Energy Aware Target Tracking in Underwater Wireless Sensor Networks (UWSNs)

Nazia Majadi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

October 2012

Energy Aware Target Tracking in Underwater Wireless Sensor Networks (UWSNs)

A Thesis Submitted to the

Department of Computer Science and Engineering

Of

Bangladesh University of Engineering and Technology

in Partial Fulfillment of the Requirement

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

By

Nazia Majadi

Student ID: 100705005P

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

October 2012

The thesis entitled "**Energy Aware Target Tracking in Underwater Wireless Sensor Networks (UWSNs)"** submitted by Nazia Majadi, Student No: 100705005P, Session: October, 2007 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING on October 13, 2012.

BOARD OF EXAMINERS

1. ----------
   (Dr. Mahmuda Naznin) **Chairman**
   Associate Professor (Supervisor)
   Department of Computer Science and Engineering
   Bangladesh University of Engineering and Technology
   Dhaka-1000, Bangladesh.

2. ----------
   (Dr. A. S. M. Latiful Hoque) **Member**
   Professor and Head (Ex-officio)
   Department of Computer Science and Engineering
   Bangladesh University of Engineering and Technology
   Dhaka-1000, Bangladesh.

3. ----------
   (Dr. Mostofa Akbar) **Member**
   Professor
   Department of Computer Science and Engineering
   Bangladesh University of Engineering and Technology
   Dhaka-1000, Bangladesh.

4. ----------
   (Dr. Md. Rashedur Rahman) **Member**
   Assistant Professor (External)
   Department of Electrical Engineering and Computer Science
   North South University
   Dhaka-1229, Bangladesh.

Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the candidate

(Nazia Majadi)

Dedication

To my beloved parents.

ACKNOWLEDGEMENT

We thank the Almighty for the successful completion of our research. I express my heartily gratitude to the respected supervisor Dr. Mahmuda Naznin, Associate Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering Technology (BUET). We would like to acknowledge with great thanks to all friends who help us with giving various information and supports. I also thank my board members for their valuable comments and suggestions.

ABSTRACT

Tracking targets in Underwater Wireless Sensor Networks (UWSNs) is more challenging than the terrestrial Wireless Sensor Networks (WSNs) because in an UWSN, Radio Frequency (RF) communication is not possible and acoustic channel is used. The propagation delay in acoustic channel is much higher and variable than that of RF channel. To track a target in an UWSN accurately, it is required to keep many sensors active. But if the sensors are always kept active, the energy consumption is high which causes the network failure. Besides in an UWSN, due to the high and variable propagation delay in data transmission, information management of a moving target is quite challenging. In this thesis, we provide an energy efficient target tracking solution for UWSNs. We give an algorithm for target detection with keeping a low number of sensors active which increases the network lifetime. Simulation results show that our tracking method is more energy efficient compared to a popular tracking method in UWSNs. We also compare the estimation of the target's position with the true target trajectory, and we find that our detected trajectory matches with the true trajectory.

Table of Contents

ACKNOWLEDGEMENT.....i

ABSTRACT.....ii

1 Introduction

1.1 Overview.....1

1.2 Underwater Wireless Sensor Networks.....3

1.3 WSNs vs UWSNs.....3

1.4 Challenges of UWSNs.....4

1.5 Applications of UWSNs.....5

1.6 System Model of UWSNs.....6

1.6.1 Communication Architecture.....6

1.6.1.1 2D UWSNs.....6

1.6.1.2 3D UWSNs.....8

1.6.2 Sensor Characteristics of UWSNs.....10

1.6.3 Communication Model of UWSNs.....10

1.7 Outline of the Thesis.....12

2 Background Study and the Challenge

2.1 Overview.....13

2.2 Research Work.....13

2.3 Problem Domain.....16

2.3.1 Preliminaries.....16

A. *Three-sigma* (3σ) *region*.....16

B. *Working Sensor Nodes* (*WNs*).....17

C. *Processing Sensor Nodes* (*PNs*)....17
D. *Sleeping Sensor Nodes* (*SNs*)....17
2.3.2 Assumptions....17
2.4 Target Tracking Architecture....18
2.4.1 Position Estimation of a Moving Target....20
2.4.2 Objectives of a Tracking Method in UWSN....20

3 Energy Aware Target Tracking

3.1 Overview....21
3.2 Target Location Estimation....21
3.2.1 System Model....22
3.2.2 Load Balancing Strategy....23
3.2.3 Node Selection....24
3.3 ETRACK in Details....29
3.3.1 ETRACK Algorithm....31
3.3.2 Summary....34

4 Simulation Results

4.1 Overview....35
4.2 Testbed Description....35
4.2.1 Sensor Node Architecture in MiXiM....35
4.2.2 Simulation Settings....37
4.2.3 Simulation Metrics....39
4.3 Simulation Results....40
A. Residual Energy....40
B. The Number of Dead Nodes....41
C. Tracking Trajectory....42

5 Conclusion

5.1 Overview....44

5.2 Summary....44

5.3 Future Work....44

References....45

Appendix A....51

Appendix B....55

List of Figures

1.1 A 2D architecture of an UWSN....7
1.2 A 3D architecture of an UWSN....9
1.3 Internal architecture of an underwater sensor node....10
2.1 An application scenario of target detection....14
2.2 An application scenario of waked-up and sleeping sensors....15
2.3 A *3σ region*, *WNs*, *PNs* and *SNs*....16
2.4 Initial step of a target tracking scenario....18
2.5 *PN* Selection of a target tracking scenario....18
2.6 Data processing and collection by a new *PN*....19
2.7 Final step of a target tracking scenario....19
3.1 Node selection example....24
3.2 Initial step of a *PN* selection....25
3.3 Step 1 of feasible solution phase....25
3.4 Step 2 of feasible solution phase....26
3.5 Step 3 of feasible solution phase....26
3.6 Step 4 of feasible solution phase....27
3.7 Step 1 of local search phase....27
3.8 Step 2 of local search phase....28
3.9 Step 3 of local search phase....28
3.10 Step 4 of local search phase....29
3.11 Summary of ETRACK Algorithm....34
4.1 Structure of a simulated sensor node....36
4.2 The residual energy of the network (total no. of nodes is 150)....40
4.3 The residual energy of the network (total no. of nodes is 200)....41
4.4 The number of dead nodes in the network (toatl no. of nodes is 150)....41

4.5 The number of dead nodes in the network (total no. of nodes is 200).....42
4.6 Target tracking accuracy (total no. of nodes is 150).....43
4.7 Target tracking accuracy (total no. of nodes is 200).....43

List of Tables

4.1 Simulation Settings....38
B.1 Experiment results of ETRACK and WuS/VMS (Total no. of nodes is 200)....55
B.2 Experiment results of ETRACK and WuS/VMS (Total no. of nodes is 150)....55
B.3 Experiment results of ETRACK for Resudial Energy(Average)....56
B.4 Comparison between Actual track and Predicted track (Average)....56
B.5 Experiment results of ETRACK and WuS/VMS for the number of dead nodes (Average)[total no. of nodes is 150]....57
B.6 Experiment results of ETRACK and WuS/VMS for the number of dead nodes (Average)[total no. of nodes is 200]....58

Chapter 1

Introduction

1.1 Overview

Wireless Sensor Network (WSN) has been one of demanding research areas now a days. It has become more and more popular as the large scale smart sensor network has diversivied applications. The original motivation behind the research of WSN was military applications. Among various military applications, we mention some common applications like - deployment of a large-scale acoustic sensor network for establishing the ocean surveillance system for the detection of moving targets, deployment of a self-organized terristrial WSN for battle field surveillance system or attaching microsensors to weapons for stockpile surveillance. As the costs for sensor nodes and communication networks is very low, sensor networks are being used for civilian applications. However, regardless of its diversified applications, the primary tasks of sensor networks are sensing or detection, data processing, and transmitting data to some remotely deployed recieveing system or base stations. The main steps of a successful sensor network include deployment of sensors, sensing or detecting, classification of detected data, localization or position estimation, data acquisition, transmission of the sensed data over multi-hop network. Recently, there has been a growing interest in monitoring aqueous environments (including oceans, rivers, lakes, ponds and reservoirs, etc.) for scientific exploration, commercial exploitation and attack protection. The ideal solution for this type of extensive monitoring is a networked underwater wireless sensor system, which is called an Underwater Wireless Sensor Network (UWSN). It provides a promising solution for efficiently exploring and observing the aqueous environments [16, 18, 35].

A wireless sensor network is a system of small, wirelessly communicating nodes in which each node is equipped with multiple components [16]. In particular,

each node has a computation engine; communication and storage subsystems; a battery supply; and sensing and, in some cases, actuating devices. Such a network in envisioned to integrate the physical world with the internet and computations. The power supply on each node is relatively limited, and frequent replacement of the batteries is often not practical because of the large number of nodes are deployed in the hostile environment. Therefore, energy is the most constraining factor for achieving the proper functionality. In order to save energy, nodes use multi-hop, short-ranged communication [17]. The sensor nodes, in addition to sensing data; also forward data originated from othe ther nodes towards the destination. Therefore, the data transmission between two nodes, or between a node and the base station are done in multiple hops, instead of using the more expensive single hop long range transmissions.

In tracking applications, sensors actively probe for occurrence of phenomenon that is, in the most of cases, sensors track the moving entities into their coverage regions. Once such activity is detected, the sensors locally gather information to determine the target's spatial location. The information is conveyed to some remote stations. Sometimes, the collected information over a particular time is fused in a sensor node to obtain the global information for the location estimation of the target.

The issue of tracking targets in UWSNs has gained worldwide attention in recent years. The energy consumptions in the resource-constrained network devices have made tracking a target in an UWSN quite challenging research field. Target tracking, being an interesting research problem, has been expressed and solved by many form different perspectives from time to time. However, the performance of a tracking method depends on the application environment also. Since different applications consist of different sets of target entities, sensing models and environmental models, target tracking models are designed for specific goals and scenarios. Among different desirable quantities and performance goals of a sensor network, energy awareness is one of the key research challenges for sensor networks. Almost all of the sensing and routing devices of sensor networks are equipped with the limited power. Therefore, while tracking of targets, an

efficient sensor energy management policy is required [19]. The goal of sensor energy management is to choose the actions for an individual sensor dynamically to maximize the overall network performance.

1.2 Underwater Wireless Sensor Networks

In an Under Water Sensor Network (UWSN), sensor nodes are deemed to enable applications for oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation and tactical surveillance applications. Multiple unmanned or autonomous underwater vehicles (UUVs, AUVs), equipped with the underwater sensors, also find applications in exploration of natural undersea resources and gathering of scientific data in collaborative monitoring missions [16, 18]. To make these applications viable, there is a need to enable underwater communications among underwater devices. Underwater sensor nodes and vehicles must possess self-configuration capabilities, i.e., they must be able to coordinate their operation by exchanging configuration, location and movement information, and to relay monitored data to an onshore station. Wireless Underwater Acoustic Networking is the enabling technology for these applications. UWSN consisting of a variable number of sensors deployed to perform collaborative monitoring tasks over a given area, self-organizes to an autonomous network which can adapt to the characteristics of the ocean environment and can perform the tracking task [18].

1.3 WSNs vs UWSNs

Although there exists many recently developed network protocols for terristrial wireless sensor networks, the unique characteristics of the underwater acoustic communication channels, such as limited bandwidth capacity and specially the variable delays, the impact of underwater signal propagations, an UWSN requires different protocols [16].

The differences between terrestrial and underwater sensor networks can be itemized as follows [16]:

1. **Cost**. The underwater sensors are more expensive devices than terrestrial sensors.

2. **Deployment**. The deployment is deemed to be more sparse in underwater networks than the terrestrial WSNs.

3. **Spatial Correlation**. While the readings from the terrestrial sensors are often correlated, this is more unlikely to happen in underwater networks due to the higher distance among the sensors.

4. **Power**. Higher power is needed in underwater communications due to the higher distances and due to the more complex signal processing.

1.4 Challenges of UWSNs

As there are many differences between Wireless Sensor Networks (WSNs) and Underwater Wireless Sensor Networks (UWSNs), in designing Underwater Acoustic Networks the following challenges are needed to be considered [16]:

1. RF communication is not possible and acoustic channel is affected by the depth of the water;

2. Propagation delay in underwater is five orders of magnitude higher than in radio frequency (RF) channels, and extremely variable;

3. The underwater channel is severely impaired, especially due to multi-paths and fading;

4. The available bandwidth is severely limited;

5. Higher bit error rates and temporary losses of connectivity can be experienced due to the extreme characteristics of the underwater channels;

6. Battery power is limited and usually, batteries cannot be recharged, and also solar energy cannot be exploited;

7. Underwater sensors are prone to failures because of fouling and corrosion.

1.5 Applications of UWSNs

The broad range of applications for underwater acoustic sensor networks is as follows [16]:

1. **Ocean sampling networks.** Networks of sensors and AUVs, such as the Odyssey-class AUVs, can perform synoptic, cooperative adaptive sampling of the 3D coastal ocean environment. Experiments such as the Monterey Bay field experiment demonstrated the advantages of bringing together sophisticated new robotic vehicles with advanced ocean models to improve the ability to observe and predict the characteristics of the oceanic environment.

2. **Environmental monitoring.** UW-ASNs can perform pollution monitoring (chemical, biological and nuclear). For example, it may be possible to detail the chemical slurry of antibiotics, estrogen-type hormones and insecticides to monitor streams, rivers, lakes and ocean bays (water quality in situ analysis). Monitoring of ocean currents and winds, improved weather forecast, detecting climate change, understanding and predicting the effect of human activities on marine ecosystems, biological monitoring such as tracking of fishes or micro-organisms, are other possible applications. For example, the design and construction of a simple underwater sensor network is described to detect extreme temperature gradients (thermoclines), which are considered to be a breeding ground for certain marine micro-organisms.

