CONTINUOUS OBJECTS DETECTION AND TRACKING IN WIRELESS SENSOR NETWORKS

BY

SHEHR YAR KHAN

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

NOVEMBER, 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **SHEHR YAR KHAN** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Dr. Tarek Rahil Sheltami
(Advisor)

Dr. Ahmad Al-Mulhem
Department Chairman

Dr. Ashraf S. Hasan Mahmoud
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Yahya E. Osais
(Member)

16/12/15
Date

DEDICATION

To my Parents, Brother, Sister and Friends

ACKNOWLEDGMENTS

In the name of Allah, the most Merciful, the most Gracious. All praise is due to Allah; we praise Him, seek His help and ask for forgiveness. Peace be upon the Prophet Muhammad, his family, his companions and all those who follow him until the Day of Judgment. .First of all thanks to Allah by whose help I was able to complete my thesis. Then I would like to thank my parents, without their motivation and constant support I would not be able to reach this far in life. It has been a great pleasure and honor to work with Dr. Tarek R. Sheltami. I would like to admire his supervision, suggestions and guidance right from the beginning till the end of this research. His constant motivation helped me to produce quality work. I would like to thank my committee members. Dr. Yahya E Osais and Dr. Ashraf Mahmoud for their useful response, advice and the time they spent reviewing this thesis. I am very obliged to KFUPM for granting fully funded scholarship for my Master's degree. I would also like to appreciate all the support that I received from the Computer Engineering Department in carrying out this research. I am also very thankful to my friends Muhammad Naseer Bajwa, Bilal Saeed, Muhammad Ijaz, Bilal Jehanzeb, Muhammad Musaddiq, Adil Qayyum, Tanvir Hussain, Awais Wahab and Muhammak Irfan Malik for providing the moral support, pleasant atmosphere and never forgettable moments.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARPANET	:	Advanced Research Projects Agency Network
BN	:	Boundary Nodes
COBOM	:	Continuous Object Boundary and Monitoring
CODA	:	Continuous Object Detection Algorithm
CODAT	:	Continuous Objects Detection and Tracking
DARPA	:	Defense Advanced Research Projects Agency
DCSC	:	Dynamic Cluster Structure for Continuous Objects
DEMOCO	:	Detection and Monitoring for Continuous Objects
GAS	:	Grid Based Asynchronous
PT	:	Phenomenon Tags
RN	:	Reporting Nodes

ABSTRACT

Full Name : Shehr Yar Khan

Thesis Title : Continuous Objects Detection and Tracking in Wireless Sensor Networks

Major Field : Computer Networks

Date of Degree : November, 2015

A considerable amount of research is done in the area of target detection and tracking in wireless sensor networks. Most of the research concentrates on a single or multiple targets tracking but limited work has been done in tracking and detection of the continuous objects such as forest fires, bio- chemical materials and mudflows etc. These continuous objects pose new challenges because of their specific nature and characteristics such as changing in size and shape, shrinking and expanding, splitting into multiple objects or merging of multiple objects into one another. Continuous objects tracking and detection require extensive communication which consumes most of the energy of the network and the key issue of today's wireless sensor networks is the energy handling, which is the backbone for enhancing the lifetime of the network. In this thesis a new algorithm known as Continuous Object Detection and Tracking (CODAT) is proposed. CODAT's performance is an improvement in accuracy and efficiency while comparing it with the two known algorithms Continuous Boundary Monitoring (COBOM) and Detection and Monitoring for Continuous Objects (DEMOCO). New data structure for reporting data is introduced. Due to this new data structure, the reported data is sent to the sink more efficiently, which eventually reduces the communication cost of the overall algorithm without compromising the accuracy for reconstructing the boundary of a continuous object at the base station. A concept for differentiating between the holes

in the phenomenon and overall phenomenon changes at the base station level is also introduced which provides additional information to the user at base station level as an added improvement while maintaining the high accuracy and efficiency. The proposed algorithm efficiency is analyzed by simulations. Simulation results show that CODAT outperforms COBOM in terms of efficiency and outperforms COBOM and DEMOCO in terms of accuracy.

ملخص الرسالة

الاسم الكامل: شهريار خان

عنوان الرسالة: تتبع ومراقبة الاهداف المتحركة في انظمة الاستشعار اللاسلكية

التخصص: شبكات حاسوبية

تاريخ الدرجة العلمية: نوفمبر 2015

هذة الايام العديد من الدراسات تم اجرائها في مجال اكتشاف الاهداف ومراقبتها في انظمة الاستشعار الموجودة. العديد من هذة الدراسات ركزت على تتبع الاهداف الثابته ولكن لا يوجد دراسات كثيرة ركزت على تتبع ومراقبة الاهداف المتحركة مثل انتشار الحرائق في الغابات, انتشار الطين داخل بحيرة وانتشار الغازات الكيميائية. هذة الاهداف المتحركة ادت الى العديد من التحديات الجديدة نظرا لمعاييرها المتغيرة باستمرار وبطريقة عشوائية في بعض الخصائص مثل: التغير في الحجم والشكل, مقدار التقلص والتمدد, انشطارها الى العديد من الاجزاء, او تجمع اكثر من جزء مع وحدة واحدة. عملية تتبع ومراقبة هذة الاجسام المتحركة تحتاج الى عملية اتصال اضافية بين وحدات الاستشعار والتي بدورها تستهلك الكثير من الطاقة, علما بأن الهاجس الاكبر في انظمة الاستشعار اللاسلكية يعود الى استهلاك الطاقة المحدودة في وحدات الاستشعار داخل النظام, وفي حال تم التقليل من استلاك الطاقة في النظام فان هذا النظام سيدوم لفترة اطول ويمكن الاستفادة منه بشكل اكبر. في هذة الرسالة, تم تطوير خوارزمية جديدة تعرف ب CODAT. وهذة الخوارزمية مجدية اكثر من الخوارزميات الحالية المطبقة لمراقبة الاهداف المتحركة COBOMو DEMOCO . تم تطوير نظام جديد لنقل البيانات بين وحدات الاستشعار, وبناء على هذا النظام البيانات المنقولة للمستقبل سيتم نقلها بفعالية اكبر وبالتالي هذا سوف يقلل من الطاقة المستهلكة في النظام ككل من دون التاثير على اي خاصية من خصائص الاجسام المتحركة عند اعادة عملية بنائها في المحطة الاساسية. تم تطوير مفهوم جديد للتفريق بين الفجوات في الاجسام المتحركة وبين عملية انكماش او تمدد الاجسام المتحركة. تم التكد من كفائة هذا النظام من خلال برامج المحاكاة. وكنتيجة لهذة الدراسة وجدنا ان الخوارزمية الجديدة CODAT مجدية اكثر من الخوارزميات الحالية المطبقة لمراقبة الاهداف المتحركة COBOMو DEMOCO خاصة في انظمة الاستشعار الكبيرة.

CHAPTER 1

INTRODUCTION

Due to the advancement in micro-electro-mechanical systems (MEMS) and wireless technologies that we have been able to design small, low cost and smart sensors. In Wireless Sensor Networks (WSN), sensor nodes are densely deployed into the environment. They collect information about the surrounding environment, process it and transmit to the sink which is connected to task manger node through internet or satellite [1].

The research in wireless sensor networks traces back to the program known as the Distributed Sensor Networks (DSN) in around 1980 at the (DARPA). By that time, the (ARPANET) had been functioning for many years and in different universities and research institutes, it had about 200 hosts [2]. In 1998 a new era of research started in wireless sensor networks which gained lot of attention and international involvement. This included new networking techniques and network information processing for dynamic ad-hoc environments and attention turned towards the resource constrained sensor nodes. With the advancement in technology, sensor nodes are becoming compact in size (pack of cards to dust particles) and cost efficient so many civilian applications of sensor networks such as environmental monitoring and body sensor networks have become common.

1.1 Wireless Sensor Networks

In WSNs [1], each sensor node performs tasks such as processing and sensing independently. Sensor nodes communicate with each other so that sensed information is forwarded to CPU or to perform local coordination such as data fusion. The Mica2Mote of Crossbow Technology [3] and Z1 mote from Zolertia are widely used sensor node platforms.

1.1.1 Hardware Platform

The components in a sensor node usually comprise of an embedded processor, radio transceiver, external memories and internal memories, power source and one or more sensors.

1.1.1.1 Embedded processor

In a sensor node, tasks scheduling, data processing and functionality control of other components is performed by an embedded processor. Microcontrollers, Digital Signal Processors (DSP) [4], Field Programmable Gate Arrays (FPGA) [5] and Application Specific Integrated Circuit (ASIC) [6] are different categories of embedded processors that are used in wireless sensor nodes. The Microcontroller is most widely used for the sensor nodes because of its ability to integrate and connect with different types of devices. For example 8051 Microcontroller is used by CC253 development board manufactured by Chipcon Texas Instruments [7] and ATMega 128L Microcontroller is used by Mica2Mote Platform provided by Crossbow Technologies.

1.1.1.2 Transceiver

A sensor node communicates wirelessly through a transceiver. Different wireless transmission media include Radio Frequency (RF), Infrared and Laser. In most WSN applications, RF communication is used because of its nature. The operational states of a transceiver are, Receive, Transmit, Idle and sleep. Two kinds of RF radios are used by Mica2Mote [8], ChipconCC1000 and RFM TR 1000. The transmission range of Mica2Mote for outdoors is about 150 meters.

1.1.1.3 Memories

A sensor node includes memories such as in-chip flash memory, ram as well as external flash memory. For example Mica2Mote [8] has ATMega128L microcontroller [9] having 128-Kbyte flash program memory and 4-Kbyte static RAM Furthermore, external memories for Mica and Mica2Mote can be provided 4-Mbit Atemel45DB041B serial flash chip.

1.1.1.4 Power Source

Sensing, communication and data processing consume power in a sensor node. Data communication requires more energy compared to sensing and data processing. Batteries or capacitors are used for power storage. Batteries are the major source of power supply for sensor nodes. For example 2AA batteries are required for Mica2Mote [8]. In WSN operation, minimizing the energy consumption is of key importance due to limited capacity of batteries. In preliminary research, energy harvesting techniques have also been conducted for WSN to remove energy constraints [10]. These techniques convert ambient energy (solar, wind) into electrical energy.

