discussions

This section first describes results obtained from the simulation, compares with previous approaches' results and shows how the proposed solution o ers better energy e cient solution.

6.1 Performance Metrics

When solving the object tracking problem some important performance metrics need to be considered. As announced in the introduction we evaluated the performance of the proposed system:

- by varying the size of the network i.e. apply the system to different scales
- by considering the reporting frequency of sensor nodes to update packets.

In order to evaluate first, the prediction accuracy and second the total energy consumes by the system for communication and computation. We used metrics such as missing rate computed by using the Root Mean Square Error the R_{mre}, communication bandwidth, total energy transmission.

6.2 Missing Rate

The missing rate is used to estimate the error that can occur while doing prediction. It represents an estimation for the distance of the actual target position from the predicted position.

6.2.1 Metric

The Root Mean Square Error is directly interpretable in terms of measurement units, and it is used to compare the deviation in predictions of a the mobile object's position. It is given by the equation bellow and denoted by R_{mre}.

$$R_{mre} = \frac{\sum \sqrt[n]{\Delta_x^2 + \Delta_y^2}}{N_p} \tag{6.1}$$

Δ_x	Distance between the predicted and actual position of the object at time t along the x component
Δ_y	Distance between the predicted and actual position of the object at time t along the y component
N_p	Number of predictions

Table 6.1: Root Mean Square Error parameters

6.2.2 Result Analysis

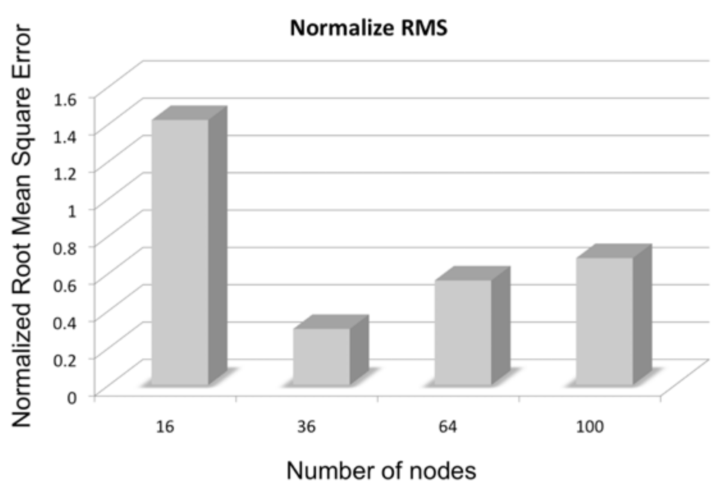


Figure 6.1: Prediction error

6.2.3 Interpretation

The graph shows the prediction rate computed for each run of the experiment. The first case shows how the number of predictions increases. This is based on the fact that the number of nodes available to handle the next movement of the tracking is less than the other cases. The graph shows a slight difference between the other three runs. This is due to the fact that less predictions are done.

6.3 Communication

In Wireless Sensor Networks, nodes are capable of distributed sensing and control to operate in dynamic environments where the occurrence of events can be very rare. In the other hand tracking applications when applied to these networks require far more reporting from the sensor nodes. This fact reveals other challenges such as how to economize and adaptively control all resources such as energy, communication bandwidth, and sensor sampling frequency.

6.3.1 Metrics

Generally speaking, communication bandwidth in computer networks is often used as a synonym for data transfer rate i.e. the amount of data that can be carried from one point to another. The following line described the three parameters used to evaluate the usage of the communication bandwidth. They are: Reporting messages, Wake up messages, Recovery requests.