3. **Undersea explorations.** Underwater sensor networks can help detecting underwater oilfields or reservoirs, determine routes for laying undersea cables, and assist in exploration for valuable minerals.

4. **Disaster prevention.** Sensor networks that measure seismic activity from remote locations can provide tsunami warnings to coastal areas [42], or study the effects of submarine earthquakes (seaquakes).

5. **Assisted navigation.** Sensors can be used to identify hazards on the

seabed, locate dangerous rocks or shoals in shallow waters, mooring positions, submerged wrecks, and to perform bathymetry profiling.

6. **Distributed tactical surveillance.** AUVs and fixed underwater sensors can collaboratively monitor areas for surveillance, reconnaissance, targeting and intrusion detection systems.

7. **Mine reconnaissance.** The simultaneous operation of multiple AUVs with acoustic and optical sensors can be used to perform rapid environmental assessment and detect mine-like objects.

8. **Pollution Monitoring.** And other environmental monitoring (chemical, biological, etc) [16].

1.6 System Model of UWSNs

An efficient and fault-tolerant network architecture plays a very important role in the success of a network. Apart from the channel delay and complexity of information transmission, interconnection, network topology has a significant impact on the network throughput. Now, we discuss the architecture of UWSNs.

1.6.1 Communication Architecture

The underwater sensor network topology is an open research issue recently which needs further analytic and simulative investigation from the research community. There are typically two types of Underwtaer Wireless Sensor Networks (UWSNs) [16]:

1. Two-dimensional UWSN

2. Three-dimensional UWSN

1.6.1.1 2D UWSNs

A 2D UWSN is constituted by sensor nodes which are anchored to the bottom of the water. Typical applications of this type of architecture may be environmental monitoring, or monitoring of underwater plates in tectonics [35].

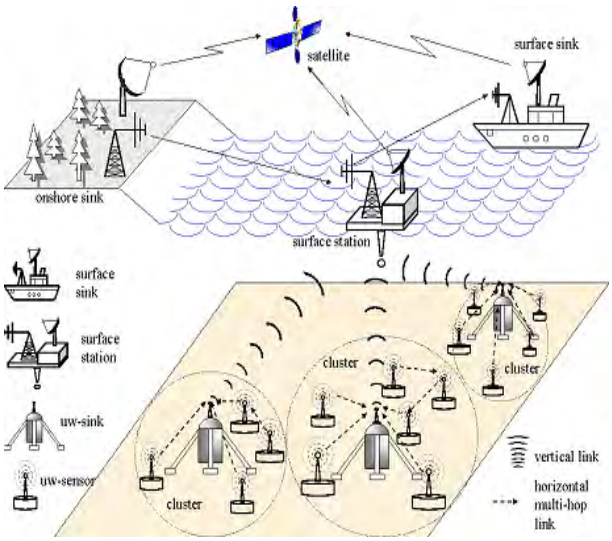

Fig. 1.1: A 2D architecture of an UWSN [16].

An architecture for two-dimensional underwater networks is shown in Fig. 1.1. A group of sensor nodes area anchored to the bottom of the ocean with the deep ocean anchors [16, 18]. By means of wireless acoustic links, underwater sensor nodes are interconnected to one or more *underwater sinks* (uw-sinks), which are also network devices in charge of relaying data from the ocean bottom network to the surface station. To achieve this objective, uw-sinks are equipped usually with two acoustic transceivers, namely –

1. A *vertical* transceiver and
2. A *horizontal* transceiver.

The horizontal transceiver is used by the uw-sink to communicate with the sensor nodes in order to:

1. send commands and configuration data to the sensors (uw-sink to sensors); and
2. collect monitored data (sensors to uw-sink).

The vertical link is used by the uwsinks to relay data to a *surface station*. Vertical transceivers must be long range transceivers for the deep water applications [16].

The surface station is equipped with an acoustic transceiver that is able to handle multiple parallel communications with the deployed uw-sinks. It is also endowed with a long range RF and/or satellite transmitter to communicate with the *onshore sink* (os-sink) or to a *surface sink* (s-sink). Sensors can be connected to uw-sinks via direct links or through multi-hop paths. In the former case, each sensor directly sends the gathered data to the selected uw-sink. This is the simplest way to network sensors, but it may not be the most energy efficient way, since the sink may be far from the node, and the power necessary to transmit may decay with the power greater than twice the distance [16, 35]. Furthermore, direct links are very likely to reduce the network throughput because of the increased acoustic interference due to the high transmission power. In case of multi-hop paths, as in terrestrial sensor networks, the data produced by a source sensor is relayed by the intermediate sensors until it reaches the uw-sink. This results in energy savings and the increased network capacity but also increases the routing complexity as well.

In fact, every network device usually takes part in a collaborative process whose objective is to diffuse topology information such that efficient and loop free routing decisions can be made at each intermediate node. This process involves signaling and computation. Since, as discussed above, energy and capacity are precious resources in underwater environments; in UWSNs the objective is to deliver event features by exploiting multi-hop paths and minimizing the signaling overhead necessary to construct underwater paths at the same time.

1.6.1.2 A 3D Architecture of an UWSN

In a 3D network, the depth can be controlled, and this kind of networks may be used for surveillance applications or monitoring of ocean phenomena (ocean biogeochemical processes, water streams, pollution, etc) [35].

In three-dimensional underwater networks, sensor nodes float at different depths in order to observe a given phenomenon. One possible solution would be to attach each uw-sensor node to a surface buoy, by means of wires whose length can be regulated to adjust the depth of each sensor node [16, 18, 35].

However, although this solution allows easy and quick deployment of the sensor network, multiple floating buoys may obstruct ships navigating on the surface, or they can be easily detected and deactivated by enemies in military settings. For these reasons, a different approach can be used to anchor sensor devices to the bottom of the ocean.

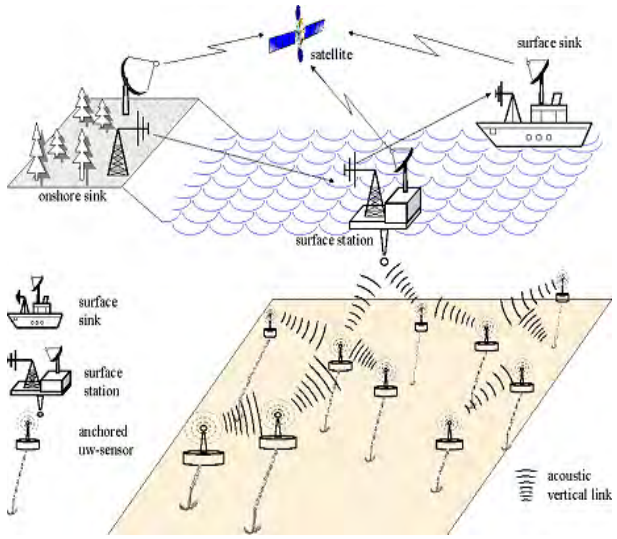

Fig. 1.2: A 3D architecture of an UWSN [16].

In this architecture, depicted in Fig. 1.2, each sensor is anchored to the ocean bottom and equipped with a floating buoy that can be inflated by a pump. The buoy pushes the sensor towards the ocean surface. The depth of the sensor can be regulated by adjusting the length of the wire that connects the sensor to the anchor, by means of an electronically controlled engine that resides on the sensor [16].

Many challenges arise with such an architecture, those needed to be solved in order to enable 3D monitoring. The challenges are described as follows:

1. **Sensing coverage**. Sensors should collaboratively regulate their depth in order to achieve full column coverage, according to their *sensing ranges*. Hence, it must be possible to obtain sampling of the desired phenomenon at all depths.

2. **Communication coverage**. Since in 3D underwater networks, the sinks are located above water surface, sensors should be able to relay information

to the surface station via multihop paths. Thus, the network devices should coordinate their depths in such a way that the network topology is always connected, i.e., at least one path from every sensor to the surface station always exists.

1.6.2 Sensor Characteristics of UWSNs

The typical internal architecture of an underwater sensor is shown in Fig. 1.3. It consists of a main controller/CPU which is interfaced with an oceanographic instrument or sensor through a sensor interface circuitry [7,18].

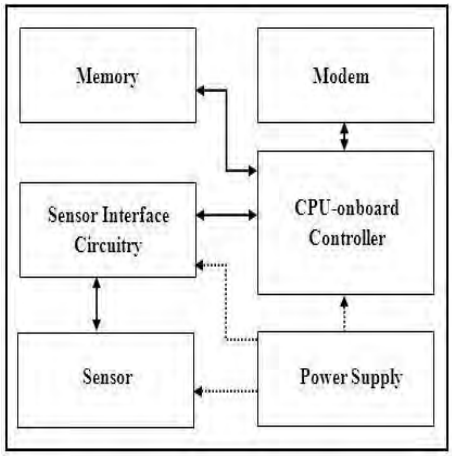

Fig. 1.3: Internal architecture of an underwater sensor node [18].

The controller receives data from the sensors and it can store and process in the onboard memory, and send it to the other network devices by controlling the acoustic modems. The electronics are usually mounted on a frame which is protected by a PVC housing. Sometimes all sensor components are protected by some bottom-mounted instrument frames those are designed to permit azimuthally omnidirectional acoustic communications, and protect sensors and modems. An internal architecture of an underwater sensor node is shown in Fig. 1.3.

1.6.3 Communication Model of UWSNs

Acoustic communications are the typical physical layer technology in underwater networks. In fact, radio waves propagate at long distances through conductive sea water only at extra low frequencies (from $30Hz$ to $300\ Hz$), which require

large antennae and high transmission power. Links in underwater networks are based on *acoustic* wireless communications [16].

The traditional communication model can not be directly applied to UWSNs, because of the following disadvantages [16]:

1. **No real-time monitoring.** The recorded data cannot be accessed until the instruments are recovered, which may happen several months after the beginning of the monitoring mission. This is critical especially in surveillance or in environmental monitoring applications such as seismic monitoring.

2. **No on-line system reconfiguration.** Interaction between onshore control systems and the monitoring instruments is not possible. This impedes any adaptive tuning of the instruments, nor is it possible to reconfigure the system after particular events occur.

3. **No failure detection.** If failures or misconfigurations occur, it may not be possible to detect them before the instruments are recovered. This can easily lead to the complete failure of a monitoring mission.

4. **Limited storage capacity.** The amount of data that can be recorded during the monitoring mission by every sensor is limited by the capacity of the onboard storage devices (memories, hard disks).

Therefore, there is a need to design underwater networks that will enable real-time monitoring of selected ocean areas, remote configuration and interaction with onshore human operators. This can be obtained by connecting underwater instruments by means of wireless links based on acoustic communication.

Underwater Wireless Sensor Networks (UWSNs) consist of sensors to perform collaborative monitoring tasks. A major challenge for the deployment of UWSNs is the development of a Medium Access Control (MAC) protocol tailored for the underwater environment. In particular, an underwater MAC protocol should provide high network throughput, low channel access delay, and low energy consump-

tion, in face of the harsh characteristics of the underwater propagation medium, while guaranteeing fairness among competing nodes [8].

A new variant communication protocol for UWSNs named UW-MAC is proposed [9]. It is a transmitter-based CDMA MAC protocol that incorporates a novel closed-loop distributed algorithm to jointly set the optimal transmit power and code length to minimize the near-far-effect. The near-far-effect occurs when the signal received by a receiver from a sender near the receiver is stronger than the signal received from another sender located further. In this case, the remote sender will be dominated by the close sender. The main features of UW-MAC are: i) it provides a unique and flexible solution for different architectures such as static two and three-dimensional in deep and shallow water; ii) it is fully distributed, since spreading codes and transmit power are distributively selected by each sender without relying on a centralized entity; iii) it is intrinsically secure, since it uses chaotic codes; iv) it fairly shares the bandwidth among active devices; and v) it efficiently supports multicast transmissions, since spreading codes are decided at the transmitter side [9].

1.7 Outline of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we discuss the related research work and the problem domain in details. Chapter 3 presents our tracking algorithm named as **ETRACK**. In Chapter 4, simulation results are provided. Finally, Chapter 5 concludes the thesis with a summary and a few proposal for future developments.

Chapter 2

Background Study and the Challenge

2.1 Overview

The challenges of underwater acoustic communication must be taken into account to detect and track an underwater target. For example, the propagation delay in underwater channel is five orders of magnitude higher than the terrestrial radio-frequency channel [16]. Besides, the delay is variable which causes communication failure also. Section 2.2 describes some related work on detection and tracking of a moving object in an UWSN. In Section 2.3, some related definitions and assumptions are given. Finally, Section 2.4 describes the tracking architecture of UWSNs.

2.2 Related Work

Targets detecting, classifying, and tracking in underwater environment are indispensable parts of modern underwater defense systems. Various types of sound navigation and ranging (sonar) arrays have been used for this purpose, such as surface-ship-hull-mounted sonar, submarine-hull-mounted sonar, side-scan sonar, and towed arrays [3, 6, 10, 20, 26, 30]. These sonar arrays are generally mounted on, or towed by a ship or a submersible [6], [26] or deployed prior to the application [10], which makes them unsuitable for one-demand tracking missions. Furthermore, the platform that tows the array or on which the array is mounted is a single point of failure for the entire system.

Sensor networks stand as a promising technology in terms of surveillance, reconnaissance, and targeting. Underwater sensor networks (UWSNs), are envisioned to enable applications for oceanographic data collection, offshore exploration, assisted navigation, and tactical surveillance applications [7].