1.1.1.5 Sensors

A sensor can measure such as pressure, temperature, humidity etc. The sensor senses a continual analog signal and analog to digital converter digitizes and sends this digitized signal to an embedded processor for further processing. The attached sensors should also be small in size and consume very low energy because the sensor node is a microelectronic device with very limited source of power. Several different types of sensors can be integrated in a sensor node.

1.1.2 Operating System

The main functionality of any operating system is that it manages both the software and hardware resources and provides common services to most of the programs. Operating systems for WSN are mostly less sophisticated compared to the general purpose Operating systems because of the particular requirements of resource limitations of hardware platforms as well as WSN applications.

One of the first operating system specifically designed for WSN is Tiny OS [11] which is capable of fast innovation and optimization and can be implemented with least code size which is the main requirement for WSN due to limited resource constraints. A number of other OSs can also be used such as SOS [12] which works on an event driven mechanism for mote class sensor nodes. The most dynamic functionality of SOS is its support of loadable modules. Smaller modules build the whole system. SOS also supports memory management to support inherent dynamism in its module interface. It is no longer actively developing.

Contiki [13] for WSN sensor nodes is an open source and lightweight operating system. It uses an event driven kernel and is portable. Preemptive multitasking is supported by Contiki and can be extended to individual process levels. 2 KB and 40 KB of RAM and ROM respectively is consumed by Contiki's configuration. Various features that are supported by this OS are preemptive multithreading, multitasking kernel, GUI, TCP/IP, IPv6, telnet client, a web browser, web server, virtual network computing and a screen saver. Supported sensing platforms are Tmote, AVR series MCU [14]. Contiki provides simulation platform known as Cooja Simulator.

MANTIS [15] which is abbreviation of The MultimodAl system for NeTworks of In-situ wireless Sensors offers a multithreaded operating system for WSN environment. MANTIS provides an energy efficient and light weight OS offering 500 bytes of footprint including kernel, network stack and scheduler. MANTIS's support of portability across multiple platforms is the main feature i.e after testing applications on PC, can be ported to sensor platforms. It uses Dynamic programming for the remote management of sensor nodes. The supported language for application development is C. The supported sensing platforms are MICA2, MICA Z and Telos.

Nano-RK [16] is a real-time OS for WSN which supports preemptive multitasking. The various features supported by Nano-Rk are multi-hop networking, multi-tasking, priority-based scheduling, extended WSN lifetime, resource usage limits for applications and a small footprint. It uses 2kb and 18kb of RAM and ROM respectively. Sensors, CPU and Network Bandwidth reservations are supported by Nano-RK. Soft and Hard real-time applications are supported through real-time scheduling algorithms like rate harmonized scheduling and rate monotonic scheduling. The supported sensing platforms are FireFLY[17] and MicaZ.

University of Illinois developed a Unix-like operating system for WSN called as Lite-OS [18]. The main aim for the design and development of LiteOS was to provide a unix-like environment for the system programmers along with accustomed programming paradigm, hierarchical file system and object-oriented programming support in the form of Unix-like shell and LiteC++. LiteOS has a small enough footprint that can support MicaZ with 128 Bytes of ROM, 4KB RAM and 8MHz CPU. It consists of three components: LiteFS, LiteShell and Kernel. The supported sensing platforms are MicaZ, and AVR series MCU [14].

1.1.3 Networking

A wireless sensor network consists of sensor nodes having sensing, computing and wireless communication functionalities. The sensor nodes are distributed and scattered. The sink nodes collect data from the whole network and send data to the user via infrastructure network. Figure 1-1 shows operation of WSN. Two kinds of WSN topologies are shown. Hierarchical network topology in which sensor data is collected and routed to the sink via fixed or mobile relays. Other is flat network topology in which sensor nodes are used as routers and send data to the sink through multi-hop routing.

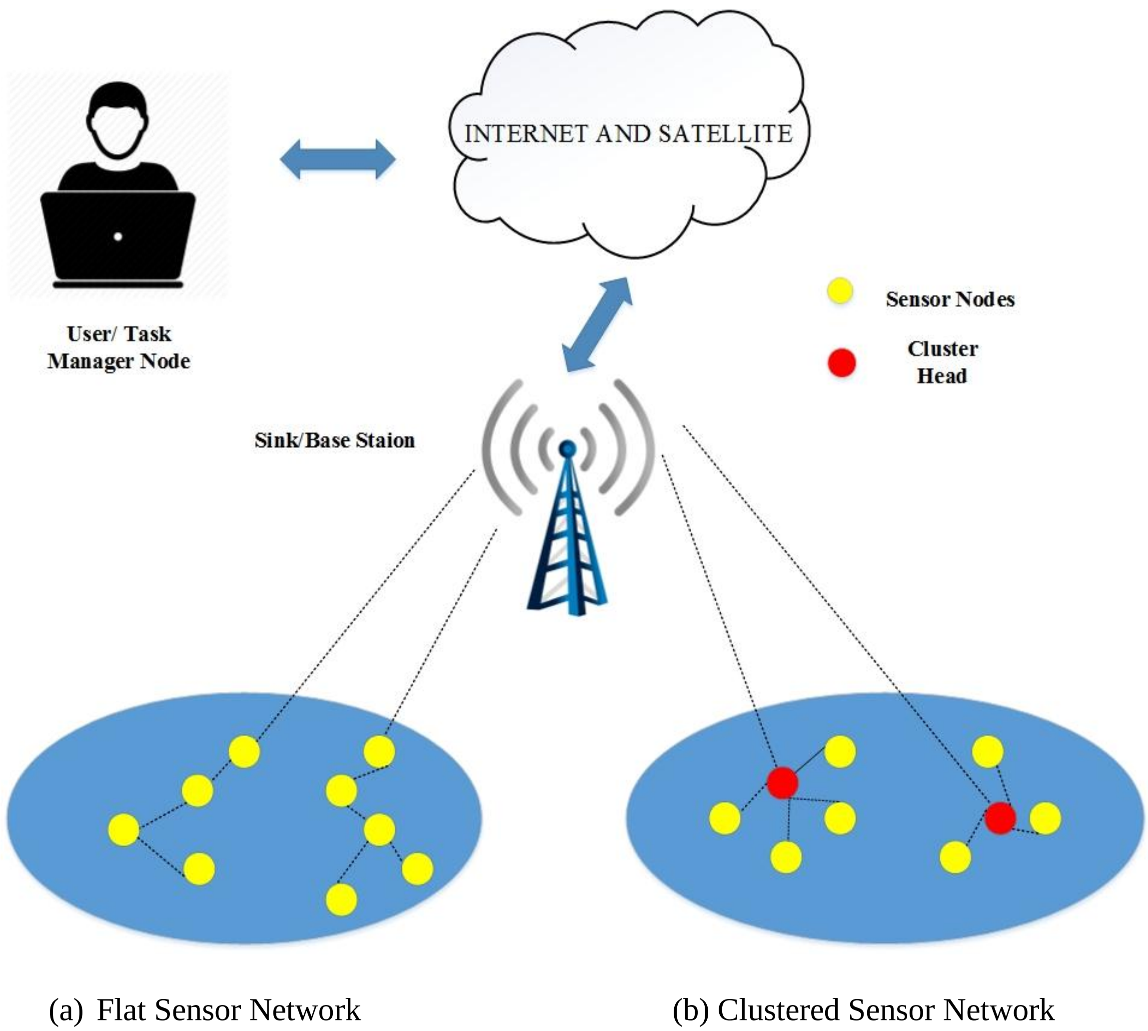


Figure 1-1: Wireless Sensor Network's operation

1.1.4 Protocol Stack

The sink, cluster head and sensor nodes use the protocol stack as shown in Figure 1-2**.** The protocol stack for sensor networks is almost identical to that of traditional network protocol stack. The layers of WSN protocol stack include application, transport, network, data link and physical layer. The tasks of carrier frequency generation, signal detection, frequency selection, data encryption and modulation are performed by the physical layer. The data link layer is responsible for the point-to-point and point-to-multipoint connectivity in communication network. Its task includes data frame detection, multiplexing of data streams, error control and medium access. The network layer routes the data to the desired destination. Some of the key features that are required in designing the WSN network layer are power efficiency, data aggregation and data centric communication. The data flow is maintained by the transport layer. The various types of application software can be used on the application layer sensing upon the sensing tasks and capabilities.

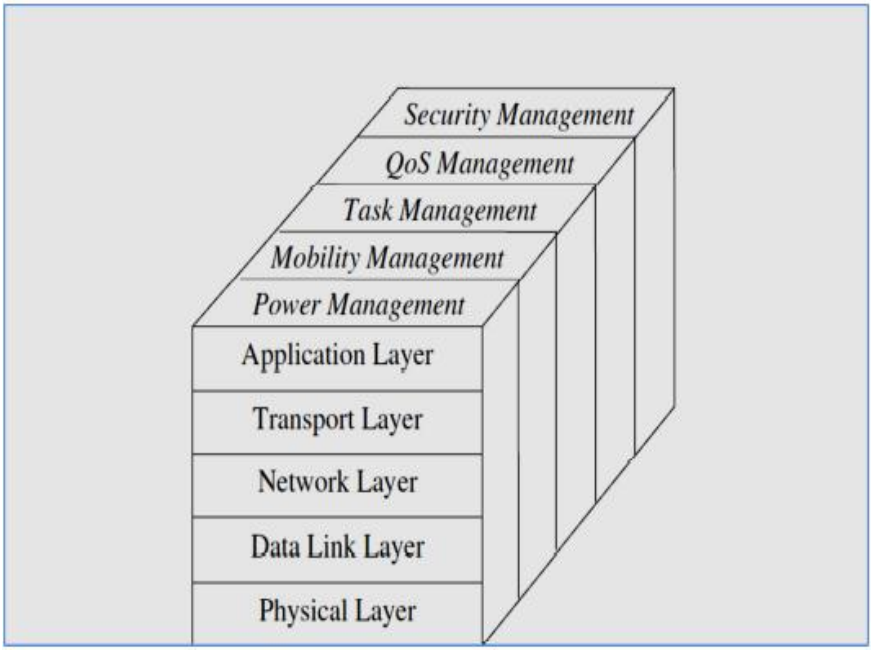


Figure 1-2: Protocol Stack of Wireless Sensor Network

The management planes for the WSN are power, task, mobility, security management planes and Quality of Service (QoS) [1]. These management planes are responsible for efficient functionality of any Wireless Sensor Network. For low power consumption, power management plane is responsible and its functionality can also be turned off for the energy preservation. For maintenance of data route to the sink, mobility management is responsible as it detects and registers the movement of sensor nodes. The balancing between the tasks of sensing and the data aggregation and routing for the sensor nodes is the responsibility of task management plane. The real-time requirement with respect to data services can be fulfilled by the QoS management plane [19]. The tasks of the QoS management plane include error control, fault tolerance and performance optimization for particular QoS metrics. The process of monitoring, managing and control of security related behavior of the network is known as the Security management. Its basic functionality is the restriction of access points to sensitive data. Its other functionalities include encryption, authentication and intrusion detection. Any networking protocol for WSN should be able to address these all five management planes.