6.3.2 Result Analysis

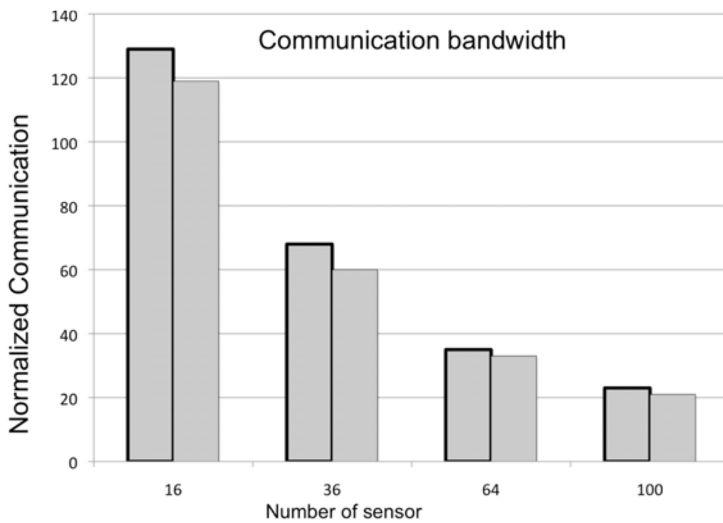


Figure 6.2: Communication

6.3.3 Interpretation

The graph compares between the proposed approach and the simple detect report approach. The first chart represents the proposed system and the second one describes the result of the naive scheme. The naive scheme considers only reporting messages while the proposed systems must handle in addition, wake up messages and recovery requests. From the obtained results shown in the graph we can notice that the difference between the schemes in term of communication is negligible within 3% i.e the proposed approach can balance the data flow as the size of the network increases.

6.4 Total Energy Consumption

6.4.1 Metrics

The total energy consumption in the system is denoted by T_{EC}. It is the sum of both the total energy consuming for monitoring denoted by T_{EM} and the total energy consuming for communication denoted by T_{ECM}.

$$T_{EC} = T_{EM} + T_{ECM} \tag{6.2}$$

6.4.2 Result Analysis

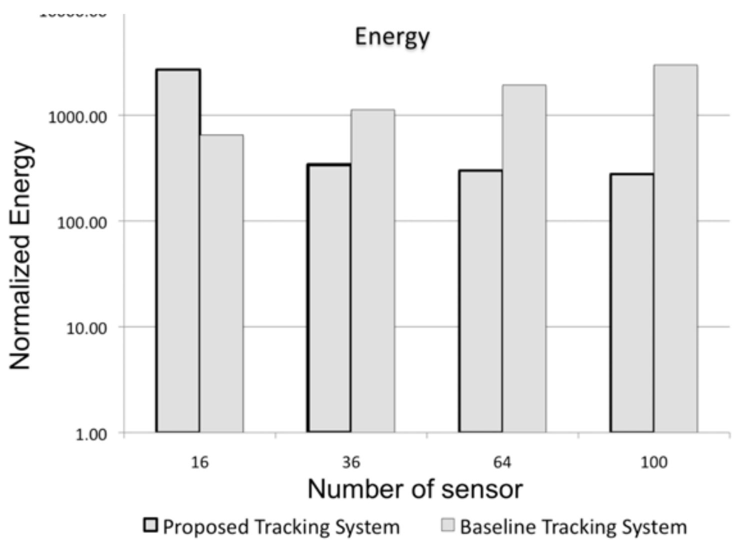


Figure 6.3: Energy

6.4.3 Explanation

To compute the energy consumption we adopted the values provided by the WINS project from Rockwell science center as proprosed by **Xu et al. 2004** [12].

Componen	Mode	Power
MCU	Activate	360
MCU	Sleep	0.9
Sensor	Active	23
Radio	Transmission	720
Radio	Receiving	369

Table 6.2: Energy Consumption on WINS nodes

6.5 Proposed approach vs DPT, PREMON and PES

By developing the object tracking application our priority was to solve the fall-backs encountered by the previous methods.

The first graph illustrated in figure 6.4 shows the effect of varying the reporting interval of sensor nodes on energy. Since object tracking application must keep the reporting frequency as low as possible in order to save energy.

We can notice that the energy efficiency of the approach falls under 50% which is better comparing to the prior approaches 55%. This is practically the same until 5000 msec, due to the fact that during this period the number of proposed sensors participating in the tracking scenario is almost the same. The rest of the graph reaches a number less then 10% for the proposed approach since more nodes are available to track the object but not active. On the other hand more useless updates is performed by the prior work.

Figure 6.5 shows the graph that illustrates the energy of the system when the number of nodes increase. The prior work divided the monitoring area into grids. We did the comparisons based on the fact that when the number of grids increases, the resulting grid location model approaches our geometrical location model. The results show that the more accurate the system gets the more energy is consumed.