There are some studies for underwater target detection and tracking with UWSNs [5, 13, 25, 28, 31, 40, 42]. However, time synchronization between the sensor nodes are assumed to be realized through global positioning system (GPS) mechanism in [36, 37, 38], which cannot be applied to UWSNs because it uses radar waves and those waves do not propagate in sea water.

A distributed UWSN provides unprecedented capabilities for target detection and localization [31]. In 2007, Zhou et al. [31] proposed localization solutions tailored to low-visibility targets based on a distributed UWSN. Their application scenario is depicted in Fig. 2.1. The authors describe their tracking method as like this, at first the source emits a waveform. They assume the propagation is to be omnidirectional, so that both the sensors and the target can receive this signal. The signal that arrives at the submarine surface gets reflected. The reflected wave is no longer omnidirectional. It propagates in a certain direction with a small beamwidth. All the sensors are equipped with a correlator to detect the transmitted signal by the source.

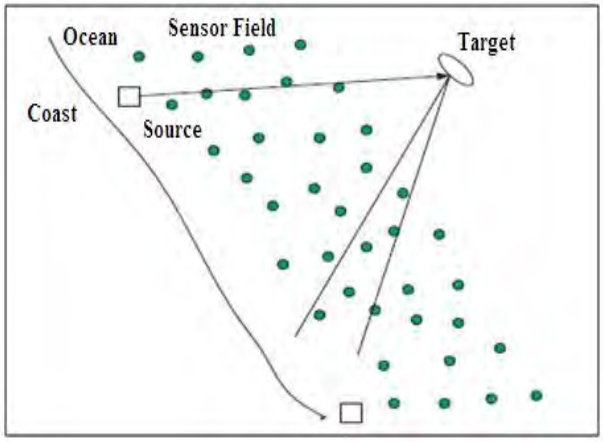

Fig. 2.1: An application scenario of target detection [31].

In this research [31], the authors calculate the detection result for each sensor in the sensor field and then they consider that, if the correlator output is higher than a certain threshold, the sensor which tracks the target declares a detection. Otherwise, it declares a no-detection. In this paper [31], the authors do this test within a certain time interval. The drwaback of this research is that the authors keep all the sensor nodes 'active' to estimate the location of the target which is

not energy efficient.

In 2008, Chang Ho Yu et al. [5] proposed an algorithm for tracking a algorithm as shown in Fig. 2.2. The authors assume that, when a target is moving (from Area **A** to Area **B** as in Fig. 2.2), one sensor is randomly selected as a data processing node. It computes the location of the target based on the information of the sensor nodes residing in the 3σ [42] range of that processing node.

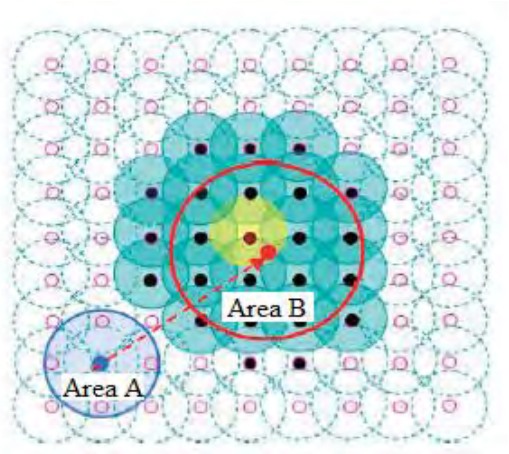

Fig. 2.2: An application scenario of selecting waked-up and sleeping sensors [5].

But all the sensors in that range may not be able to detect the target. Therefore, the authors calculate the distance between the target and the sensor nodes lying within the 3σ range of the processing node. They keep the sensors 'ON' which are only near the target. In this way, the authors claim energy efficiency. They call their method as Valid Measurement Scheme (VMS). But according to their tracking method, any node is selected randomly as processing node. It may happen the randomly selected processing node is less power than the other nodes near the target. The failing processing node may cause communication failure. Moreover, the authors did not consider the depth in estimating the location of the target.

In 2011, Isbitiren and Akan [13] proposed an underwater target tracking algorithm. For this purpose, the velocity and the projected location of the target are calculated by trilateration. Based on these calculations, the nodes along the path of the target are activated to continuously collect information about the target. This process continues until there is no signal that is received from the target.

The limitation of this paper is that the authors keep many sensor nodes as border nodes as active nodes to detect the entrance of the target in the network area. But keeping the border nodes always active may cause network cut.

2.3 Problem Domain

In this section, we define the probelm and some preliminaries relevant to our research problem. Here, we discuss some related definitions and assumptions. We also describe the problem in details.

2.3.1 Priliminaries

Now, we give some definitions which are used in our target tracking technique.

Definition A. ***Three-sigma*** **(*3σ*)** ***region***

In statistics, the three-sigma rule, or empirical rule states that for a normal distribution, nearly all values lie within 3σ (standard deviations) of the mean [5, 42].

About 68.27% of the values lie within 1σ of the mean. Similarly, about 95.45% of the values lie within 2σ of the mean. Nearly, all (99.73%) of the values lie within 3σ of the mean. In Fig. 2.3, the dotted circle presents the previous 3σ region and the solid circle presents the updated 3σ region.

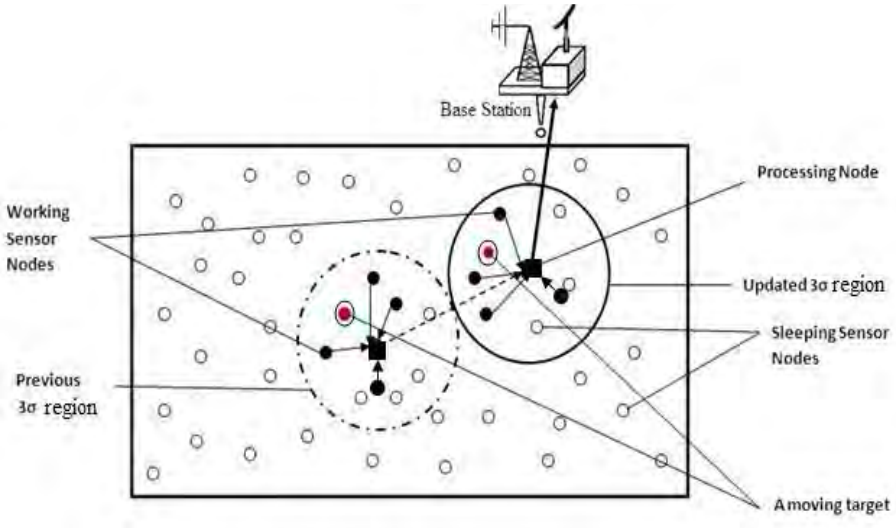

Fig. 2.3: A *3σ region*, *WNs*, *PNs* and *SNs*.

Definition B. *Working Sensor Nodes* (*WNs*)

In 3σ region, four sensor nodes are selected. These four nodes are called *WNs*. These nodes participate to track the moving target. In Fig. 2.3, black solid circles represent *WNs*.

Definition C. *Processing Sensor Nodes*(*PNs*)

A sensor node is selected based on the remaining energy and low power consumption is called a *Processing Node*(PN), instead of sending target tracking information by all working sensor nodes to the base station. A PN is responsible for the following tasks:

- Collecting the tracked data from all *WNs* within the 3σ region,
- Fusing the measured data
- Sending the fused data to the next PN or to the base station.

Rectangles represent *PNs* in Fig. 2.3.

Definition D. *Sleeping Sensor Nodes*(*SNs*)

Those nodes which are not participating in the target detection and tracking are in 'sleep mode'. These nodes are named as *Sleeping Sensor Nodes*(*SNs*). In Fig. 2.3, circles without no fill represent *SNs*.

2.3.2 Assumptions

We assume that all sensors are static and deployed at the bottom of underwater. The sensors are capable of communicating with each other wirelessly over acoustic channel.

Furthermore, the following assumptions are considered for our network:

1. A target can enter into the sensing field from any point on the boundary.
2. All sensor nodes are capable to relay target information to the base station or to other computational nodes.

2.4 Tracking Architecture

In this section, we describe the basic idea of the target tracking architecture and the position estimation technique.

The basic idea of the target tracking architecture is described in Fig. 2.4, where the dashed circles mean the sensor detection area.

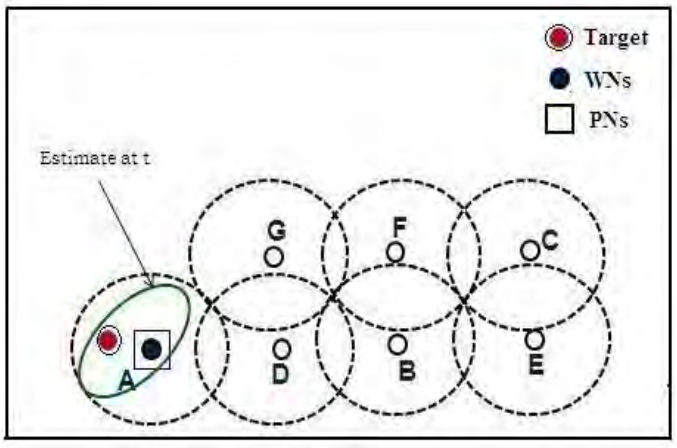

Fig. 2.4: Initial step of a target tracking scenario.

Initially, there exists an ellipse which shows the estimation area of the target (double lined circle) at any k^{th} time shown in Fig. 2.4. At the $(k+1)^{th}$ time step, a new ellipse represents the estimation area of the target computed by using Kalman Filter [14, 32].

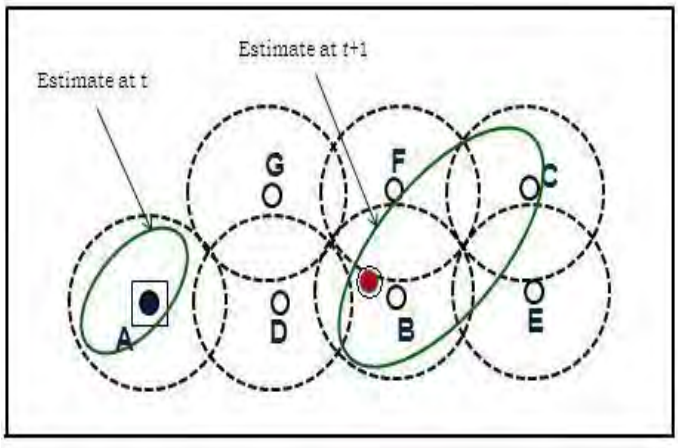

Fig. 2.5: *PN* selection of a target tracking scenario.

As Node '**B**' is the nearest sensor node to the center of the predicted ellipse, also Node '**B**' has more remaining energy and less power needed to track the

target, therefore, Node '**B**' is selected as the *PN* as shown in Fig. 2.5, and the information of the *PN* '**A**' is transferred to a new *PN* '**B**' as shown in Fig. 2.6.

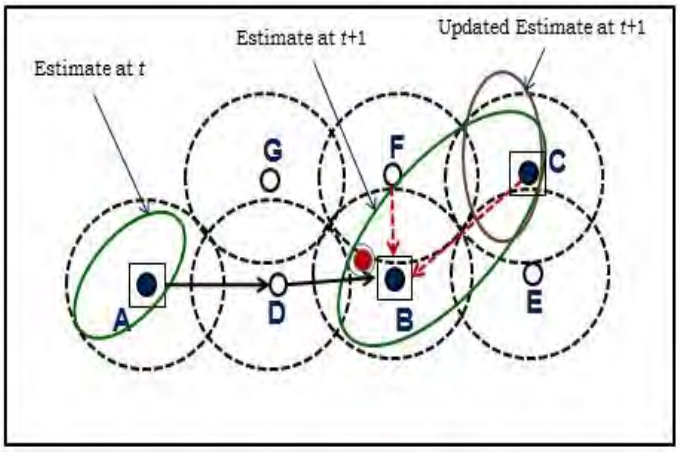

Fig. 2.6: Data processing and collection by new *PN*.

A new *PN* '**B**' collects measurements from the sensors which are inside the predicted ellipse. For example, '**B**' gets measurement data from Node '**F**'. We keep only four sensors active as this number is enough to locate the position in a three dimensional area [13]. These four sensor nodes are selected based on their remaining energy and power consumption.

Finally in Fig. 2.7, at the $(k+1)^{th}$ time step, a new updated area can be estimated by Kalman Filter. At every time step, the same procedure continues.

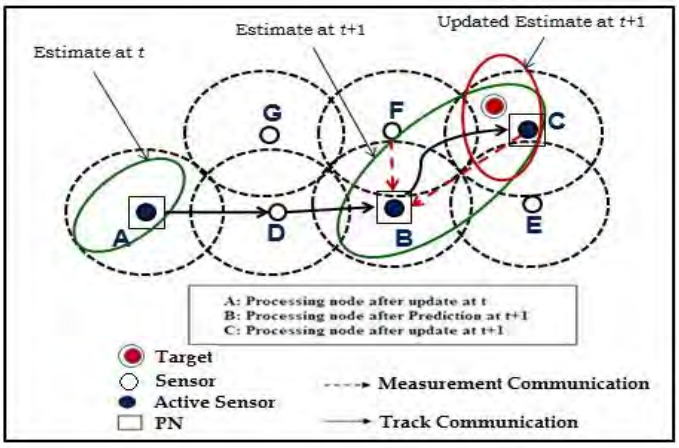

Fig. 2.7: Final step of a target tracking scenario.