1.1.5 Applications

Wireless sensor networks were originally developed for military applications. These military applications include randomly distributed and self-organized WSN for surveillance in battlefield and weapon's micro sensors for stockpile surveillance [20]. WSNs are also used for detection of submarines by implementing large scale acoustic systems in oceans for surveillance. Due to the low cost of sensors and communication equipment now a days, many other civilian applications for WSN have also emerged which include the following

1.1.5.1 Environmental Monitoring

Wireless sensor networks can be used to monitor the environment that includes forest surveillance, animal tracking, weather forecasting and flood detection. For applying sensor networks, environmental monitoring is the most suitable candidate because of various variables to be monitored such as temperature, humidity etc. In Norway [21], University of Southampton researchers have developed a glacial monitoring system using WSN. Sensor nodes are installed within ice and data is collected wirelessly. The EPFL researchers deployed an outdoor WSN for the prevention of avalanches and accidental deaths [22]. The system is installed on the high mountain path between Italy and

Switzerland and sends spatially dense measurements to the Risk management department of Swiss authorities.

1.1.5.2 Health Monitoring

The patients and medical resources in a hospital can be monitored and tracked by embedding WSN in the hospital building. Now a day's special kind of sensors are used for remote monitoring. These sensors collect data such as blood pressure, body temperature, electrocardiograph (ECG) and send to the doctors for further examination and analysis. A specific sensor network known as the Body Sensor Network (BSN) is formed after these sensors are worn or implanted in the body. The BSN have revolutionized the today's health care system due to low cost, efficient and real-time monitoring of medical records via internet. One of the first research in the field of health care monitoring system was conducted in Imperial College of London where researchers developed BSN node and development kit [23].

1.1.5.3 Traffic Control

Sensor networks are used for some time for traffic monitoring and control. For traffic lights control and vehicle detection, sensors are either buried or installed overhead. Normally video cameras are also used for monitoring of traffic in congested areas. The traditional sensors networks are expensive and were installed only in critical areas. WSNs can completely change the traffic control and monitoring system using low cost sensors that can be installed on the road side, in the parking lots and in cars. One of the company Streetline Inc. [24] specializes in the sensor technology to assist drivers to find the empty spaces in parking lots and avoid traffic congestion. These type of implementations can significantly improve the traffic management in congested cities as well as environment due to low carbon emissions.

1.1.5.4 Industrial Sensing

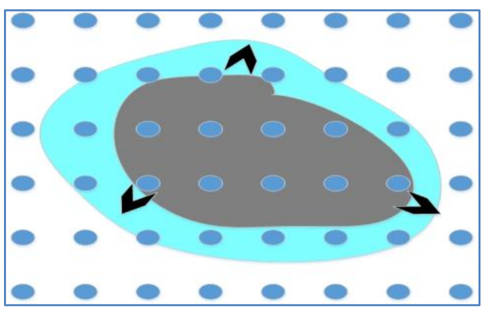
Most of the unplanned downtime in industry occurs due to aging equipment. According to US ARC Advisory group, almost 5% production is lost due to unplanned downtime in North America. The WSN network can be used efficiently to monitor the aging equipment in industries as well as safety of the personnel. These sensor nodes can be implanted in the machinery and can collect the data about safe operation of the equipment. In oil and gas industry, the main problem is the aging of tanks and pipelines. The inspection and monitoring of these huge pipelines and tanks is manual process which is unreliable and extremely costly as well as time consuming. The wireless sensors can installed to tackle this problem. These sensors can cost effectively and reliably monitor the health of the pipelines, tanks and other machinery before huge catastrophic accidents happen.

1.1.5.5 Infrastructure Security

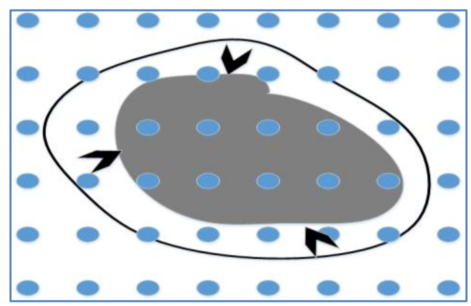
The WSN can be used for the application of intrusion detection and prevention for large infrastructure sites. The sensor networks can be deployed to protect the facilities such as airports, power plants, military bases etc. A large scale network of video, acoustic and other various sensors can be implemented, an initiative of Shanghai Pudong International Airport. Similar kind of WSN has also been deployed in the Shanghai china Expo 2010 for intrusion detection and prevention of critical sites.

1.2 Phenomenon Detection in WSN

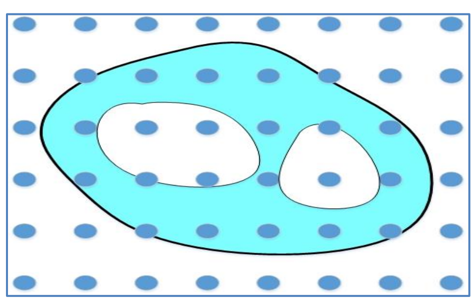
A considerable amount of research is done in Object Tracking in Wireless sensor Networks (OTSNs) [25]. In OTSNs, sensor nodes are deployed over a monitored area with geographical boundaries that are predefined. The base station issues commands and collects data of interests and acts as an interface between applications and OTSN. Sensor nodes track the objects that intrude its area of detection and status of the object is reported to the base station. This application of object tracking with ability to detect anytime and anywhere became only possible due to advancement in development of embedded processors technology and low cost of sensor nodes. The OTSN has wide range of applications including business, military, environmental safety etc.



(a) Phenomenon Expansion



(b) Phenomenon Shrinkage



(c)Holes inside the Phenomenon

Figure 1-3: Different behaviors of Phenomenon

Traditional object detection and tracking focuses on one or several individual objects, such as intruder, tank, and vehicle [25] [26] and [27]. Recently, continuous objects spreading in a very large region, such as diffused noxious gas, biochemical, chemical liquid, and forest fire, are considered [48]-[55]. Compared with individual objects, continuous objects can diffuse, increase in size, or split into multiple continuous objects as shown in Figure 1-3. Moreover, a continuous object usually has a size much larger than that of an individual object, which implies that there should be a lot more event nodes in continuous object detection and tracking. If all event nodes report their location information to sink nodes, it will result in extremely high traffic load. An event node is defined as a reporter if it is required to report its location information to sink nodes. Thus, the main challenge in designing continuous object detection and tracking protocol is to reduce the number of reporters. Therefore, individual object detection and tracking protocols are not applicable to continuous object detection and tracking.

Because of sensor's ability to detect and monitor anytime and anywhere, continuous object (or phenomenon) tracking in sensor networks can be used for monitoring fire outbursts, nuclear explosions and dangerous bio-chemical diffusion .To monitor such spatial phenomena, it is necessary to understand the possible evolving situation of the phenomenon. These shapes can emerge when one phenomenon shrinks or expands, two phenomena merge into each other, one phenomenon splits into two or more new phenomena, and one or two holes appear or disappear in phenomenon.

1.3 Problem description

Continuous Phenomenon detection challenge is different from target detection due to random changes such as expansion, shrinkage, splitting and merging. Our main aim is to detect and track the boundary of the continuous object efficiently and accurately.

1.4 Research Objectives

- Extensive literature review of recent protocols and algorithms for continuous objects tracking and detection.
- Evaluate different algorithms in detection and tracking of the moving phenomenon.
- Choose two candidate algorithms and an improved algorithm might be proposed.
- Simulate and evaluate the proposed algorithm.

1.5 Research Methodology

Following methodology is used to achieve the research objectives:

- **Simulation of existing algorithms**

The existing algorithms are critically analyzed for efficiency and accuracy in detection and tracking of Continuous objects. Candidate algorithms will be chosen and simulated on the network platforms such as Java… etc.

CHAPTER 2

LITERATURE REVIEW

The research in detection and tracking of individual or multiple targets is ample. These targets might include animals, vehicles and persons etc. that come inside sensor network region. For habitat monitoring, applications for wireless sensor networks are discussed by Mainwaring et al [28].

2.1 Individual and Multiple Target Tracking Schemes

A number of techniques are proposed for identifying targets in [29], [30] and [31]. It has been discussed by Krishnamachari et al in [32] that how to track and detect an object via three sensor nodes. Similarly different signal processing techniques are discussed and classified in [33]. Information security is utilized to predict future sensing actions in [34] by Zhao et al.

Similarly another approach for target tracking was discussed in Dynamic Convoy Tree-Based Collaboration (DCTC) [35] and [36]. The tree structure described in DCTC is known as convoy tree. Addition and pruning of sensor nodes enable dynamic configuration of the tree as the object moves within the region of deployed sensor nodes. Prediction-based Energy Saving (PES) scheme and Dual Prediction-based (DPR) scheme are described in [37] and [38] respectively by Xu et al. for Object Tracking in Sensor Networks. Intelligent prediction of object movement is done via sensor nodes as well as the base station. In [39], for acoustic tracking a dynamic clustering scheme with light weight and decentralized mechanism was proposed by Chen et al. Sensor cluster is used for acoustic tracking. In this proposed approach a Cluster Head (CH) identifies the strength of an acoustic signal. When it exceeds its predetermined threshold, this CH broadcasts a message with the invitation to the sensors in its vicinity to join the cluster and provide their sensing information. This CH then identifies the acoustic target via localization techniques after it receives sufficient information from newly joined sensor nodes.