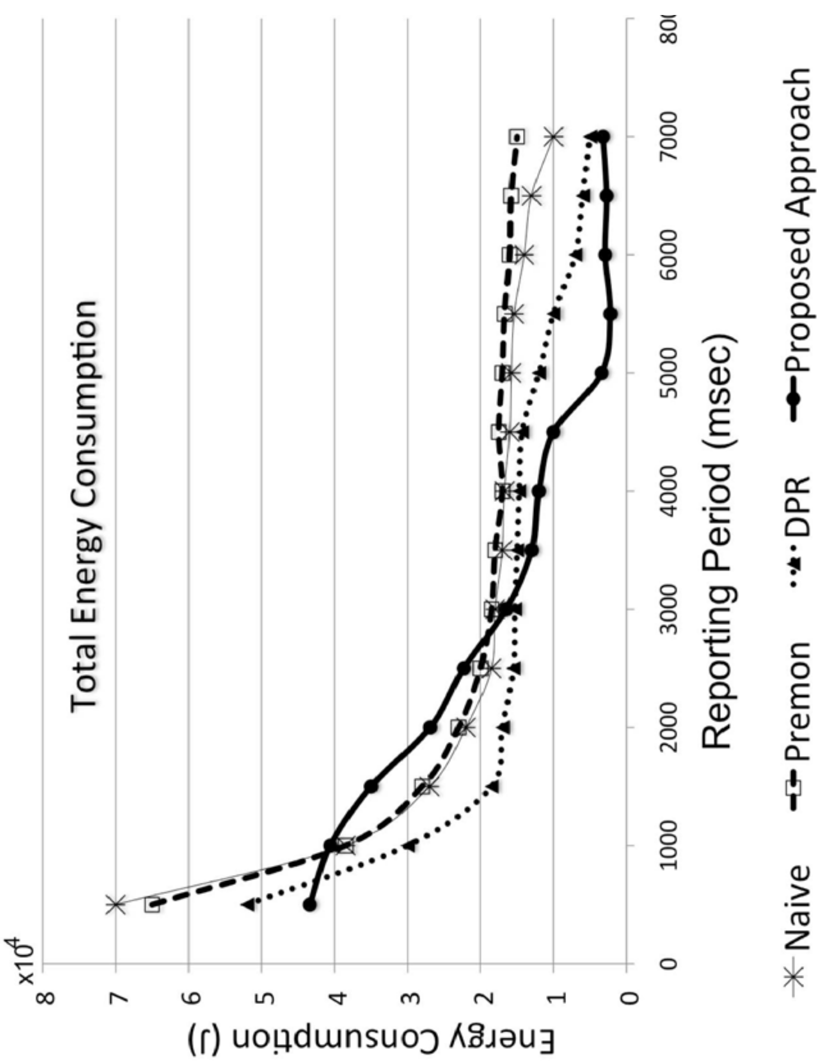


Figure 6.4: Total Energy Consumption

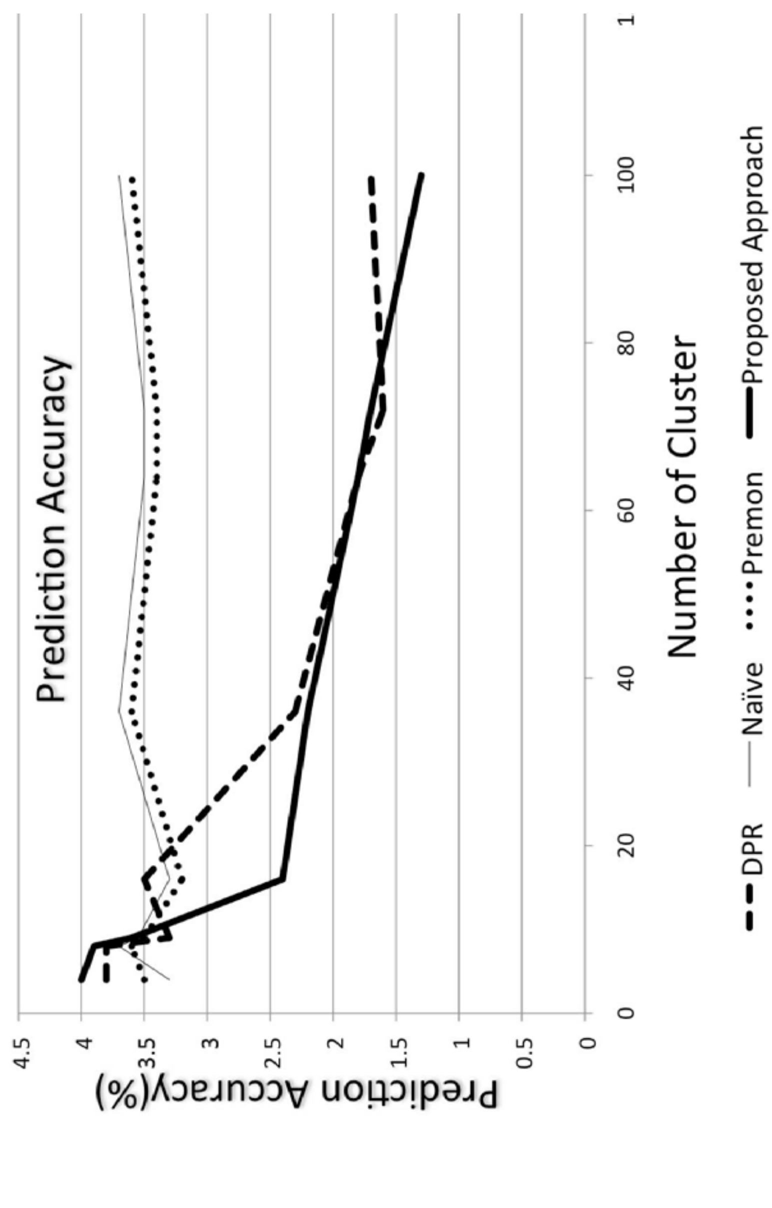


Figure 6.5: Prediction Accuracy

Chapter 7

Conclusion and Future Work

This section emphasizes the system improvement and mentions work that can be conducted to further enhance the proposed approach.

7.1 Conclusion

The thesis explored the usage of mobile object tracking applications in Wireless Sensor Networks. In this kind of applications, scalability or reporting period or data precision or network workload, or any other issue is somehow or another energy consumption related. This is due on the fact that the power carried by each sensor node is very limited. This thesis main concern was two fold, first to simulate a tracking application to handle architecture modifications and second to address the energy saving problem under such conditions.

To handle the architecture modifications problem we propose a multi-tiers architecture. Over this architecture we simulate an object tracking application. This latter is based on a prediction mechanism in which cluster heads decide whether or not sensor nodes should be alerted to wake up and report the movement states of the object. Otherwise nodes keep sleeping thus save energy. The system consists of a detection phase that determines and reports the movement states of an object. Based on a Sleep/Predict/Awake mechanism a prediction estimates the next movement. As this method is subject to some miss-prediction and errors, we proposed a recovery phase to reduce the effect of those miss predictions. At last a locating model to determine which node to wake up to handle the next tracking scenario.