2.4.1 Position Estimation of a Moving Target

To ensure the minimum number of data propagation, we follow prediction-based KF for estimating the target position moving in an UWSN [5, 10]. We prefer KF algorithm due to the following reasons:

1. Only the state mean and covariance are needed to be propagated from one local processing node to the next processing node.

2. This requires less data transmission which provides the prolongated network lifetime.

2.4.2 Objectives of a Tracking Method in UWSN

The objective of our research is looking for the policies which minimizes the energy consumption of the sensor nodes implying increased lifetime of the sensor network. Therefore, we want a tracking method with the goals stated as follows:

1. To design a load balanced tracking technique which minimizes the energy consumption.

2. To incorporate 3D architecture of the network for realistic target position estimation.

Chapter 3

Energy Aware Target Tracking

3.1 Overview

In this chapter, we describe our target tracking algorithm (**ETRACK**) which tracks a moving target in an UWSN with the load balancing. In Section 3.2, we describe our method in details. Load balancing strategy is described in Section 3.3. Section 3.4 gives the detail of **ETRACK** algorithm.

3.2 Target Location Estimation

This section describes the design models which we consider for detecting and tracking a moving target in UWSNs. The state vector of a single target consists of its position and velocity, in a three dimensional (3D) Cartesian coordinate system. At the k^{th} time step, the state vector $X(k)$ is defined as [5],

$$X(k) = \begin{bmatrix} x(k) \\ x'(k) \\ y(k) \\ y'(k) \\ z(k) \\ z'(k) \end{bmatrix} \ldots\ldots (3.1),$$

where a target is located at (x, y, z) coordinate with (x', y', z') velocity.

The state dynamics equation is given by [5],

$X(k+1) = \Phi X(k) + \tau w(k)$ ....(3.2),

where Φ is the state transition matrix, τ is the process noise matrix, and $w(k)$

is the independent process noise with zero-mean, white, Gaussian probability distribution $N(0,Q(k))$.

For a constant velocity model, parameters are defined as follows [5],

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.3)$$

$$\tau = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & T \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.4)$$

The process noise covariance matrix is [5],

$$Q(k) = \begin{bmatrix} qT & 0 & 0 \\ 0 & qT & 0 \\ 0 & 0 & qT \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.5),$$

where q is the intensity of the process noise, and T is the time interval between samples.

3.2.1 System Model

In this section, we define our network model. We assume that an UWSN is composed of N acoustic sensor nodes which are deployed in a $(x,\ y,\ z)$ coordinate. The positions of sensor nodes, in Cartesian coordinates denoted by $(x_i,\ y_i,\ z_i)$,

where i=1, 2, 3,. . . .,N, are arbitrary but known to the fusion center [5]. For example, Sensors s_1, s_2,. . . ,s_N provide the measurements $z_1(k)$, $z_2(k)$, . . . ,$z_N(k)$ at the k^{th} time step, the measurement model is given by [5],

$$Z(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \\ . \\ . \\ . \\ z_N(k) \end{bmatrix} = \begin{bmatrix} HX_1(k) \\ HX_2(k) \\ . \\ . \\ . \\ HX_N(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ . \\ . \\ .. \\ v_N(k) \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.6),$$

where H is a measurement function of sensor s_i, and $v_i(k)$ is its measurement noise which is assumed to be independent with zero-mean, white, Gaussian probability distributions $N(0,R_i(k))$ [5].

3.2.2 Load Balancing Strategy

In this section, we describe our load balancing strategy that we use in our target tracking algorithm. We call this algorithm **ETRACK**. In our tracking algorithm **ETRACK**, the load balancing strategy is an iterative process, where each iteration consists of two phases described as follows:

A. *Possible WNs Selection Phase*

In this phase, at a time, a feasible solution is built with one *WN*. At each iteration, the next *WN* to be added is determined by ordering all possible sensor nodes in a list with respect to a greedy function that measures the near-term benefit of selecting each *WN*. This list is called the *Possible Working Nodes Selection List* (*PWNSL*). The adaptive component of the heuristic arises from the fact that the benefits associated with each *WN* are updated at each iteration of this phase to reflect the changes brought by the selection of the previous *WN*s. The point is that, the selection of *WN*s in the *PWSNL* is based on the remaining energy of all sensor nodes those are in the 3σ region [5], and those which need

low power to track the moving target.

B. *Local Search Phase*

The solutions generated by the possible *WN*s phase are not guaranteed to be locally optimal. Hence, it is almost always beneficial to apply a local search to improve each constructed solution. A local search algorithm works in an iterative fashion by successively replacing the current solution by a better solution from its neighborhood. It terminates when there is no better solution found in the neighborhood.

3.2.3 Node Selection

We select the minimum number of nodes which are called *WNs* in our **ETRACK** algorithm.These take the responsibility for tracking a moving target. Rather than keeping all sensor nodes active for tracking target, selecting the minimum number of nodes for tracking increases the network lifetime.

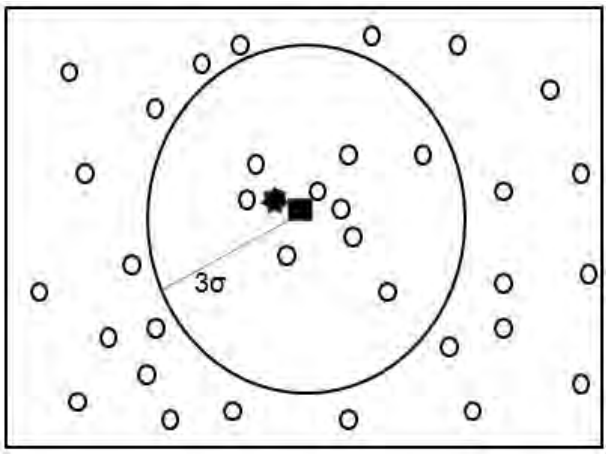

Fig. 3.1: Node selection example.

As shown in Fig. 3.1, let us assume that, an UWSN consists of some sensor nodes those are deployed in the target dectection region. The 'asterisk' symbol denotes a target. When a target enters into the detection region, a *PN* is selected based on the remaining energy and low power consumption needed to track a moving target. The rectangle denotes a selected *PN*. This *PN* draws a 3σ region centering itself.

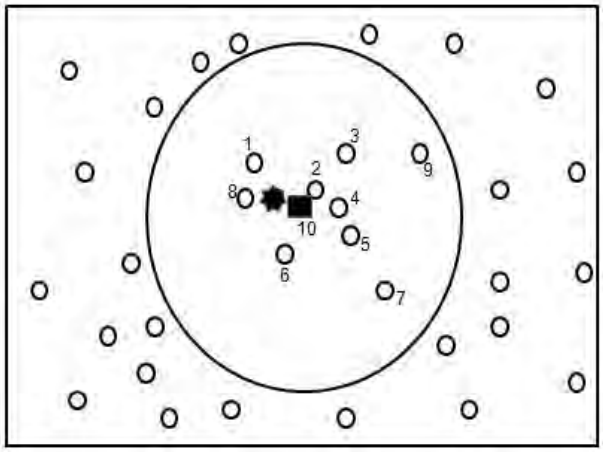

Fig. 3.2: Initial step of a *PN* selection.

Then within the 3σ region, let us assume that the selected *PN* is numbered as Node 10 and the remaining nine sensor nodes numbered as 1, 2, ..., 9 (see Fig. 3.2). As we know at least four sensor nodes are needed to track the target. Therefore, four best sensor nodes are selected by *Possible WNs Selection* phase and *local search* phase which are described below. These four selected sensor nodes are denoted as *WNs*.

A. *Possible WNs Selection Phase*

Initially we start with Sensor 1. The *Active Sensor List*(*ASL*) looks like: {1} as shown in Fig. 3.3.

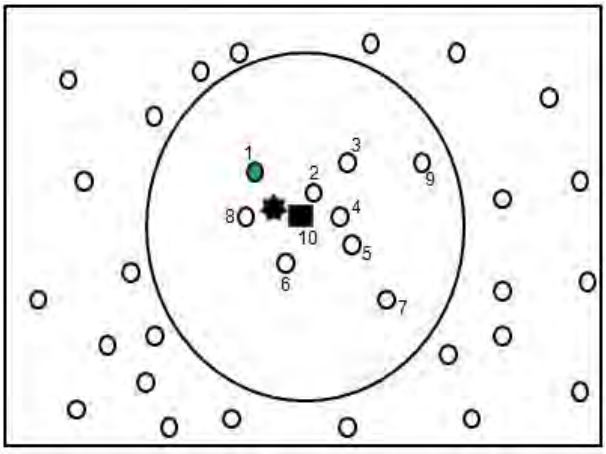

Fig. 3.3: Step 1 of *feasible solution* phase.

To generate Possible Working Nodes Selection List (PWNSL), we loop through the remaining 8 sensor nodes and find the four sensor nodes based on the distance

between the predicted position of the target and the position of the sensor node. After looping through the remaining sensor nodes, let us assume the *PWNSL* looks like: {3, 6, 7, 5}.

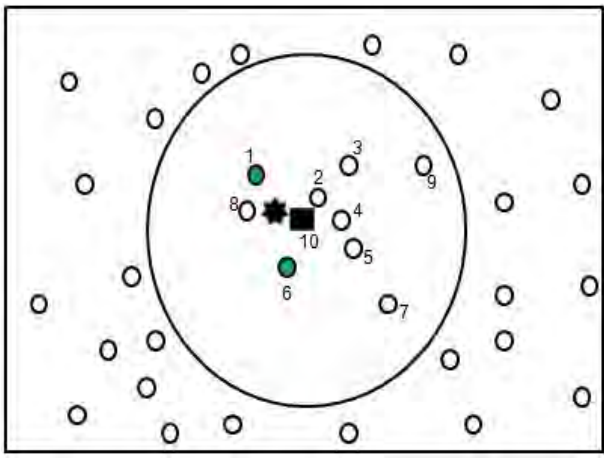

Fig. 3.4: Step 2 of *feasible solution* phase.

As shown in Fig. 3.4, We randomly select an entry from the *PWNSL*. Let us assume that this is Node 6. Now, the *ASL* looks like: {1, 6}. Repeating the previous algorithm, we loop through the remaining seven sensor nodes and find the four sensor nodes based on the distance between the predicted position of the target and the position of the sensor node. After looping through these remaining sensor nodes, let us assume our *PWNSL* looks like: {3, 8, 9, 7}. Again, one of these four sensor nodes is randomly selected and added to the *ASL*. Let us assume that, Node 3 is selected from our *PWNSL* (see Fig. 3.5).

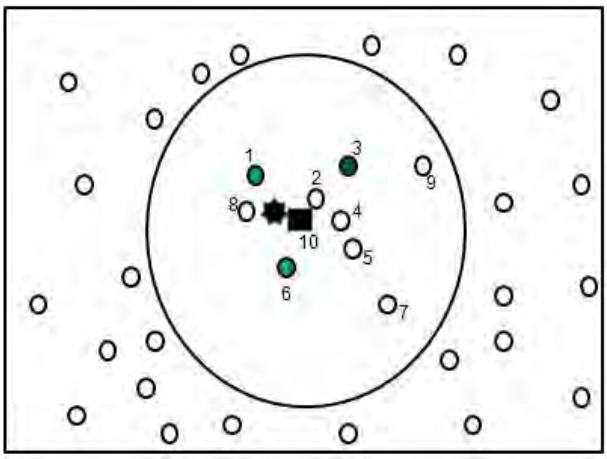

Fig. 3.5: Step 3 of *feasible solution* phase.

Then the *ASL* looks like:{1, 6, 3}. The previous algorithm is repeated among the remaining six sensor nodes for selecting the fourth sensor based on the distance between the predicted position of the target and the position of the sensor node. Now, let us assume that our *PWSL* looks like: {8, 4, 5, 7}. Now, the *ASL* looks like: {1, 6, 3, 5}.

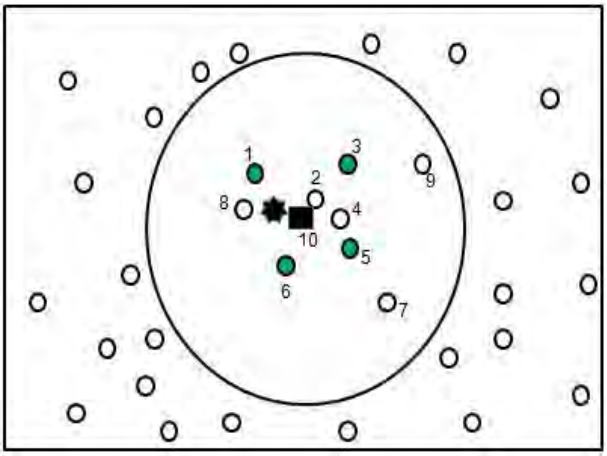

Fig. 3.6: Step 4 of *feasible solution* phase.

As we select four sensor nodes as *WNs* for detecting the target, therefore, the whole process is stopped here. Otherwise, the whole process will be repeated. Therefore, after *Possible WNs Selection* phase, the *ASL* looks like: {1, 6, 3, 5} (see Fig. 3.6).

B. *Local Search Phase*

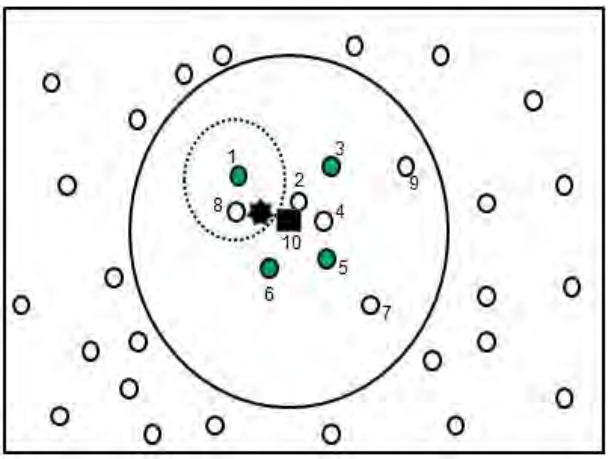

Fig. 3.7: Step 1 of *local search* phase.