2.2 Continuous Objects Tracking Schemes

The above mentioned schemes were specifically designed for the detection and tracking of individual targets e.g. animals, people, vehicles etc. In many cases it becomes a necessity that a phenomena or a continuous object that spreads on a large area is tracked. Such type of phenomena might include toxic gases, wild fires, Oil spills etc. These continuous objects are different from multiple or individual targets in a manner that they are not discretely distributed over specific area. They are continuously distributed over a large region and can cover a massive area. They can usually diffuse, can change their shape, merge into one another and can split into comparatively smaller continuous objects.

2.2.1 Isobar Based Schemes

An Isobar scheme was proposed by Hellerstein et al [40]. In this scheme for mapping purposes, spatially correlated data aggregation method was used. The sensor nodes with the same readings status are collected by the Isobar into polygons as soon as the information from the nodes flow towards the sink. This scheme reduces the transmitted data. The disadvantage of the proposed approach is that every node has to participate in the collection of the data and pass on to the parent node, which is not very efficient mechanism.

2.2.2 Isolines Based Schemes

For the solution to the problem found in Isobars, Solis and Obraczka [41][42] devised a scheme of isolines. By this scheme if the nodes detect an isoline between itself and the surrounding neighbors only then it will report to the sink otherwise there is no need to report to the sink. This isoline based technique sends less bytes to the sink compared to Isobars However, the disadvantage of Isolines is that in each sampling round, almost 50% of the nodes report data to the sink. This causes too much communication cost. If the sensor nodes are deployed in dense settings then only the sensor nodes that detected the Isolines report to the sink.

A scheme similar to Isoline was described in [43] which is a combination of spatial as well as temporal suppressions. Spatial suppression is defined as when a node has similar reading compared to neighbors, it does not need to report data to sink. Temporal suppression is defined as when a nodes' reading remains same compared to previous time period, it does not need to report data to the sink.

2.2.3 COCH

Another algorithm known as Constraint Chaining (COCH) was proposed by Silberstein et al in [44]. This scheme makes use of spatial and temporal suppressions in very efficient manner. According to this algorithm only one node on each side of boundary of the continuous object report data to the sink. COCH performance is good in case the shape of the continuous object does not change and it moves regularly. But its performance is considerable reduced in case the object's shape changes irregularly. In that case the number of nodes reporting data to the sink increases costing more energy. Chintalapudi and Govindan [45] tried to address the problem of boundary detection. In this scheme enough data is reported so that the sink can later on construct an accurate boundary. Since all the boundary nodes report data to the sink, the communication cost becomes very large. A fault tolerant algorithm based on data mining techniques was proposed in [46] by Ding et al. The algorithms in [45] and [46] have not mentioned anywhere that how the nodes would report data to the sink. They also did not take into account the situation where continuous object changes shape. It would cost too much energy if all the nodes report their data to the sink simultaneously.

2.2.4 Snapshot Queries

To solve the high cost of reporting, Snapshot Queries [47] proposed an algorithm in which representative nodes will report data on behalf of neighboring nodes. Some queries are ignored by the nodes due to which efficiency and accuracy of Snapshot Queries is considerably reduced. For the reduction of errors, update messages are sent at definite intervals and due to this approach extra energy is consumed by Snapshot Queries and hence making it inefficient for the use of continuous boundary changes.

2.2.5 DCSC

For detection and tracking of continuous objects, a dynamic cluster structure is proposed in DCSC (Dynamic Cluster Structure for Continuous Objects detection and tracking) [48]. If the emergence of an object is detected in the sensor's local area at a current time slot then it means that object's boundary has moved through its area of detection in previous time slot and the object's boundary may be close to the sensor. Then communication takes place between the sensor and its one hop neighboring sensors to query if the object is also detected by the neighboring sensors as well. If there is no detection of object by the neighboring sensors then the sensor becomes the boundary sensor. Similarly if the disappearance of object is detected by the sensors in the current time slot then the boundary of object must have moved through its area of detection in the previous time slot. Then sensor node enquires its one hop neighbors about the detection of the object. If the neighboring sensor nodes can detect the object then those neighboring nodes become the boundary sensor nodes. The neighboring nodes in DCSC [48] reply back to the node that inquired whether they have different reading or same reading compared to the inquirer node and this process adds to the communication cost. Cluster is formed after the selection of Boundary Nodes (BNs). However, the description of cluster formation is not very clear in [48].It is also arguable that cluster formation is not suitable when the purpose of its formation was to save energy because considerable communication is required to form clusters which can cause delay in application and hence it is not suitable for real time monitoring of unexpected diffusion or drift such as explosions etc. Moreover all the BNs along with the cluster heads are directly or indirectly involved in routing the data to the sink which can increase the precision and accuracy in identifying the shape of the boundary but at the expense of high traffic delays and overheads.

2.2.6 CODA

In CODA (Continuous Object Detection Algorithm) [49], a dynamic plus static clustering protocol for continuous object detection is proposed. At the time of deployment, static clusters are formed and it is assumed that Cluster Head (CH) of every static cluster assigns a cluster ID to that cluster. After the joining of new node to the cluster, the node exchanges messages with CH and during this process new node is conveyed about the cluster ID. An assumption is made that every sensor node is knows its location and this information is sent to Cluster head when the cluster is joined for the very first time. After knowing the location of all the nodes in a cluster, the CH determines boundary nodes and sends messages to notify them that they are the boundary sensor nodes of that particular cluster.

Whenever an object is detected, control messages are used for the transmission of detection information to CH. Control messages are of two types

Sense message which is used only by SIs (Static Cluster inner sensors) and once the SIs detect target object, sense message is sent to CH

Report message is only used by SBs (Static cluster boundary sensors).This message is in the form of bitmap and if detected object is identified within cluster n then nth bit is set to 1. Communication takes place between SB and its one hop neighbors after the detection of object to inquire the neighboring nodes of their object detection statuses .When SB receives this information, the corresponding bit of the bitmap is set to 1 and report message is sent to CH so that all the static clusters are identified in which the target object has spread.

All the nodes in the static cluster transmit their information of detection to CH and dynamic cluster is then formed from the boundary nodes. In this case, the boundary nodes for the detected object are determined by cluster heads of the static cluster instead of dynamic clusters in order to save energy but still high communication cost is required in case of large number of nodes and also too much energy is consumed in grouping the boundary nodes into dynamic cluster.

2.2.7 COBOM

COBOM (Continuous Boundary Monitoring) [50], an energy efficient algorithm for boundary detection and tracking. Initially the nodes communicate with each and each node sends its Boundary Node Array (BN-Array), start node and its unique ID to the sink. The start node is chosen randomly initially. The BN-Array contains detection readings of the neighboring nodes. Initially the BN-Array readings are all zeros means that phenomenon is not detected by any node. The sensor broadcasts its own ID and reading if its current reading is different compared to its previous reading i.e detected the phenomenon or vice versa. The receiving node stores the reading in its BN-Array and the sensor becomes the Boundary node (BN) if its BN-array contains at least one different reading. For example initially BN-Array was all zeros and after detection the BN-Array has reading of 1 also. As shown in the Figure 2-1 the BN-Array stores the readings in counter clockwise direction with the start node c.

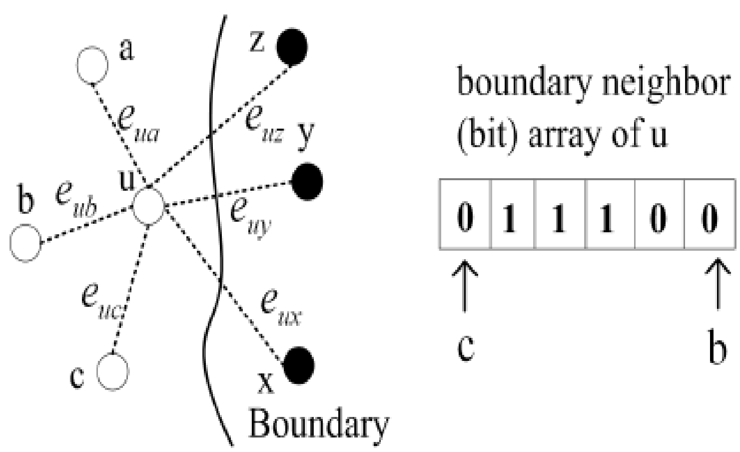


Figure 2-1: COBOM BN-Array [50]

Among these boundary nodes few nodes are selected to become Reporting nodes (RNs). The BN-Arrays of the nodes with more number of different detection readings i.e more number of 1s readings as shown in Figure 2-1, the more likely it will become a candidate for Reporting Node (RN). The reporting nodes are selected when a BN-Array of a particular node crosses a threshold set. As BN-Array reaches threshold, a back off random timer will start. At the end of this back off random timer, the node will become a Reporting Node (RN). The earlier the BN-Array of a particular node reaches the threshold, the early it will become the RN and suppresses the nodes within its range from reporting. COBOM claims to be energy efficient as only few reporting nodes are selected and the report message size is small as it only contains neighbors' detection readings as bits instead of keeping neighbors' IDs. The report message size contains ID of the RN, its reading and the node's BN-array. The drawback of COBOM is that it is not mentioned that how the base station will determine the location of RN's neighbors since the BN-Array only contains the detection readings not the IDs. Base station has ID of RN only which compromises the accuracy when base station tries to reconstruct the boundary later on. Another inefficiency in COBOM is that BNs and RNs will be formed on both sides of the Boundary. As shown in the Figure 2-1, if we examine the BN-Array of node y, it will be similar to that of u indicating that node y will also become the Boundary Node. This will result in yielding more number of BNs and RNs and thus increasing the communication cost.

2.2.8 DEMOCO

For detection and tracking of continuous object, an energy efficient algorithm DEMOCO (Detection and Monitoring for Continuous Objects) [51] is proposed. An object is detected by a sensor node by comparing its current detection status to its previous slot. CompareOnezero (COZ) Messages containing current detection status are sent to the neighboring nodes. The Boundary Nodes (BN) for the continuous objects are those nodes that receive the COZ messages and their object detection status is different from the status included in the messages. If the detection status is same as that of neighboring nodes, the receiving node ignores this message.