Simulation has been done to evaluate the proposed system. Besides other problems encountered while running the experiments, the synchronization issue was considered. This problem showed up during the reporting operations of sensor nodes but the system responded well with simple synchronization techniques. As shown in chapter 6 the proposed system could:

1. Reduced the rate of increase in energy consumption by taking advantages of the hierarchical structure, figure 6.4.

2. Reduced the rate of monitoring by using sleep/predict /wake up mechanism.

3. Reduced the rate of communication bandwidth by suppressing useless wake up message in figure 6.2.

4. Reasonable recovery from miss predictions managed to reduce the effect of detection failures (miss prediction, real failures, target sudden variations) as shown in figure 6.1

5. Minimized the management overhead of the prediction and wake up mechanism. 6.4

6. The higher average sleep time the more energy saved, as we analyzed the metrics described in the experiment chapter. We showed that the proposed mechanism increases the sleep time per node thus it decreases the energy consumption, figure 6.5.

In conclusion, the proposed mechanism increases the sleep time per node thus it decreases the energy consumption.

7.2 Future Work

1. Network In order to strongly experience the scalability issue of the system, the network must handle a large number of nodes. Therefore a potential future research is to bring the system on a testbed. Due to node mobility, environmental obstructions, restricted ressources etc the sensor networks exhibit a highly dynamic network topology, additional future work can be to use a dynamic network to improve our system.