In this phase, the neighborhoods of the selected sensor nodes which are in the *ASL* are checked based on the low-power communication needed to track the target. We find the *ASL* is {1, 6, 3, 5}.

The first entry (Node 1) from *ASL* is selected in this phase. The neighborhood of Node 1 (in the communication range of Node 1) is only Node 8 which needs less power to track the target than Node 1 as in Fig. 3.7.

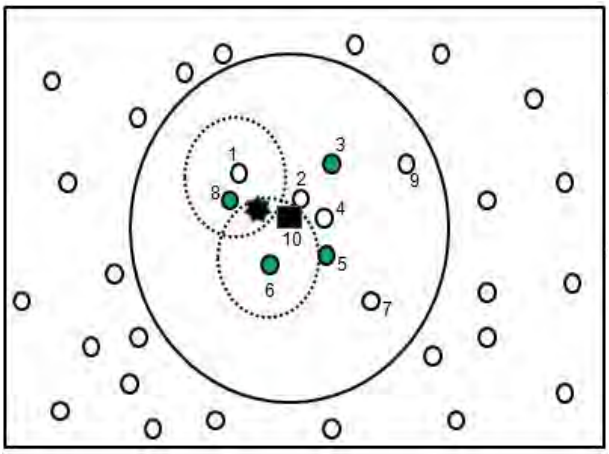

Fig. 3.8: Step 2 of *local search* phase.

Therefore, Node 1 is replaced by node 8. That is, the updated list looks like {8, 6, 3, 5}. Then, the next entry (Node 6) is checked. There is no neighbor of Node 6. Therefore, the solution will remain the same as Fig. 3.8.

Then, the third entry (Node 3) is checked. There are two neighbors of Node 3 like Node 2 and Node 4.

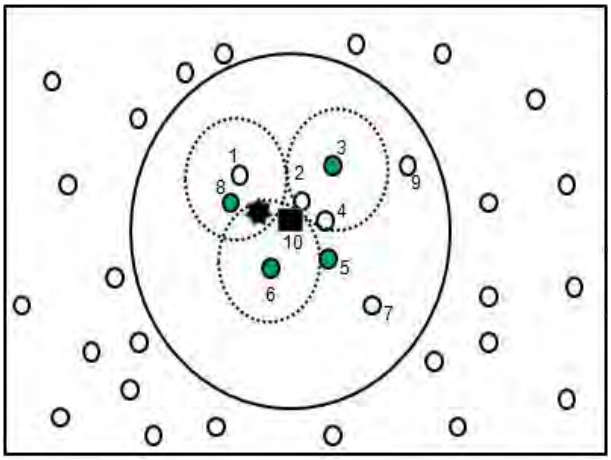

Fig. 3.9: Step 3 of *local search* phase.

Among these two nodes, Node 2 requires less power to track the target than Node 3 and Node 4. Thus, the similar way Node 3 is replaced by Node 2 as Fig. 3.9. Now, the updated list is, {8, 6, 2, 5}. Then, the fourth entry (Node 5) is checked. There is only one neighbor of Node 4, and Node 4 requires less power to track the target than Node 5. Therefore, Node 5 is replaced by Node 4 like the previous one.

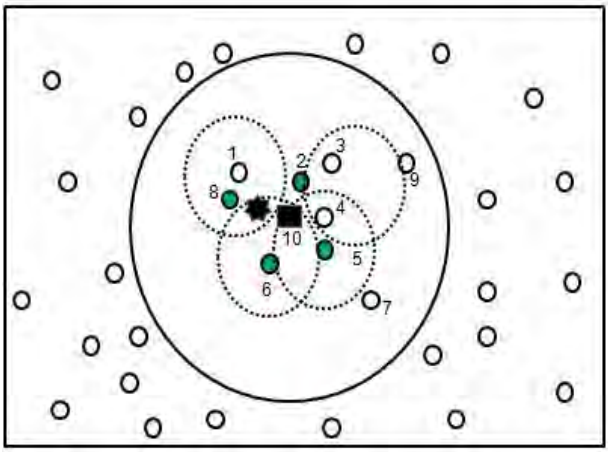

Fig. 3.10: Step 4 of *local search phase*.

Finally the solution is {8, 6, 2, 4} as shown in Fig. 3.10.

3.3 ETRACK in Details

In this section, we discuss about our target tracking algorithm **ETRACK** in details. We consider 3σ region for the detection area of the moving target. To update the state of the target, we have to take new measurement data within the predicted 3σ region.

The distance d_i between the predicted position of the target and the position of the i^{th} sensor node s_i is,

$$d_i = \sqrt{(\hat{X}_x(k+1|k) - s_{i,x})^2 + (\hat{X}_y(k+1|k) - s_{i,y})^2 + (\hat{X}_z(k+1|k) - s_{i,z})^2} \quad ..(3.7)$$

where $\hat{X}_x(k+1|k)$, $\hat{X}_y(k+1|k)$and $\hat{X}_z(k+1|k)$ are the predicted location of the target in x, y, z coordinate and $s_{i,x}$, $s_{i,y}$ and $s_{i,z}$ are the x, y and z positions of the i^{th}sensor node s_i.

ETRACK scheme selects only four *WNs* from N sensor nodes based on low power communication to track the target. And the other sensor nodes are kept in '*sleep mode*'. A (PN)'s sensing area is selected by,

$$\min(d_1, d_2, d_3, \ldots, d_n) \ldots\ldots (3.8)$$

Thus, from this tracking scheme we can select only four sensor nodes among all sensor nodes in the detection area and therefore the processing node does not require extra communicating effort with all other sensor nodes in the area. This results in increasing the energy efficiency.

The selected four sensor nodes provide measurement $z_1(k)$, $z_2(k)$, $z_3(k)$ and $z_4(k)$ to the processing node, and the processing node calculates the fused one measurement $z_{fusion}(k)$ using the expectation of all three sensor nodes measurements. Thus, Equation 3.6 can be modified by as follows:

$$z_{fusion} = \begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \end{bmatrix} = HX(k) + v(k) \ldots\ldots (3.9),$$

where H is a measurement matrix whose parameter is,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \ldots\ldots (3.10)$$

and $v(k)$ is the mean of every measurement noise which is also assumed to be independent with zero-mean, white, Gaussian probability distributions $N(0,\ R(k))$.

The *processing node*(PN) updates the state of the target and the state covariance matrix form the fused measurement $z_{fusion}(k)$. At every time step, the similar procedures continue iteratively, shown in Fig. 3.11.

3.3.1 ETRACK Algorithm

Algorithm 1 describes our tracking algorihtm **ETRACK** algorithm. Initially, all sensor nodes are in sleeping mode. When a moving target enters into the detected region, *PNs* having detection areas which are overlapped with the 3σ region are selected. From *PNs*, four sensor nodes are selected as *WNs.* These nodes track the moving target.

Algorithm 1: ETRACK Algorithm

---

procedure ETRACK ( ) // Energy Efficient Target Tracking Algorithm **begin**
Step-1: Initially, every sensor node is sleeping.
Step-2:
do when a moving target enters
PSKF() // Prediction step of KF
begin
A predicted 3σ region centered at the predicted position of the target;
SPSN(); // Selection of *PN*
Some sensor nodes having the detection areas that are overlapped this 3σ region, are wake-up;
S=Number of wake-up sensor nodes;
for each wake-up sensors **do**
SWSN(*ListSize,MaxIter,sens.energy,sens.power*)//Selecting *WN*s
for k=1,. . . ,MaxIter **do**
begin
CS(*ListSize,sens.energy,sens.power*); //Possible *WN*s
LS(*BestSolutionFound*); // Local Search Phase
US(*BestSolutionFound*); // Update the solution
endfor
return *BestSolutionFound*; //four sensor nodes are selected for tracking
endfor
S=S-4 sensor nodes go to sleep mode and the selected four sensor nodes are still wake-up and send their fused data to the processing node.
end
go to Step 2
end procedure

In Algorithm 2, sensor nodes having detection areas overlapped with the 3σ region are selected as *PNs.* For each *PN*, the distance between the predicted position of the target and the position of the sensor $d_i$is calculated and the minimum distance is returned among them.

Algorithm 2: *Processing Node*(*PN*) Selection

---

procedure *SPSN*( )
S_i = A Sensor node having detection area overlapped with the 3σ region;
for each S_i **do**
    Calculate d_i which is the distance between the predicted position of the target and the position of the sensor S_i;
endfor
return $min(d_i)$;
end procedure

Algorithm 3: Sensing Area Overlapping Detection

---

procedure *Overlap*( ) // The sensing area overlapping detection
begin
$T[S]$= Unmarked; // T is an array of all sensor nodes and initially it is unmarked
for each sensor S_i **do**
    Calculate d_i between the predicted position of the target and the position of S_i;
        if $d_i = r_i + 3\sigma$ **then**
            $T[S_i]$=Marked;
        endif
endfor
end procedure

Algorithm 3 describes the selection of sensor nodes those have overlapped with the 3σ region. In Algorithm 4, four *WNs* are selected based on the remaining energy and communication power consumption. Sensor nodes with more remaining energy and low powered communication participate in tracking the moving target.

Algorithm 4: *WNs* Selection

procedure *SWSN*(*ListSize*, *sens.energy*, *sens.power*) // Selection of *WNs*
begin
K = null; // K denotes a partial solution in this case
PWNSL_size = 4; // Size of *possible working sensor nodes list*
i = 1; // A counter to count the PWNSL size
while (i<= *PWNSL_size*) **do**
    if (S_i.*energy* >= *sens.energy*) && (S_i.*power* <= *sens.power*) **then**
        K=K U S;
    endif
    i++;
endwhile
end procedure

Algorithm 5: Local Search to select *WNs*

Procedure *LS* (*K*)
//*distance*(N_i) denotes the distance between the predicted position of the target and the position of the neighbors of sensor S_i
//*distance*(S_i) denotes the distance between the predicted position of the target and the position of the sensor S_i
begin
L=K; // K denotes solution which is not guaranteed to be locally optimal
  for each sensor element S_i of K **do**
    for each neighbor sensor nodes N_j of S_i **do**
      if (*distance*(N_i) < *distance*(S_i) **do**
        K=K-S_i;
        K=K U N_i;
      endif
    endfor
  endfor
end procedure

From the selected *WNs,* if there are any other better options to select four *WNs* using local search in the detected region. This local search is described in Algorithm 5.

3.3.2 Summary

In this section, we give a flowchart of the whole tracking procedure. Fig. 3.11 describes the summary of our **ETRACK** algorithm.

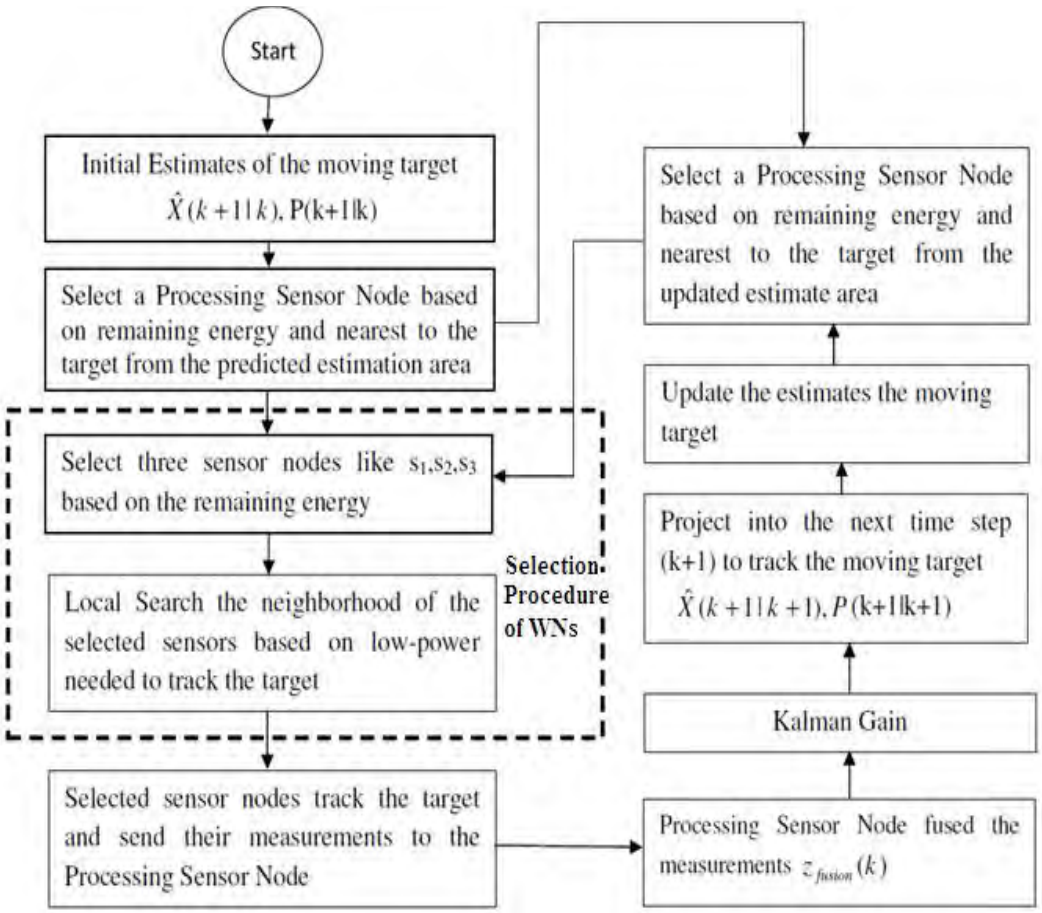

Fig. 3.11: Summary of ETRACK Algorithm.