Different random back off timers are assigned to the BNs according to equation 2.1. The nodes with the short back off timers send the data to the sink and suppresses the other boundary nodes from sending the data to the sink. These nodes are called as the Representative Nodes (RN). Very few RNs report data to the sink. The report back message includes the RN's own ID and neighboring BN's ID that has the most powerful signal strength. The back off timers for the BNs are calculated by the following equation

$$\mathbf{D} = \begin{cases} \frac{W}{COZ_{total}} - U\frac{\left(\frac{W}{COZ_{total}+1} - \frac{W}{COZ_{total}}\right)}{2}, if\ \ COZ_{total} > 2 \\ \frac{W}{COZ_{total}} + U\frac{\left(\frac{W}{COZ_{total}-1} - \frac{W}{COZ_{total}}\right)}{2}, if\ \ COZ_{total} = 2 \\ \frac{W}{COZ_{total}} + U\left(\frac{W}{COZ_{total}-1} - \frac{W}{COZ_{total}}\right), if\ COZ_{total} = 1 \end{cases} \quad (2.1)$$

It is not clearly discussed that how the signal strength is compared to the received signals. As the signal strength is in the form of 1's and 0's, the estimation of signal strength is not possible by simply comparing 1 and 0.

2.2.9 GAS

Grid Based protocol known as GAS (Grid Based Asynchronous) [52], a selective wake up protocol is used to track the continuous objects efficiently and accurately at the right time. WSN is divided geographically into cells and each cell is a square shaped having specific areas around the boundary known as Guard Band as shown in Figure 2-2. As soon as the sensor nodes in the guard band detect, they activate guard bands in the adjacent touching cell before the target actually approaches. Each cell in the grid has its own coordinates and the sensor nodes in the cell can find its coordinates via reference point. Now each cell also has a cell head i.e CH (a,b) coordinates of the cell are a and b. The cell head in a cell is elected by the backward timer via formula

$$\frac{2R}{a{*}E_{residual}} \qquad (2.2)$$

Where $E_{residual}$ is the residual energy of the sensor node and R is the range of the sensor node's location from the center point of the cell. After the timer expires, a sensor node is elected as the cell head and it performs restricted flooding to notify others about its role of cell head.

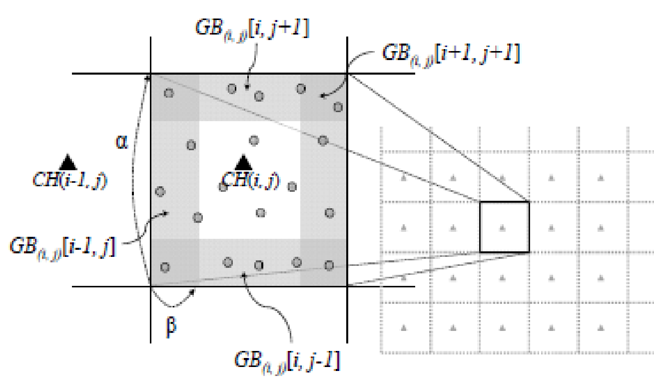


Figure 2-2: Guard Bands [52]

When sensor node $GB_{(i,j)}$[i-1,j] detects a boundary of the object, it sends message to its cell head CH(i,j) which in turn sends message to the adjacent cell's cell head CH(i-1,j) to be ready for monitoring. There are six types of signaling messages. Wakeup Request (WR), Object Detect (OBD) , Sleep Request (SR), Cell Sleep Request (CSR), Object Not Detected (OND), Cell wakeup Request (CWR) and. The sensors that usually detect boundary objects and do not detect send OBD messages and OND messages to their cell heads respectively. WR and SR is sent by the cell heads to the neighboring cell heads. CWR and CSR is sent by the cell head to the sensors after it receives WR and SR from adjacent cell heads respectively. The wake up operation of the protocol is performed when the sensor in the GB detects a boundary object. Cell head CH(i, j) receives OBD message and this cell heads sends WR message to its neighboring CH(i-1,j) which in turn send CWR to its sensors to activate only the sensors in its own GB. Later on when the cell in this GB send OBD message to the corresponding cell head the other sensors outside the GB are also activated. Sleep operation of the protocol is performed when the there is no boundary in the cell. The cell head issues CSR to its sensors except the sensors in GB in order to not to miss the tracking of sudden movement of the boundary. Later on when the cell head receives SR from the neighboring cell head then CSR is issued to sleep the sensors in the GB also.

For random deployment of the sensor nodes there might be void area in the wireless sensor network. Due to these void areas, the activation control message cannot reach the predicted areas and hence the task of detection and tracking of continuous object fails.

2.2.10 A Chaining Selective Wakeup Protocol

A chaining selective wake up protocol is proposed in [53]. Information about void area is collected at the network configuration time and if the next boundary area is in the void area then a chained area that is surrounding the void area is activated with the activation control message. Grid structure is used to predict the next location of the continuous object. WSN is divided geographically into cells and each cell is a square shaped having specific areas around the boundary known as Guard Band. As soon as the sensor nodes in the guard band detect, they activate guard bands in the adjacent touching cell before the target actually approaches. Each cell in the grid has its own coordinates and the sensor nodes in the cell can find its coordinates via reference point. Now each cell also has a cell head i.e. CH (a, b) whereas coordinates of the cell are a and b. The cell head in a cell is elected by the backward timer via equation 2.2.

A sensor node is elected as the cell head after expiration of time and it performs restricted flooding to notify others about its role of cell head. The void areas can be of five types such as Central, Edged, Disjoint, Multiple disjoined and fully void area. The detection of void area is a two-step process first communication with in a cell and secondly communication between the cells. The cell head CH (i,j) sends Void Area Probe (VAP) to all the sensor nodes with in the cell in all four directions. The sensor nodes reply with Void area probe- reply message and the cell head makes the Guard Band GB table according to these reply messages. When there no reply from the sensors indicating the void area, the cell head increases the size of GB in that direction. After the construction of GB table, the CH sends the Cluster Head Alive (CHA) messages to the neighboring cell heads and this alive information of neighboring cell heads is stored in the

Neighboring cell table and the cell heads refer to this table when sending and receiving signaling messages from the neighboring cell heads.

When sensor node $GB_{(i,j)}$[i-1,j] detects a boundary of the object, it sends message to its cell head CH(i,j) which in turn sends message to the adjacent cell's cell head CH(i-1,j) to be ready for monitoring. There are 8 types of signaling messages which consist of detection messages and request messages. Detection messages are Object Detection Message (OBD) and Object Not Detected (OND) message. The request messages include Sleep request (SR), Cell sleep request(CSR), Wakeup Request (WR), Cell Wakeup Request (CWR), Left hand guard band wakeup request(LGWR) and Right hand Guard band wakeup request (RGWR). The sensors that usually detect boundary objects and do not detect send OBD messages and OND messages to their cell heads respectively. WR and SR is sent by the cell heads to the neighboring cell heads. CWR and CSR is sent by the cell head to the sensors after it receives WR and SR from adjacent cell heads respectively. If the next cell is a void area the LGWR and RGWR will be sent by the cell head.

CHAPTER 3

CONTINUOUS OBJECTS DETECTION AND TRACKING (CODAT)

There are number ways to detect and track continuous objects. The simplest and straightforward approach for the detection of boundary is that every node sends its detection information to the sink node which further forwards information to the user via internet or any infrastructure network. The main disadvantage of this approach is that most of the energy is dissipated at the breakpoint and network lifetime of the sensor network is considerably reduced. In DCSC [48], small number of nodes are selected at the boundary of the continuous object grouped into clusters. This approach considerably reduces the cost of communication and will require much less energy as only nodes at the boundaries are responsible for forwarding data to the sink nodes.

In COBOM [50], the Boundary Nodes (BN) selected are much more compared to the DCSC [48] but only few Reporting Nodes (RN) are responsible for forwarding data to the sink. Our proposed approach is similar to the approach used by DEMOCO [51] in which Boundary Nodes (BN) as well as Reporting nodes (RN) are further reduced compared to the previous algorithms but the main advantage of our proposed algorithm lies in improvement in accuracy and efficiency of the detection and tracking of continuous objects. Our proposed approach is different in respect to the average report data size forwarded by the RNs to the sink node.

For analysis, we compared our proposed solution with COBOM [50] and DEMOCO [51] on the basis of efficiency, accuracy and uniform random deployment. The following Table 3-1 shows some of the features along with the drawbacks found in these two algorithms.

Table 3-1: Comparative Analysis of COBOM and DEMOCO

Features	COBOM	DEMOCO
Detection	Based on BN array	Based on diff in detection readings
Boundary Nodes	Both sides	One side
Number of BNs and RNs	N	N/2
Drawback	- No neighboring node ID - BN-Array readings only	Nearest ID based on signal strength

We are assuming some initial conditions for our algorithm to work on a wireless sensor network.

- Static nodes are deployed in a large number along with sink nodes.
- All sensor nodes are uniformly distributed.
- All nodes have same functionality and capability.
- All sensor nodes have same sensing range and communication range r.
- Locations of all sensor nodes are known by the sink nodes through unique IDs via GPS unicast messaging.

- Through GPS or any other triangulation or localization technique, all sensor nodes know their own location and position.
- The possibility of any data loss is not considered.
- The damage or destruction of nodes is not taken into consideration for our algorithm.
- Every node has a unique ID.

3.1 Definitions

Phenomenon Detection Status (PDS) The sensor node's detection status changes such as from undetected to detected or from detected to undetected. Detected status is represented by 1 and the status of undetected is represented by 0.

Phenomenon changed message (PCM) The node having different PDS compared to its previous timeslot, will broadcast a message to its neighboring nodes. This message is called as Phenomenon changed message. This message contains node's ID, detection status and Phenomenon tags.

Boundary Nodes (BN)

When a node receives PCM, it compares its own PDS compared to the PDS received in broadcasted PCM. If a node has at least one different detection status compared to the received message then that node becomes the boundary node.

Reporting Nodes (RN)

The Reporting Nodes (RN) send information to the sink on behalf of the surrounding neighboring nodes in order to conserve the network energy. In our algorithm a Reporting node sends its own ID and a Boundary node array.