2. Reporting Mechanism Using other predictors would require more computational complexity consuming more time or more sophisticated node architecture which again uses more energy. Yet a better synchronization solution can improve the accuracy of the prediction and reduce the need to recover. The linear prediction accuracy is maximized when the steps are very close, i.e., more samples are involved (transmissions). This process can consume more energy, thus a compromise should be made between accuracy and energy conservation.

3. Deployment Finally, in the future we are expecting or at least we can challenge a real-world world application based on our system.

Bibliography

[1] Feng Zhao and Guibas Leonidas J. **Wireless Sensor Networks An information procedessing approach**, Edited by Karyn Johnson Rick Adams. San Francisco, California: Morgan Kaufman: Elsevier Science 2004, 2004.

[2] Perkins, Charles E. **Ad hoc networking** Edited by Charles E. Perkins Editor. Boston, MA: Addison-Wesley, 2001.

[3] Stevens, W. Richard. **UNIX network programming 2nd ed** Vol. 2. Upper Saddle River, NJ 07458: Prentice Hall PTR, 1999.

[4] Tatiana, Bokareva, et al. **Wireless Sensor Networks for Battlefield Surveillance** Land Warfare Conference (LWC 2006). Brisbane, 2006. 8.

[5] Alan, Mainwaring, Polastre Joseph, Szewczyk Robert, Culler David, and Anderson John. **Wireless Sensor Networks for Habitat Monitoring** Edited by ACM. WSNA. Georgia, 2002. 10.

[6] Konrad, Lorincz, et al. **Sensor Networks for Emergency Response: Challenges and Opportunities.** PERVASIVE Computing (IEEE Pervasive Computing) 3, no. 4 (October-December 2004): 16-23.

[7] Tabar, Ali Maleki, Keshavarz Arezou, and Aghajan Hamid. **Smart Home Care Network using Sensor Fusion and Distributed Vision-based Reasoning.** VSSN'06. California: ACM, 2006. 10.

[8] S. Banerjee and S. Khuller **A clustering scheme for hierarchical control in multi-hop wireless networks** Proceedings of IEEE INFOCOM, pp. 1028-1037, Anchorage AK, April 2001.

[9] Kung, H.T. and Vlah, D. **E cient location tracking using sensor networks** Wireless Communications and Networking Conference (WCNC), 2003.

[10] Winter, J.; Xu, Y. and Lee, W. C. **Prediction based strategies for energy saving in object-tracking sensor networks** IEEE International Conference on Mobile Data Management 2004, 364-357.

[11] Yingqi Xu and Wang-Chien Lee **On Localized Prediction for Power E cient Object Tracking in Sensor Networks** Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops (ICDCSW03).

[12] Yingqi Xu; Winter, J. and Wang-Chien Lee **Dual prediction-based reporting for object tracking sensor networks** The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous4), Aug. 22-26, 2004, pp. 154 163.

[13] T. Tung and A. Jamalipou **A novel sectional paging strategy for location tracking in cellular networks** Communications Letters, IEEE, 8(1), Jan 2004.

[14] S Goel and T. Imielinski, **Prediction-based monitoring in sensor networks: taking lessons from MPEG** ACM computer Communication Review, 31(5), October 2001.

[15] Song Ci, Mohsen Guizani and Hamid Sharif, **Adaptive clustering in Wireless Sensor Networks by mining sensor energy data**, Computer Communications, Volume 30, Issues 14-15, 15 October 2007, Pages 2968-297.

[16] Yang, H. and Sikdar, B. **A protocol for tracking mobile targets using sensor networks** Proceedings of the First IEEE. 2003 IEEE International Workshop 71- 81, 11 May 2003.

[17] **Network Embedded Systems Technology (NEST). http://dtsn.darpa.mil/ixo/programdetail.asp?progid=42.**

[18] **Engineering, UC Berkeley College of. http://www.coe.berkeley.edu .**

[19] **Inc., Dust. http://www.dust-inc.com .**

[20] **Research, Intel. http://www.intel.com/research/index.htm**

[21] **Systems, UCLA Center for Embedded Network. http://cens.ucla.edu/**

[22] **Wikipedia. http://en.wikipedia.org/wiki/Sensor_Networks.**

[23] **Crossbow Tehnology. http://www.xbow.com/index.aspx**

[24] **Rockwell Science Center. http://www.wins.rsc.rosckwell.com**