Chapter 4

Simulation Results

4.1 Overview

In the previous chapters, we described the architecture and the tracking algorithm and the load balancing strategy of a moving target in an underwater sensor network (UWSN). In this chapter, we provide our experimental results. We also compare our simulation results with a popular and widely used tracking method of UWSNs.

4.2 Testbed Description

The simulation of **ETRACK** is run on 1.83 *GHz Intel Dual Core Processor* with $1GB$ memory. The operating system used to run the simulation is *Windows Vista*.

We simulate target tracking using OMNet++ which has an extensible, modular, component-based C++ simulation library and framework, with an Eclipse-based Integrated Development Environment (IDE) and a graphical runtime environment [2]. OMNeT++ provides a discrete event simulation engine, graphical and command-line execution of simulations, and visualization of results. MiXiM [1] is a framework for wireless and mobile networks developed using OMNeT++. We use MiXiM for simulating our underwater WSN application [43].

4.2.1 Sensor Node Architecture in MiXiM

A sensor node structure can be seen in Fig. 4.1 [44]. Based on this generic design, a simulated sensor node has been built as illustrated in Fig. 4.1 [43, 44].

The *Layer* 0 *module* represents the physical layer of a sensor node. It is reponsible for making connection between the node and its neighbors, and forwarding messages from a higher layer to its neighbors, and vice versa.

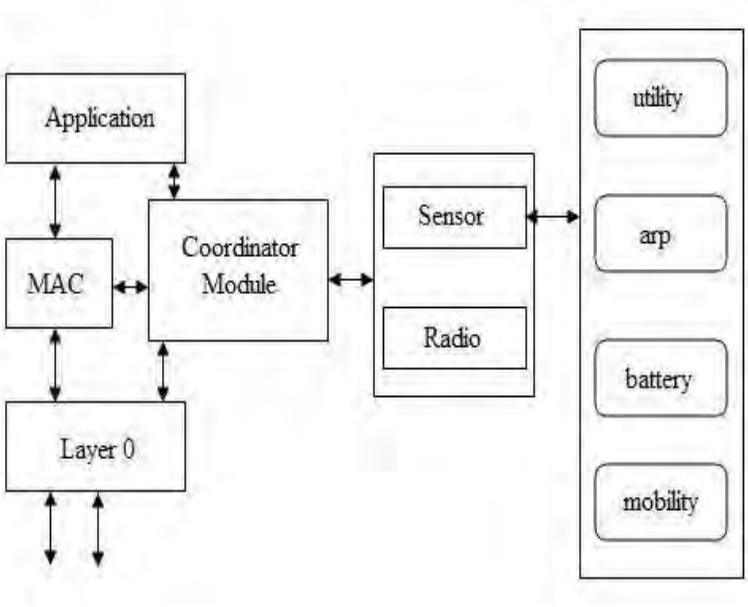

Fig. 4.1: Structure of a simulated sensor node [43, 44].

The *MAC module* represents pre-processing packet layers. It consists of gates (in/out) and queues (incoming queue and outgoing queue). When the queue is full, it deletes some of the oldest messages to make room in the queue for new messages. It helps to evaluate performance of the node.

The *Application module* represents the application layer. Note that, each time after sending a message, the module automatically sends a *DECREASE_ENERGY* message to the *energy module* (through the *coordinator module* to let the module decreases the energy by a number of energy units).

The *Coordinator module* is an interface to connect all modules together. It categorizes an incoming message in order to deliver it to the right module.

The *Sensor module* represents the sensor board in a sensor node. If the *SENSOR_SWITCH* parameter is *ON*=(1), the module consumes energy, therefore, after an interval (timer), the module sends a *DECREASE_ENERGY* message to the *energy module* (through the *coordinator module*).

The *Radio module* represents the radio board in a sensor node. If *RADIO_SWITCH* parameter is *ON*(=1), the module consumes energy, therefore, after an interval (timer), the module sends a *DECREASE_ENERGY* message to the *energy module* (through the *coordinator module*). In case of UWSNs, the *RADIO_SWITCH* parameter is *OFF*(=0).

Besides the above modules, there are the following additional modules in MiXiM [43]:

The *mobility module* is responsible for the movements of a node or an object.

The *battery module* is used for energy related issues. For a sensor node, e.g., the battery drainage due to communication and processing can be simulated.

The *arp module* handles the Address Resolution Protocol (ARP), i.e. the translation between network and MAC addresses.

The *utility module* has two main tasks:

1. Firstly, it provides a general interface for collecting statistical data of a simulation. Using the utility module for statistical data collection only has minimal impact on the performance of the simulation and leaves full flexibility for different analysis methods.

2. Secondly, the utility module maintains parameters that need to be accessed by more than one module within a node.

4.2.2 Simulation Settings

To set up a simulation environment, at first we need to specify some parameters for the construction of the base network. We keep the settings similar for our method and the other tracking method to compare the performance. In Table 4.1, the simulation settings are listed as follows:

Table 4.1: Simulation Settings

Parameter	**Value**
General Settings	
Network Dimension	600m x600m x 600m
Number of Sensor Nodes	150/200
Deployment of Nodes	Uniform random
Node Frequency	20 kHz
Event Settings	
Simulation Type	Discrete-event driven
Target Intrusion Point	User defined
Data Rate	1000 bits/sec
Wave Length	4 meters
Wave Height	100 meters
Wave Period	5 seconds
Energy Model	
Initial Energy	100J
Signal propagation Speed	1500m/s
Power Dissipation in radio transmission	10^{-4}mW/m^2
Energy dissipation rate of wake-up sensor nodes	Active mode: 20mW Sleep mode: 15μW

The communication range for sensor nodes and sink nodes was 60 meters while data channel was set to 5 kbps. Based on the energy consumption model in [5, 31], the least transmission power is required can be expressed as follows:

$$E(d) = P_0 \times d^k \times 10^{d \times \frac{\alpha(f)}{10}} \quad \ldots\ldots (4.1)$$

where d is the distance between transmitting node and receiving node, k is the loss factor that is affected by extension of wave surface, $\alpha(f)$ is an absorption coefficient of frequency where f is the acoustic transmitting frequency.

We simulate a scenario where a target moves randomly within a 3-dimensional

sensing area. We first distribute the sensors uniformly over a field of size 600m x 600m x 600m. The movement pattern of the target follows the random waypoint model wherein the direction of the target is randomly updated at fixed intervals [5].

We assume the time interval T as 1sec. The initial and the estimated state as [5],

$$X(0) = [\ 20 \quad 20 \quad 20 \quad 20 \quad 20 \quad 20\ ]^T \qquad (4.2)$$

$$\hat{X}(0) = [\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ]^T \qquad (4.3)$$

The intensity of the process noise q is 10, and the measurement covariance $R(k)$ is assumed as,

$$R(k) = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix} \qquad (4.4)$$

And the initial state error covariance is assumed as

$$P(0) = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 25 \end{bmatrix} \qquad (4.5)$$

The detection radius of the i^{th} sensor node s_i, r_i is assumed 20m, that is, the detection radius of all sensor nodes is assumed same.

4.2.3 Simulation Metrics

In this section, we present the simulation metrics as performance measures to show the effectiveness of our **ETRACK** algorithm. The simulation metrics are as follows:

1. The *residual energy* is the remaining energy after consuming all nodes of the network. We define *residual energy* as follows:

Residual Energy=Initial Energy - Consumed Energy

2. The *number of dead nodes* determines how many nodes are died after consuming energy to track the moving target. So, the number of less dead nodes indicates the prologated network lifetime.

3. *Tracking trajectory* shows the tracking path of a moving object.

4.3 Simulation Results

In this section, we present the simulation results to evaluate the performance and effectiveness of **ETRACK** algorithm.

A. Residual Energy

We compare the residual energy in our **ETRACK** with the energy of WuS/VMS mechanism [5]. Fig. 4.2 and Fig. 4.3 show the effect on the residual energy of the network when the total number of nodes is 150 and 200 respectively. We see that upto 50 rounds (in Fig. 4.2) and upto 200 rounds (in Fig. 4.3), our **ETRACK** and the existing WuS/VMS give the same residual energy. But as the time increases, our **ETRACK** gives better result compared to WuS/VMS.

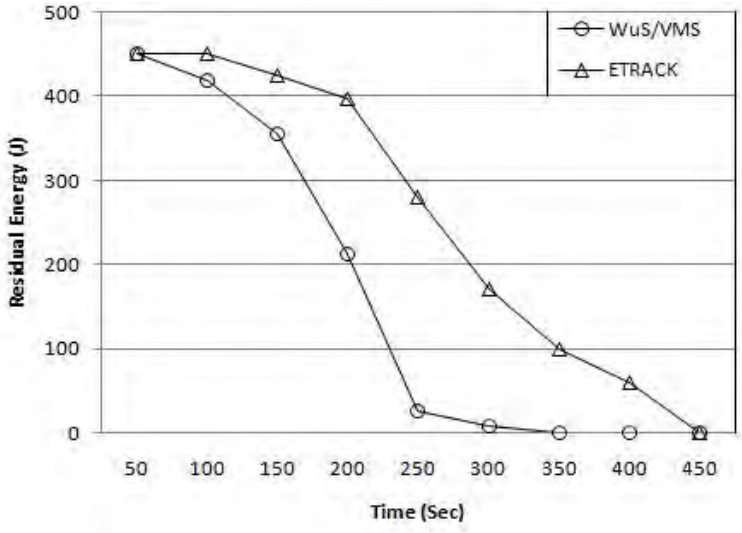

Fig. 4.2: The residual energy of the network (total no. of nodes is 150).

Therefore, Fig. 4.2 and Fig. 4.3 illustrate that, our **ETRACK** is more energy efficient than Wus/VMS mechanism. The reason is that, **ETRACK** keeps only four *WNs* as active nodes for tracking the moving target which save energy.

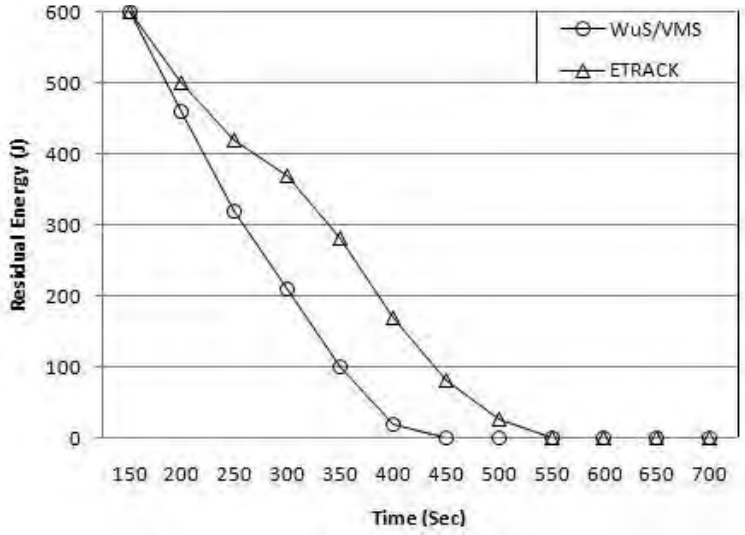

Fig. 4.3: The residual energy of the network (total no. of nodes is 200).

B. The Number of Dead Nodes

Fig. 4.4 and Fig. 4.5 show the number of dead nodes over a time period when the total number of nodes is 150 and 200 respectively. We compare **ETRACK** to WuS/VMS mechanism.

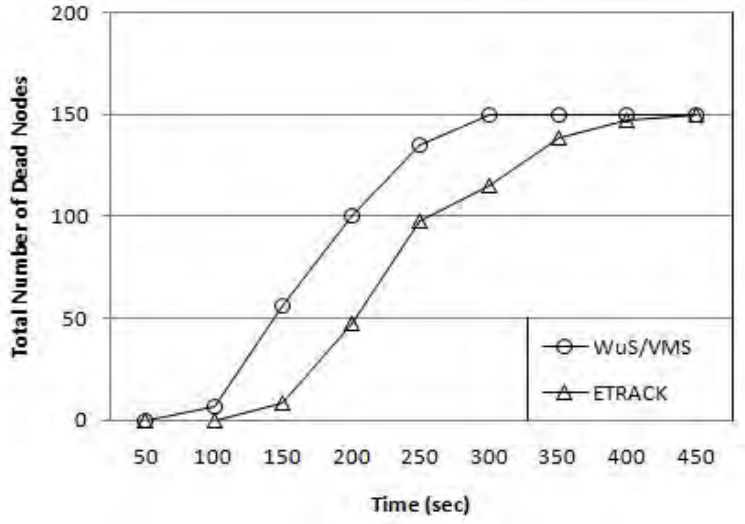

Fig. 4.4: The number of dead nodes in the network (total no. of nodes is 150).

From Fig. 4.4, it is clear that the sensor nodes start dying after 50 rounds and from Fig. 4.5, we see that the sensor nodes start dying after 200 rounds in case

of WuS/VMS. But in **ETRACK**, the sensor nodes start dying after 100 rounds when the total number of nodes is 150 (as shown in Fig. 4.4) and after 350 rounds when the total number of nodes is 200 (as in Fig. 4.5). The number of less dead nodes indicates the prolongated network lifetime. Therefore, Fig. 4.4 and Fig. 4.5 show that in **ETRACK** method the network nodes survive respectively 25% (in average) and 35.26% (in average) than in WuS/VMS.