3.2 The Proposed Algorithm

Here we propose an energy efficient Algorithm CODAT (Continuous Object Detection and Tracking)

All the sensor nodes periodically activate to detect the phenomenon such as wild forest fire, gas leakage, oil spills etc. depending upon the specific application. If they do not detect any thing, the sensors go back into idle state.

If a sensor node detects the phenomenon and finds that the current Phenomenon Detection Status (PDS) is different from the PDS it observed in the previous timeslot, then it broadcasts a message known as Phenomenon Changed Message (PCM). This PCM contains a node's unique ID and PDS.

A sensor node x when receives a PCM from a neighboring node within its communication range, it compares its own PDS with the received PDS from PCM. If both the PDSs are same then node ignores this broadcast message. If the node finds at least one received PDS different from its own PDS, the node x then becomes a Boundary Node (BN).This boundary node counts the number of received PCMs for setting different waiting time which will be used later for making Reporting Nodes (RN). This Boundary node also maintains a Boundary Node-Array (BN-Array) which contains neighboring nodes' detection readings and a couple of IDs along with Phenomenon Tags. This BN-Array will be explained in later section.

In our algorithm when the object is expanding then only the nodes in the Outer region of the Phenomenon become BNs and similarly when the object is shrinking, only the nodes in the inner region become BNs as shown in Figure 3-1 and Figure 3-2. This approach is different from the algorithm proposed in COBOM in which the nodes in both the outer and inner region of the phenomenon become BNs regardless of the expansion or shrinkage of the object because COBOM selects the BNs based on the difference in detection readings within a node's BN-Array which eventually results in more number of BNs compared to our algorithm CODAT.

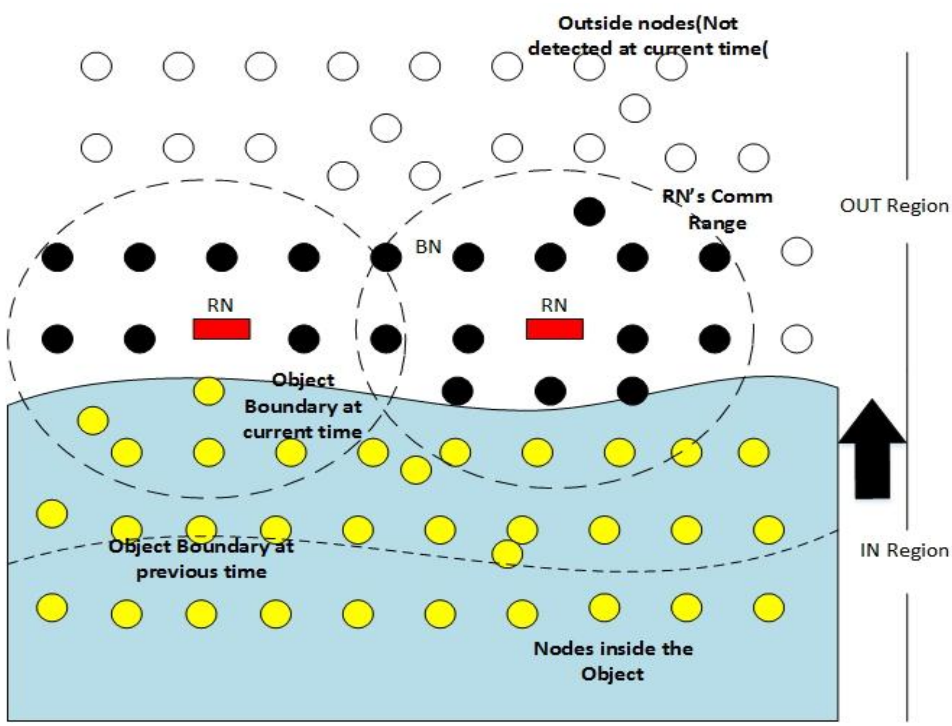


Figure 3-1: Expansion of Phenomenon

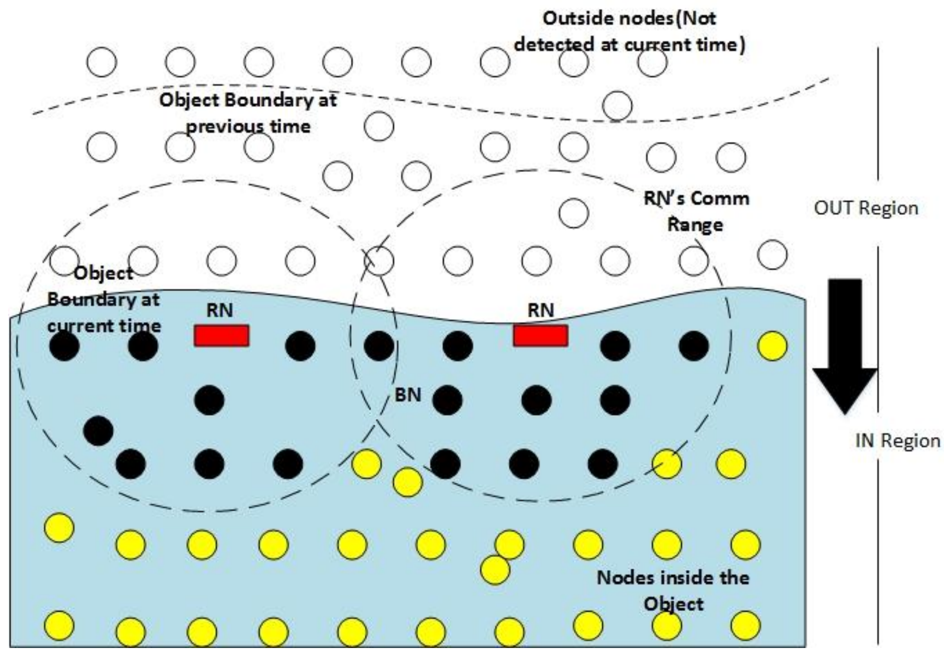


Figure 3-2: Shrinkage of Phenomenon

3.3 Differentiating Between Outer and Inner Boundary of the Phenomenon

Although most of the algorithms can differentiate between the holes and overall phenomena but we propose an additional information as an added improvement forwarded to the user along with the reported data with more accuracy and with less number of reporting nodes compared to COBOM. For differentiating between The Outer Boundary (Boundary of the phenomenon such as expansion or shrinkage of the phenomenon) and the Inner Boundary (Boundary of the hole such as expansion or shrinkage of hole inside the phenomenon) by the Base station we used an approach similar to [56] explained below.

Every node detects a change in its status after definite interval. If the node does not detect change in its PDS then it becomes idle. For outer boundary there are two cases. When the node detects the phenomenon in a current time and previously it did not detect anything then it broadcasts a Phenomenon Changed Message (PCM) which identifies change in the detection status to its neighbors that are in range. This node also receives PCMs from the neighboring nodes as well. If at least one of the received messages has the detection status different from the node itself then the node becomes the boundary node otherwise it will ignore the received PCM. Based on the number of PCMs that this node receives, back off timer will be set for it to make it a reporting node. A reporting node sends the data to the sink. At the end of this period boundary nodes will tag themselves as Phenomenon Expansion (PE). The reporting nodes will send the boundary information to the sink along with the current tags of the nodes. Similarly when the nodes did not detect phenomenon in the current time but detected the phenomenon in the previous time slot and has tag of PE (Phenomenon Expansion) then it will broadcast PCM and phenomenon shrinkage will be represented by this change. The node will become boundary node if it receives any PCM from neighboring node having different detection status compared to itself. On basis of number of received PCMs back off timers will be set and reporting nodes are selected. At the end of this process all nodes tag themselves as Phenomenon Shrinkage (PS).

For Inner boundary detection there are also two cases. One is when node detected in the previous time slot but does not detect in the current time slot and has no tag. This change represents the expansion of Hole shown in Figure 3-3. Other is when node did not detect in the previous time slot but detects in the current time slot and has tag of Hole Expansion (HE). This change represents the Shrinkage of Hole shown in Figure 3-3. In both cases, the node that detects the change, it broadcasts PCMs and receives PCMs from its neighbors. This node also becomes a boundary node provided that the status in at least one of the received PCMs is different from its own detection status. Reporting nodes are selected on basis of back off timers. In case of Expansion of Hole, nodes tag themselves as Hole Expansion (HE) and in the case of Shrinkage of Hole, nodes tag themselves as Hole Shrinkage (HS). These tags are valid for single time period only. These tags actually create difference at the base station level between the Phenomenon Shrinkage and Hole Expansion and similarly between Phenomenon Expansion and Hole Shrinkage.