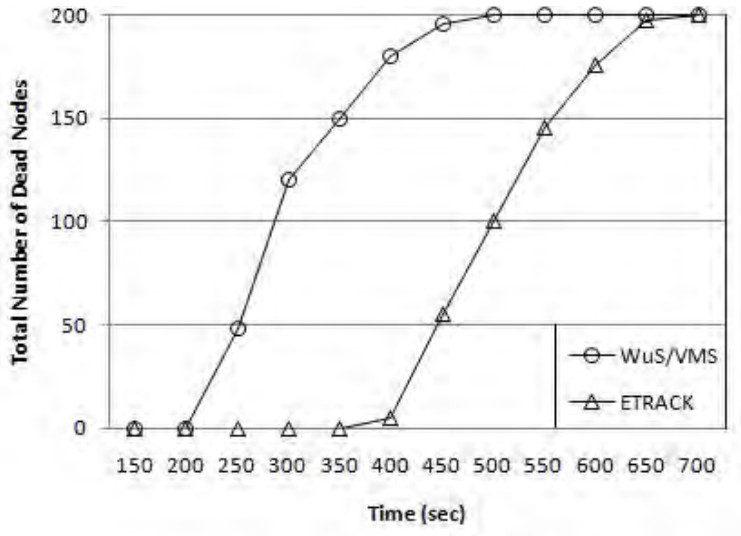

Fig. 4.5: The number of dead nodes in the network (total no. of nodes is 200).

C. Tracking Trajectory

We compare the estimated target position computed by **ETRACK** algorithm with the true target position and also with the estimated target position computed by WuS/VMS [5].

Tracking results of simulation runs are shown in Fig. 4.6 and Fig. 4.7. Here, **ETRACK** achieves higher tracking accuracy than Wus/VMS. The reason is that, **ETRACK** uses 3D Kalman Filter in position estimation and WuS/VMS uses 2D Kalman Filter. Besides, as many nodes die earlier in WuS/VMS (as shown in Fig. 4.4 and Fig. 4.5), the position estimation is badly affected. In case of **ETRACK**, more nodes survive and these nodes compute tracking position for a long time.

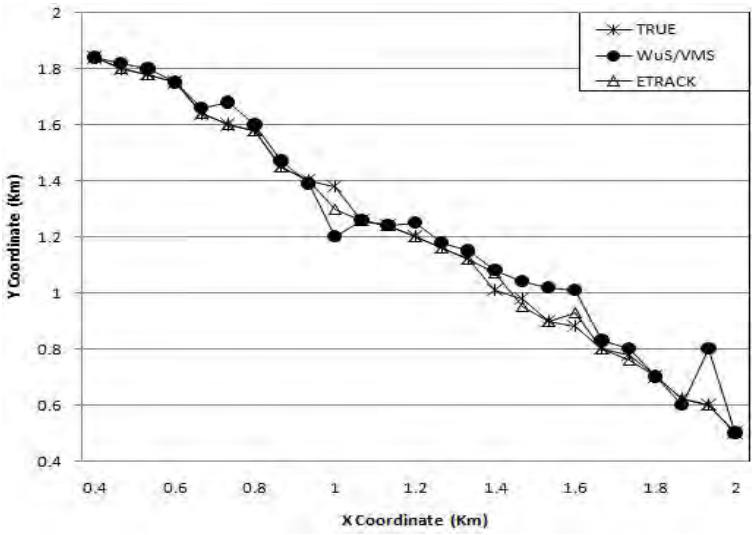

Fig. 4.6: Tracking trajectory (total no. of nodes is 150).

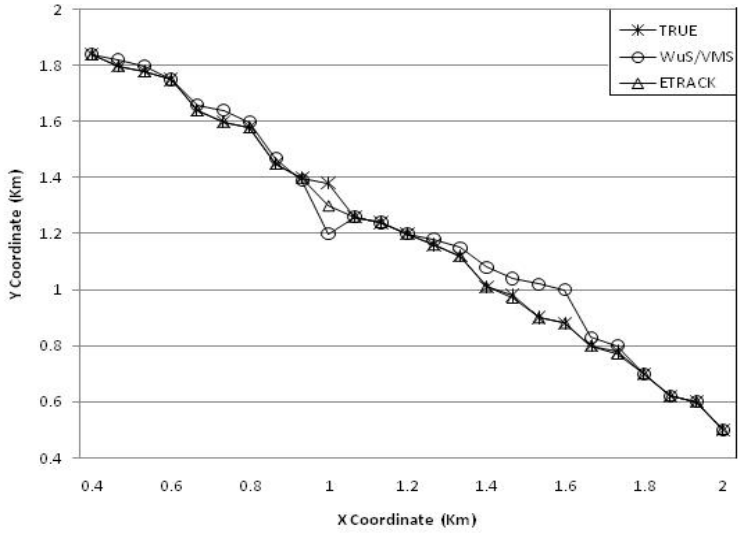

Fig. 4.7: Tracking trajectory (total no. of nodes is 200).

Fig. 4.6 indicates that, **ETRACK** provides 88% accurate of true tracking trajectory and approximately 20% better estimation than WuS/VMS method and Fig. 4.7 illustrates that, **ETRACK** provides 96% accurate of true tracking trajectory and approximately 28% better estimation than WuS/VMS method.

In our method there may be some energy expenditure for more calculation using 3D Kalman Filter to estimate the position of the target than that of WuS/VMS. But we find that as we select *processing node* in a load balanced way, this does not hamper in energy calculation.

Chapter 5
Conclusion and Future Work

5.1 Overview

In this chapter, we draw the conclusion of our thesis followed by some future directions.

5.2 Summary

In this thesis, we study different exiting strategies for target tracking in underwater environment. We identify some major limitations of those techniques. This thesis addresses the problem of energy efficient tracking limitations affecting the network lifetime. We develop a load balanced strategy for target tracking which ensures the energy efficiency as well as the prolongated lifetime of the UWSN.

To handle the challenges of target tracking, we present an algorithm for target detection and selection with keeping a low number of sensors active to detect the target which increases the network lifetime. We simulate our technique with OMNET++ and MiXiM [1, 2, 43, 44]. Simulation results depict that about 36% energy is saved in our method compared to a popular tracking method in UWSNs. We also compare the estimation of the target's position with the true target trajectory, and we find that our detected trajectory matches with the true trajectory.

5.3 Future Work

Underwater sensor networks (UWSNs), being a relatively new field in computing, still has plenty of room for further network research. The scope of our thesis also provides the direction for its exploration. We would like to extend our tracking approach to multiple targets tracking problem. If we want to apply **ETRACK** for tracking multiple targets, we have to provide a distributed framework to handle different velocities and directions of the multiple targets.

References

[1] "\Mixim," Available: http://mixim.sourceforge.net/, visited on: September 25, 2012.

[2] "\Omnet++," Available: http://www.omnetpp.org/, visited on: September 25, 2012.

[3] A. D. Waite," *SONAR for Practicing Engineers*," 3rd edition, Wiley Publisher, 2001.

[4] B. Scheuermann, C. Lochert, M. Mauve, "*Implicit hop-by-hop congestion control in wireless multihop networks*," Ad Hoc Network, vol. 6, pp. 260-286, 2008.

[5] C. H. Yu, K. H. Lee, J. W. Choi and Y. B. Seo, "*Distributed Single Target Tracking in Underwater Wireless Sensor Networks,*" SICE Annual Conference, vol. 1, pp. 1351-1356, August 2008.

[6] D. Eickstedt, M. Benjamin, H. Schmidt, and J. Leonard, "*Adaptive tracking of underwater targets with autonomous sensor networks,*" Journal of Underwater Acoustics, vol. 56, pp. 465-495, 2006.

[7] D. L. Codiga, J.A. Rice, P.A. Baxley, "*Networked acoustic modems for real-time data delivery from distributed subsurface instruments in the coastal ocean: Initial system development and performance,"* Journal of Atmospheric and Oceanic Technology, vol. 21, pp. 331-346, 2004.

[8] D. Pompili and I. F. Akyildiz, "*Overview of networking protocols for underwater wireless communications,*" IEEE Communication Magazine, vol. 47, no. 1, pp. 97-102, 2009.

[9] D. Pompili, T. Melodia and I. F. Akyildiz, "*A CDMA-Based Medium Access Control for Underwater Acoustic Sensor Networks,*" IEEE Transactions on Wireless Communications, vol. 8, no. 4, pp. 1899-1909, 2009.

[10] E. Dalberg, A. Lauberts, R. K. Lennartsson, M. J. Levonen, and L. Persson, "*Underwater target tracking by means of acoustic and electromagnetic data fusion,*" in Proceedings of the 9^{th} International Conference on Information Fusion, vol. 1, pp. 1-7, July 2006.

[11] E. Sozer, J. Proakis, M. Stojanovic, J. Rice, A. Benson, and M. Hatch, "*Direct sequence spread spectrum based modem for underwater acoustic communication and channel measurements,*" in Proceedings of MTS/IEEE Conference Exhibition Ocean Engineering, Science Technology (OCEANS), vol. 1, pp. 228-233, 1999.

[12] G. Holland, N. Vaidya, "*Analysis of TCP performance over mobile ad hoc networks,"* in Proceedings of the 5^{th} Annual ACM/IEEE International Conference on Mobile Computing and Networking, vol. 1, pp. 219-230, 1999.

[13] G.Isbitiren and O. B. Akan, "*Three-Dimensional Underwater Target Tracking with Acoustic Sensor Networks,*" in Proceedings of IEEE Transactions on Vehicular Technology, vol. 60, no. 8, pp. 3897-3906, 2011.

[14] G. Welch and G. Bishop, "*An Introduction to the Kalman Filter,*" UNC-Chapel Hill, TR 95-041, July 2006.

[15] H. Yan, ZJ Shi, J. Cui, "*DBR: depth-based routing for underwater sensor networks*," in Proceedings of the 7^{th} International IFIP-TC6 Networking Conference on Adhoc and Sensor Networks, vol. 1, pp. 72-86, 2008.

[16] I. F. Akyildiz, D. Pompili, and T. Melodia, "*Challenges for Efficient Com-*

munication in Underwater Acoustic Sensor Networks," ACM SIGBED Review - Special Issue on Embedded Sensor, vol. 1, no. 2, pp. 3-8, 2004.

[17] I. F. Akyildiz, W. Su, Y. Sankarasubramanium, and E. Cayirci,"*Wireless Sensor Networks: A Survey,"*Computer Networks, vol. 38, pp. 393-422 , 2002.

[18] I. F. Akyildiz, D. Pompili, T. Melodia," *Underwater Acoustic Sensor Network: Research Challenges,*" Ad Hoc Networks (Elsevier), vol. 3, no. 4, pp. 257-279, 2005.

[19] L. Arienzo , and M. Longo,"*An Energy Efficient Strategy for Target Tracking through Wireless Sensor Netwoks,"* GTTI, vol. 1, pp. 1-8, 2008.

[20] M. Asif, M. Rizal Arshad, and A. Yahya, "*An active contour for underwater target tracking and navigation,*" in Proceedings of International Conference on Man-Machine Systems, vol. 1, pp. 1-6, 2006.

[21] M. Ayaz, A. Abdullah, "*Underwater Wireless Sensor Networks: Routing Issues and Future Challenges,"* in Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia, vol. 1, pp. 370-375, 2009.

[22] M. Ayaz, I. Baig, A. Abdullah, I. Faye, "*A survey on routing techniques in underwater wireless sensor networks,*" Journal of Network and Computer Applications, vol. 34, no.6, pp. 1908-1927, 2011.

[23] M. Erol, LFM Vieira, M. Gerla, "*Localization with Dive'N'Rise (DNR) beacons for underwater acoustic sensor networks,*" in Proceedings of the 2^{th} Workshop on Underwater Networks, vol. 1, pp. 97-100, 2007.

[24] M. Erol, S. Oktug, "*A Localization and Routing Framework for Mobile Underwater Sensor Networks,*" in Proceedings of the IEEE INFOCOM Workshops,

vol. 1, pp. 1-3, 2008.

[25] M. Hahn and J. Rice, "*Undersea navigation via a distributed acoustic communication network,*" in Proceedings of Turkish International Conference on Acoustic , 2005.

[26] M. I. Pettersson, V. Zetterberg, and I. Claesson, "*Detection and imaging of moving targets in wideband SAS using fast time backprojection combined with space–time processing,*" in Proceedings of MTS/IEEE Oceans, vol. 3, pp. 2388-2393, 2005.

[27] P. Xie, et al. "*SDRT: a reliable data transport protocol for underwater sensor networks*", Ad Hoc Networks, vol. 8, no. 7, pp. 708-722, 2010.

[28] Q. Liang and X. Cheng, "*Underwater acoustic sensor networks: Target size detection and performance analysis,*" Ad Hoc Networks (Elsevier), vol. 7, no. 4, pp. 803-808, 2009.

[29] S. Binato, H. F. Jr., and M. G. C. Resende, "*Greedy Randomized Adaptive Path Relinking,*" in Proceedings of the 4^{th} Metaheuristics International Conference, vol. 1, pp. 393-397, 2001.

[30] S. K. Rao, "*Application of statistical estimators for underwater target tracking,"* in Proceedings of IEEE/ION Position, Location and Navigation Symposium, vol. 1, pp. 1040-1044, 2006.

[31] S. Zhou and P. Willett, *"Submarine location estimation via a network of detection-only sensors,"* IEEE Transaction Signal Processing, vol. 55, no. 6, pp. 3104-3115, 2007.