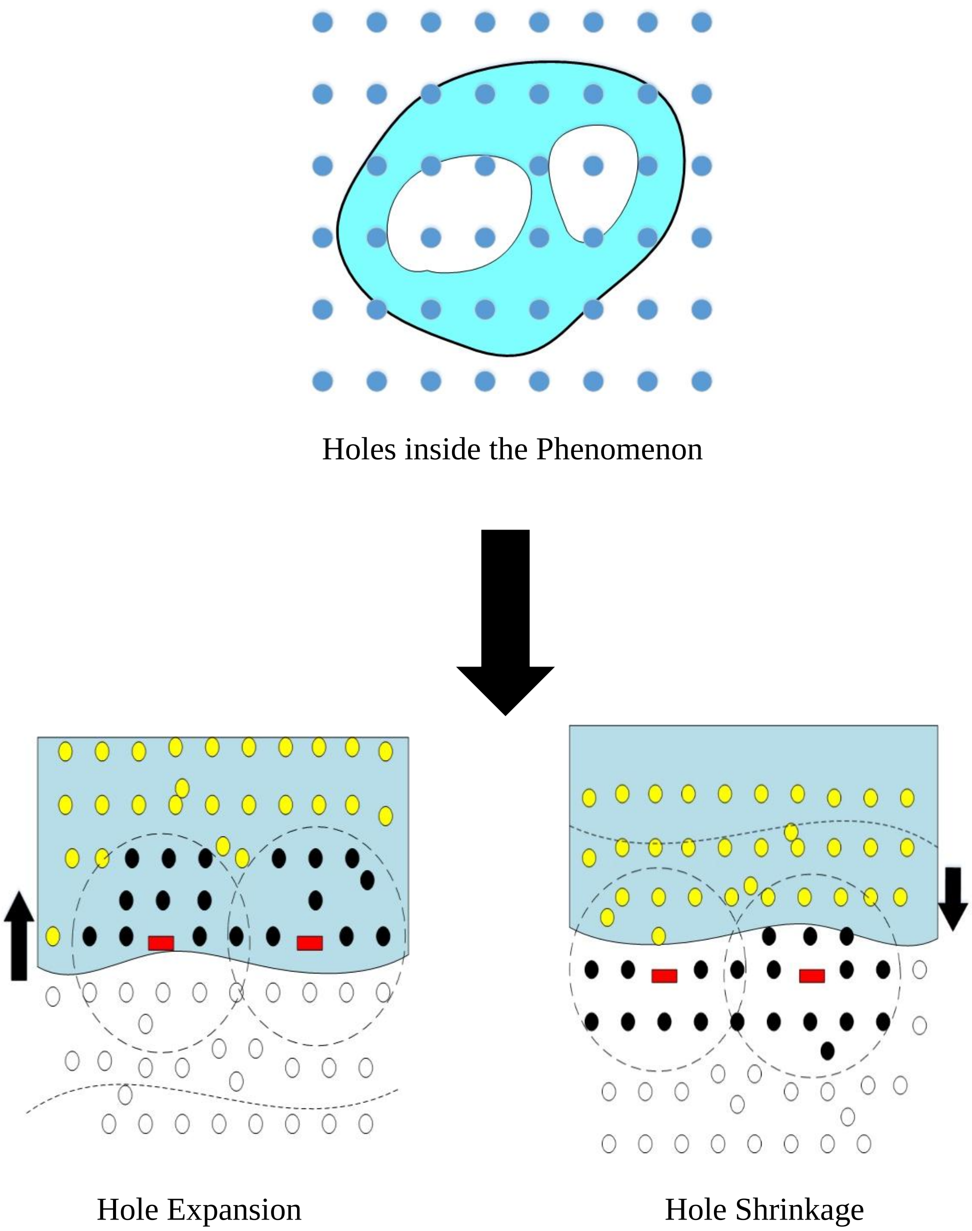


Figure 3-3: Hole Expansion and Shrinkage

In case of undetected to detect situation there are two possibilities, either the phenomenon is expanding or the Hole is shrinking. As it is assumed that Hole will not shrink unless it expands so if the node detects phenomenon currently and did not detect in previous time slot and has the tag HE(Hole expansion) then it means that it was previously involved in finding the boundary nodes for Hole expansion and currently Hole is shrinking. In other case if node detects currently and did not detect previously and has no tag then this indicates that phenomenon is expanding.

In case of detect to undetected, there are also two situations either the phenomenon is shrinking or the hole is expanding and it is assumed that phenomenon will not shrink unless it has expanded. So whenever the node detects in previous time slot and did not detect in current time slot and has the tag of Phenomenon expansion (PE) then it indicates that this node was previously involved in the finding boundary nodes for expansion of phenomenon and currently phenomenon is shrinking. In the same situation of detected to undetected if the node has no tag then it indicates that currently Hole is expanding. These tags will expire after a single time period via timers. New tags will be formed in the next time period.

3.4 Reporting Nodes

The simple approach would be that all the boundary nodes send their data to the sink but this approach is cost inefficient. Our main objective is to increase the network life time as well as improve the accuracy in detection and tracking of phenomenon. So we choose few Reporting Nodes (RN) among the existing BNs that will report data to the sink on behalf of the neighboring nodes while maintaining the reasonable accuracy. The RNs are chosen based on the number of PCMs. The BN that receives more number of PCMs will set a random waiting time. The more a BN receives PCMs, the shorter will be its random waiting time and it will be more likely to become a RN. This approach will ensure the high probability of the boundary nodes which are near the actual boundary of the phenomenon to become RNs. If two or more BNs have same random waiting time then the BN with the highest energy level is selected as the RN. The reelection of RN is performed after each time period. The waiting time as mentioned in DEMOCO [51] can be defined in Equation 3.1.

$$\mathbf{D} = \left\{ \begin{array}{c} \frac{W}{PCM_{total}} - U\,\frac{\left(\frac{W}{PCM_{total}+1} - \frac{W}{PCM_{total}}\right)}{2}, if\ \ PCM_{total} > 2 \\ \frac{W}{PCM_{total}} + U\,\frac{\left(\frac{W}{PCM_{total}-1} - \frac{W}{PCM_{total}}\right)}{2}, if\ \ PCM_{total} = 2 \\ \frac{W}{PCM_{total}} + U\left(\frac{W}{PCM_{total}-1} - \frac{W}{PCM_{total}}\right), if\ PCM_{total} = 1 \end{array} \right\} \quad (3.1)$$

Where back off random timer is denoted by D, maximum waiting time (time period in our case) by W. The total number of PCM messages that are received is denoted by PCM_{total} and uniform distribution in between the first and second term of U, i.e., [W/PCM_{total} – 1, W/PCM_{total}]. As shown in Equation 3.1 if PCM_{total} >2 then the back off timer will be shortest. If the PCM_{total} is 2 or 1 the back off timer becomes longer according to the equation. This equation of back off time ensures that back off timers for boundary nodes that are in communication range are different from one another.

As mentioned earlier, each node maintains a Boundary Node Array (BN-Array). This BN-Array contains detection status readings of the neighboring nodes that have been received through PCMs. BN-Array also contains the unique IDs of the first two nodes received through PCM.

As shown in Figure 3-4, a BN-Array has Phenomenon Detection Status's (PDS) along with unique IDs of the two nodes received through PCMs and Phenomenon Tags (PT). This data structure for BN-Array enables a sink to recognize neighboring nodes via first two unique IDs received through PCM and the rest of the nodes are located through a series of the detection readings in BN-Array near to the first two IDs. This is possible because the sink has a map to all the nodes in the network and comparatively more processing power as well as memory resources. Our approach is different from DEMOCO in which Reporting node sends its own ID and the ID of the nearest neighbor based on signal strength. We argue that estimating the nearest node based on signal strength is not very accurate approach since the signal strength is estimated in the form of 1's and 0's.

ID	ID	0	1	1	1	PT	PT	PT	PT

Figure 3-4: BN-Array of CODAT

The BNs with shortest waiting time will send broadcast message to its surrounding neighbors indicating that it will become RN and the neighboring nodes don't need to send data to the sink node. This RN will send report data to the sink on behalf of all the neighboring nodes which are in its communication range. In report data this RN will send its own unique ID, a BN-Array and the phenomenon tags as explained earlier. We are considering that each unique ID is approx.2 bytes. We will discuss in later section that how average report size of our proposed algorithm does not exceed the previous algorithms considering improvement in accuracy and efficiency of our proposed algorithm. The Figure 3-5 shows the Flow chart for CODAT algorithm.

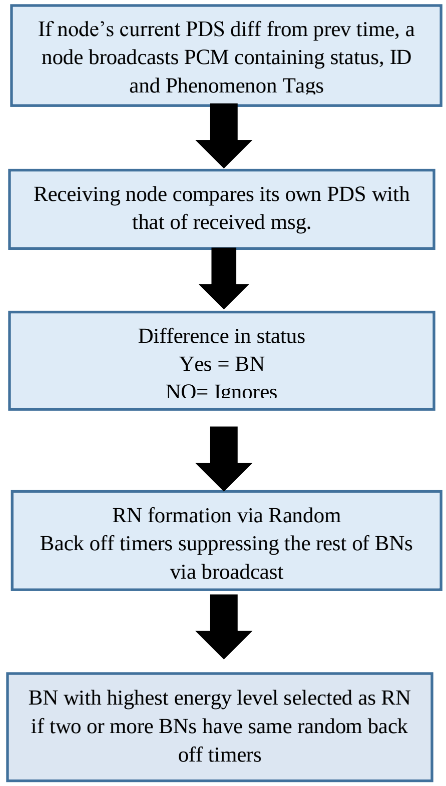


Figure 3-5: CODAT's Algorithm Flowchart

CHAPTER 4

SIMULATION RESULTS

In this section we evaluate and compare our proposed approach CODAT against both DEMOCO and COBOM. A simulator was developed in Java to test the performance of these algorithms. We are evaluating the algorithms for continuous moving phenomena. We are also neither considering possible contention nor concerned with the routing approach used by the nodes to send data to the sink. Each simulation is run 300 times and each node's unique ID is assumed as 2 bytes.

The number of BNs and RNs generated depends upon the density of sensor nodes. In our simulation model, the area is fixed while the number of nodes may vary. Wireless sensor nodes are deployed on the field of 500 x 500 m^2. The number of sensor nodes are set to 1500 and 5000 for sparse setting and dense setting respectively. The uniform distribution of 5000 sensor nodes is shown in Figure 4-2. In order to simulate real life scenario, an unsmoothed object is simulated. The object expands like a staircase structure from the bottom left corner shown in Figure 4-1. The object expands in each time period (each time period is 5sec) up to 25^{th} Time period with Low Diffusion Rate (Expanding 3 meters in each time period). The communication range for sensor nodes is set to 15 and 25m in order to test the performance of our proposed algorithm CODAT compared to COBOM and DEMOCO.