[32] T. Lacey, "*Tutorial: The Kalman Filter*," Chapter-11, Available from: http:

//www.cc.gatech.edu/classes/cs7322_98_spring/PS/kf1.pdf, visited on September 20, 2012.

[33] T. A. Feo and M.G.C. Resende, "*A probabilistic heuristic for a computationally difficult set covering problem*," Operations Research Letters, vol. 8, no. 2, pp. 67-71, 1989.

[34] T. A. Feo and M.G.C. Resende. "*Greedy randomized adaptive search procedures*," Journal of Global Optimization, vol. 6, no. 2, pp. 109-133, 1995.

[35] Underwater Sensor Networks at BWN Laboratory, Georgia Institute of Technology, http://www.ece.gatech.edu/research/labs/bwn/UWASN/, visited on September 20, 2012.

[36] V. Zetterberg, M. I. Pettersson, and I. Claesson, "*Comparison between whitened generalized cross correlation and adaptive filter for time delay estimation with scattered arrays for passive positioning of moving targets in Baltic Sea shallow waters*," in Proceedings of Oceans MTS/IEEE, vol. 3, no. 1, pp. 2356-2361, 2005.

[37] V. Zetterberg, M. I. Pettersson, L. Tegborg, and I. Claesson, *"Acoustic passive 3D imaging method for scattered arrays,"* in Proceedings of Undersea Defence Technology Conference, vol. 1, pp. 1-6, 2006.

[38] V. Zetterberg, M. I. Pettersson, L. Tegborg, and I. Claesson, *"Passive scattered array positioning method for underwater acoustic source,"* in Proceedings of Oceans MTS/IEEE, vol. 1, no. 1, pp. 1-6, 2006.

[39] W. Peng, "*Tracers in the sea*," New York: Eldigio Press, Lamont Doherty Earth Observatory of Columbia University, 1982.

[40] W. Xiao, L. Xie, J. Lin, and J. Li, "*Multi-Sensor Scheduling for Reliable Target Tracking in Wireless Sensor Networks,*" Proceedings of the 6^{th} International Conference on ITS Telecommunications, vol. 1, no. 1, pp. 996-1000, 2006.

[41] YZ Chen, J. He, "*A localization scheme for underwater wireless sensor networks,*" International Journal of Advanced Science and Technology, vol. 4, no. 1, pp. 9-16, 2009.

[42] http://gpsinformation.net/main/errors.htm, visited on: September 20, 2012

[43] "*Simulating Wireless and Mobile Networks in OMNeT++ The MiXiM Vision*", http://www.cs.uni-paderborn.de/fileadmin/Informatik/AG-Karl/ Publica tions/mixim-vision.pdf, visited on: August 24, 2012.

[44] "*Simulating wireless sensor networks with omnet++*", Available: http://sslab. cs.nthu.edu.tw/ welentsai/WSN/SensorSimulator-IEEE-Computers.pdf, visited on: August 24, 2012.

Appendix A

Kalman Filter

I. Kalman Filter

The Kalman Filter is a plant-model based tracking technique that minimizes the mean squared error at each iteration of the algorithm [14, 32]. A Kalman filter is an optimal estimator – i.e, infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive.

II. Computation in Kalman Filter [14, 32]

Assume that we want to know the value of a variable within a process of the form,

$x_{k+1}=\Phi x_k+ w_k$ ..... (A-1),

where x_k is the state vector of the process at time k, (n x 1); Φ is the state transition matrix of the process from the state at k to the state at k+1, and is assumed stationary over time, (n x m); w_k is the associated white noise process with known covariance, (n x 1).

Observations on this variable can be modeled in the form,

$z_k=Hx_k+ v_k$ ..... (A-2),

where z_k is the actual measurement of x at time k, (m x 1); H is the noiseless connection between the state vector and the measurement vector, and is assumed stationary over time (m x n); v_k is the associated measurement error. This is again assumed to be a white noise process with known covariance and has zero cross-correlation with the process noise, (m x 1).

The covariances of the two noise models are assumed stationary over time and are given by,

$Q = \mathrm{E}[w_k {w_k}^T]$ ..... (A-3)

$R = \mathrm{E}[v_k {v_k}^T]$ ..... (A-4)

The mean squared error is equivalent to,

$\mathrm{E}[e_k {e_k}^T] = P_k$ ..... (A-5),

where $P_k$is the error covariance matrix at time k, (n x n).

Equation A-5 may be expanded to give,

$P_k = E\left[e_k e_k^T\right] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$ ..... (A-6)

Assuming the prior estimate of $\hat{x}_k$and $\hat{x}_k{}'$, was gained by the knowledge of the system. It is possible to write an update equation for the new estimate, combining the old estimate with measurement data thus,

$\hat{x}_k = \hat{x}_k' + K_k(z_k - H\hat{x}_k')$ ..... (A-7),

where K_k is the Kalman gain, which will be derived shortly. The term $z_k - H\hat{x}_k'$ in Equation A-7 is known as the innovation or measurement residual,

$\mathrm{i}_k = z_k - H\hat{x}_k'$ ..... (A-8)

Substitution of Equation A-2 into Equation A-7 gives,

$\hat{x}_k = \hat{x}_K^{/} + K_k(Hx_k + v_k - H\hat{x}_k')$ ..... (A-9)

Substituting Equation A-9 into Equation A-6 gives,

$P_k = E[[(I - K_k H)(x_k - \hat{x}_k') - K_k v_k][(I - K_k H)(x_k - \hat{x}_k') - K_k v_k]^T]$ ..... (A-10)

At the point it is noted that $x_k - \hat{x}_k'$is the error of the prior estimate. It is clear that this is uncorrelated with the measurement noise and therefore the expectation may be rewritten as,

$P_k = (I - K_k H)E[(x_k - \hat{x}_k')(x_k - \hat{x}_k')^T](I - K_k H)^T + K_k E[v_k v_k^T]K_k^T$ ..... (A-11)

Substituting Equations A-4 and A-6 into Equation A-10 gives,

$P_k = (I - K_k H)P_k'(I - K_k H)^T + K_k R K_k^T$ ..... (A-12),

where $P_k{}'$ is the prior estimate of P_k.

Equation A-12 is the error covariance update equation. The diagonal of the covariance matrix contains the mean squared error as shown,

$$P_{kk} = \begin{bmatrix} E[e_{k-1}e_{k-1}^T] & E[e_k e_{k-1}^T] & E[e_{k+1}e_{k-1}^T] \\ E[e_{k-1}e_k^T] & E[e_k e_k^T] & E[e_{k+1}e_k^T] \\ E[e_{k-1}e_{k+1}^T] & E[e_k e_{k+1}^T] & E[e_{k+1}e_{k+1}^T] \end{bmatrix} \quad \text{..... (A-13)}$$

The sum of the diagonal elements of a matrix is the trace of a matrix. In the case of the error covariance matrix the trace is the sum of the mean squared errors. Therefore, the mean squared error may be minimized by minimizing the trace of P_k which in turn will minimize the trace of P_{kk}.

The trace of P_k is the first differentiated with respect to K_k and the result set to zero in order to find the conditions of this minimum.

Expansion of Equation A-12 gives,

$$P_k = P_k' - K_k H P_k' - P_k' H^T K_k^T + K_k (H P_k' H^T + R) K_k^T \quad \text{(A-14)}$$

Note that the trace of a matrix is equal to the trace of its transpose, therefore it may written as,

$$T[P_k] = T[P_k'] - 2T[K_k H P_k'] + T[K_k (H P_k' H^T + R) K_k^T] \quad \text{(A-15)}$$

where $T[P_k]$ is the trace of the matrix P_k.

Differentiating with respect to K_k gives,

$$\frac{dT[P_k]}{dK_k} = -2(H P_k')^T + 2K_k (H P_k' H^T + R) \quad \text{(A-16)}$$

Setting to zero and rearranging gives,

$$(H P_k')^T = K_k (H P_k' H^T + R) \quad \text{(A-17)}$$

Now solving for K_k gives,

$$K_k = P_k' H^T (H P_k' H^T + R)^{-1} \quad \text{(A-18)}$$

Equation A-18 is the Kalman gain function. The innovation, i_k defined in Equation A-8 has an associated measurement prediction covariance. This is defined as,

$$S_k = H P_k' H^T + R \quad \text{(A-19)}$$

Finally substitution of Equation A-18 into Equation A-14 gives,

$$P_k = P_k' - P_k' H^T (H P_k' H^T + R)^{-1} H P_k'$$

$$= P_k' - K_k H P_k' = (I - K_k H) P_k' \quad \text{(A-20)}$$

Equation A-20 is the update equation for the error covariance matrix with optimal gain. The three Equations A-7, A-18, and A-20 develop an estimate of the variable x_k. State projection is achieved using,

$$\hat{x}_{k+1}' = \Phi \hat{x}_k \quad \text{(A-21)}$$

To complete the recursion it is necessary to find an equation which projects the error covariance matrix into the next time interval, k+1. This is achieved by first forming an expression for the prior error,

$e'_{k+1} = x_{k+1} - \hat{x}'_{k+1} = (\Phi x_k + w_k) - \Phi\hat{x}_k = \Phi e_k + w_k$ ..... (A-22)

Extending Equation A-6 to time k+1,

$P'_{k+1} = E[e'_{k+1}e^{T'}_{k+1}] = E[(\Phi e_k + w_k)(\Phi e_k + w_k)^T]$ ..... (A-23)

Note that e_k and w_k have zero cross-correlation because the noise w_k actually accumulates between k and k+1 whereas the error e_k is the error up until time k. Therefore,

$P'_{k+1} = E[e'_{k+1}e^{T'}_{k+1}] = E[\Phi e_k(\Phi e_k)^T] + E[w_k w_k^T] = \Phi P_k \Phi^T + Q$ ..... (A-24)

This completes the recursive filter.

Appendix B

Simulation Runs

Table B.1: Experiment results of ETRACK and WuS/VMS (Total no. of nodes is 200)

Time (sec)	ETRACK		WuS/VMS	
	Number of dead nodes	Residual Energy (J)	Number of dead nodes	Residual Energy (J)
150	0	600	0	600
200	0	460	0	500
250	48	320	0	420
300	120	210	0	370
350	150	100	0	280
400	180	20	5	170
450	196	0	55	80
500	200	0	100	25
550	200	0	145	0
600	200	0	176	0
650	200	0	197	0
700	200	0	200	0

Table B.2: Experiment results of ETRACK and WuS/VMS (Total no. of nodes is 150)

Time (sec)	ETRACK		WuS/VMS	
	Number of dead nodes	Residual Energy (J)	Number of dead nodes	Residual Energy (J)
50	0	450	0	450
100	7	420	0	450
150	56	356	8	426
200	100	212	47	397
250	135	25	98	280
300	150	8	115	170
350	150	0	138	100
400	150	0	147	60
450	150	0	150	0

Table B.3: Experiment results of ETRACK for Residual Energy (Average)
[Total no. of nodes is 200]

Time (sec)	Residual Energy (J)	Variance	Standard Deviation
150	600	7.50	2.74
200	500	31.00	5.57
250	420	20.00	4.47
300	370	22.00	4.69

Table B.4: Comparison between Actual track and Predicted track (Average)
[Total no. of nodes is 200]

Runs	Actual Track	Predicted Track	Difference
0	(0,0,20)	(0,0,20)	0.00
150	(200,150,20)	(190,150,20)	10.00
200	(260,180,20)	(250,180,20)	10.00
250	(290,200,20)	(270,190,20)	14.14
300	(300,210,20)	(285,200,20)	18.03
350	(350,240,20)	(320,230,20)	31.62
400	(400,270,20)	(400,280,20)	10.00
450	(420,290,20)	(420,290,20)	0.00
500	(450,310,20)	(455,310,20)	5.00
550	(500,370,20)	(490,360,20)	14.14
600	(540,410,20)	(530,400,20)	14.14

Table B.5: Experiment results of ETRACK and Wus/VMS for the number of dead nodes (Average) [Total no. of nodes is 150]

Time(sec)	WuS/VMS	Average	ETRACK	Average	Standard Deviation (Wus/VMS)	Standard Deviation (ETRACK)
50	0	0	0	0	0	0
50	0		0			
50	0		0			
50	0		0			
50	0		0			
50	0		0			
50	0		0			
50	0		0			
50	0		0			
50	0		0			
100	0	7	0	0	5.66	0
100	0		0			
100	2		0			
100	2		0			
100	5		0			
100	7		0			
100	10		0			
100	11		0			
100	14		0			
100	15		0			
150	45	56	0	8	7.15	5.04
150	48		2			
150	51		3			
150	53		5			
150	55		8			
150	58		11			
150	61		11			
150	63		12			
150	64		13			
150	66		14			

Table B.6: Experiment results of ETRACK and Wus/VMS for the number of dead nodes (Average) [Total no. of nodes is 200]

Time(sec)	WuS/VMS	Average	ETRACK	Average	Standard Deviation (Wus/VMS)	Standard Deviation (ETRACK)
300	115	120	0	0	2.77	0
300	117		0			
300	118		0			
300	119		0			
300	120		0			
300	120		0			
300	121		0			
300	122		0			
300	123		0			
300	124		0			
350	142	150	0	0	4.74	0
350	143		0			
350	146		0			
350	148		0			
350	150		0			
350	150		0			
350	152		0			
350	154		0			
350	154		0			
350	156		0			
400	170	180	0	5	4.59	3.84
400	175		0			
400	179		0			
400	180		3			
400	181		4			
400	182		6			
400	183		7			
400	183		7			
400	184		8			
400	185		11			