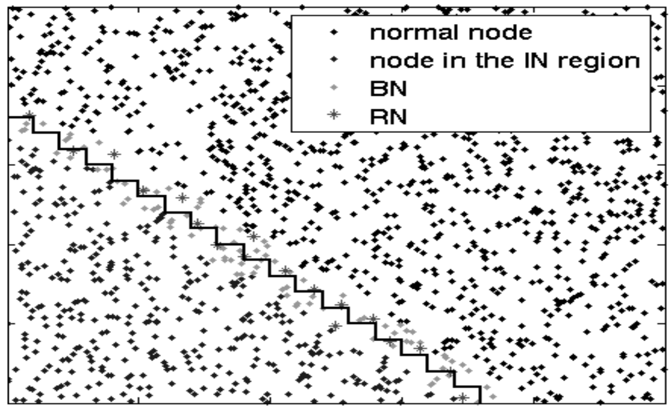


Figure 4-1: Expansion of Object in staircase pattern

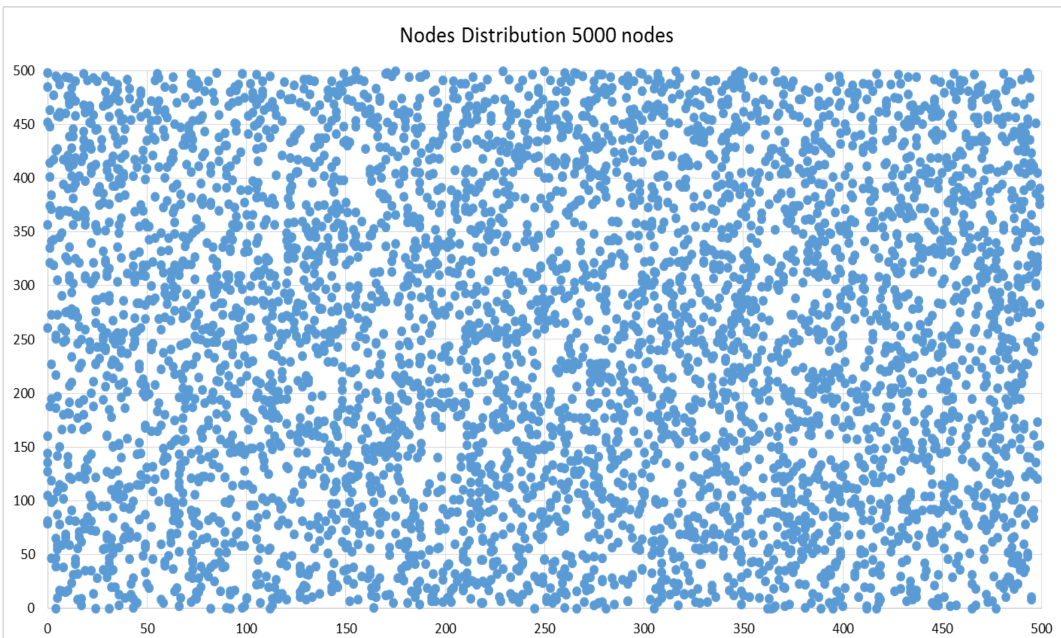


Figure 4-2: 5000 Sensor Nodes Distribution in Sensor Field

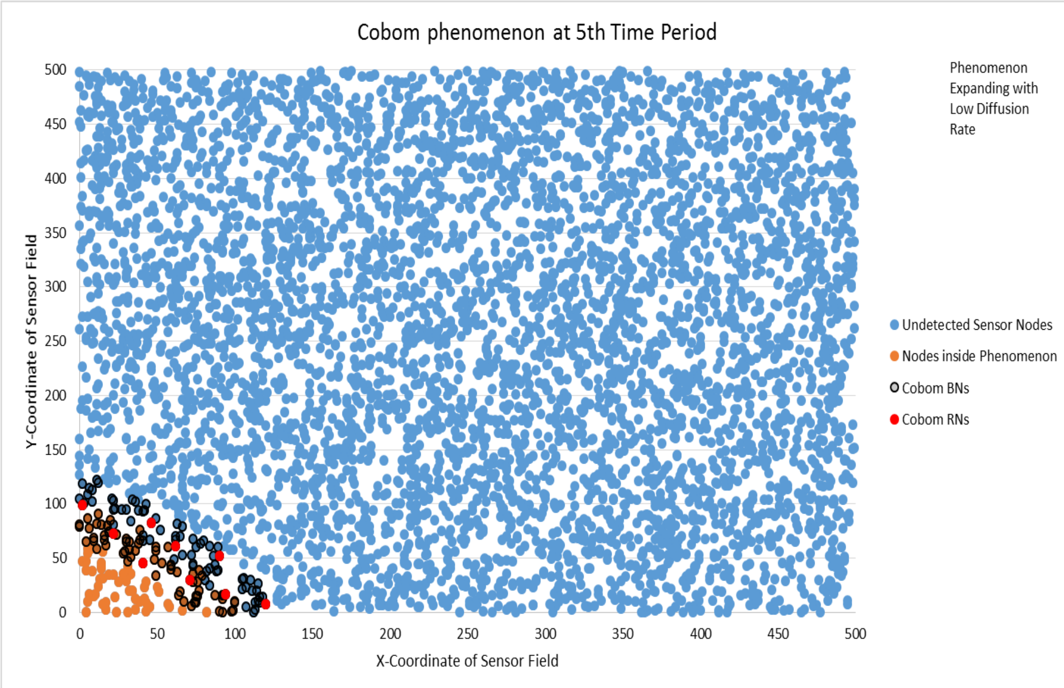


Figure 4-3: COBOM Phenomenon at 5th Time Period

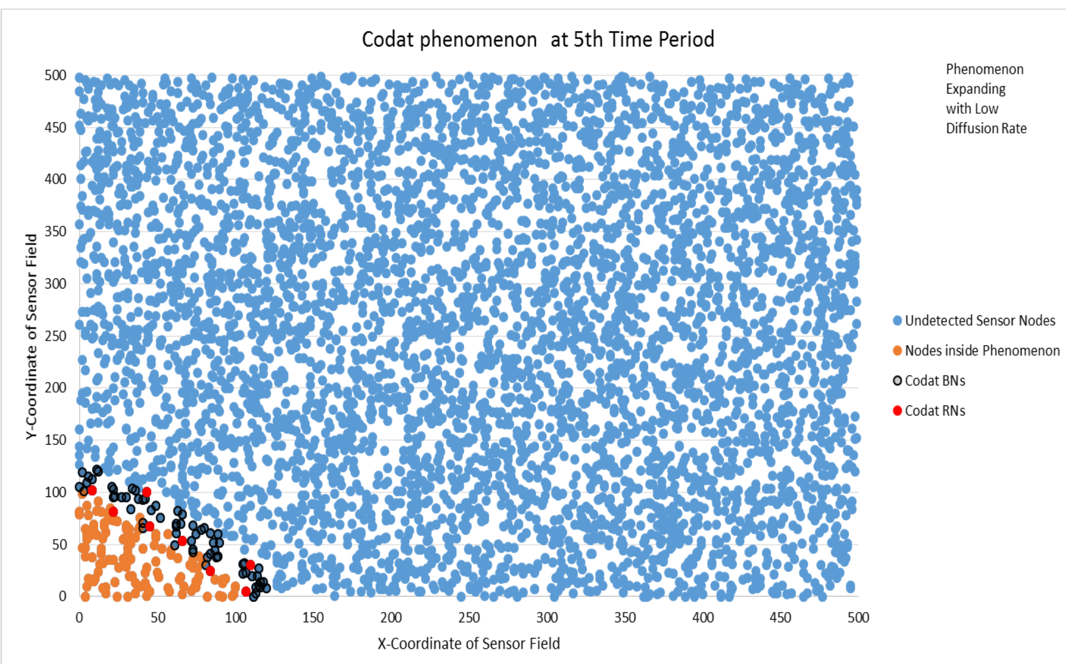


Figure 4-4: CODAT Phenomenon at 5th Time Period

In Figure 4-3 and Figure 4-4, the phenomenon expansion for total 5000 nodes is shown for both CODAT and COBOM algorithm at 5th Time period with 25m communication range for all sensor nodes. Similarly phenomenon expands at 10th Time period with 25m communication range for all sensor nodes as shown in Figure 4-5 and Figure 4-6. We can clearly see in these figures that in case of COBOM the Boundary Node strip (Blackish nodes) is more in width compared to the Boundary node strip of CODAT. This is because in case of COBOM Boundary Nodes (BNs) are formed on both sides of the boundary of the phenomenon, where as in case of CODAT Boundary nodes are formed on single side of the boundary of phenomenon resulting in overall less number of Boundary Nodes (BNs) and Reporting Nodes (RNs) compared to COBOM. Among these Blackish Nodes, Reporting nodes are shown in red color.

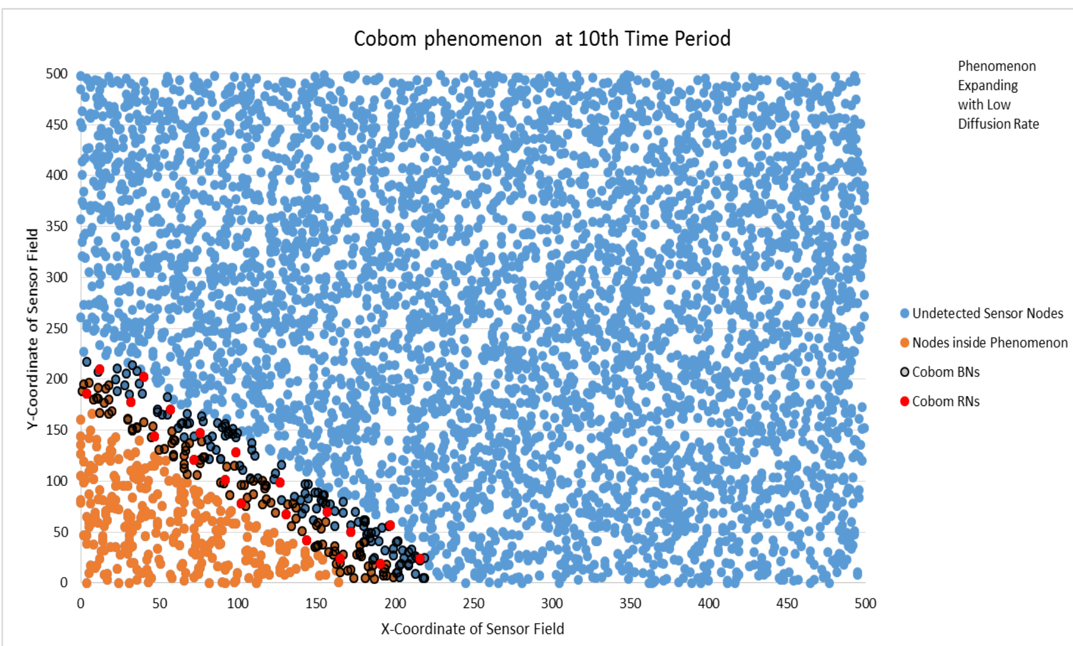


Figure 4-5: COBOM Phenomenon at 10th Time period

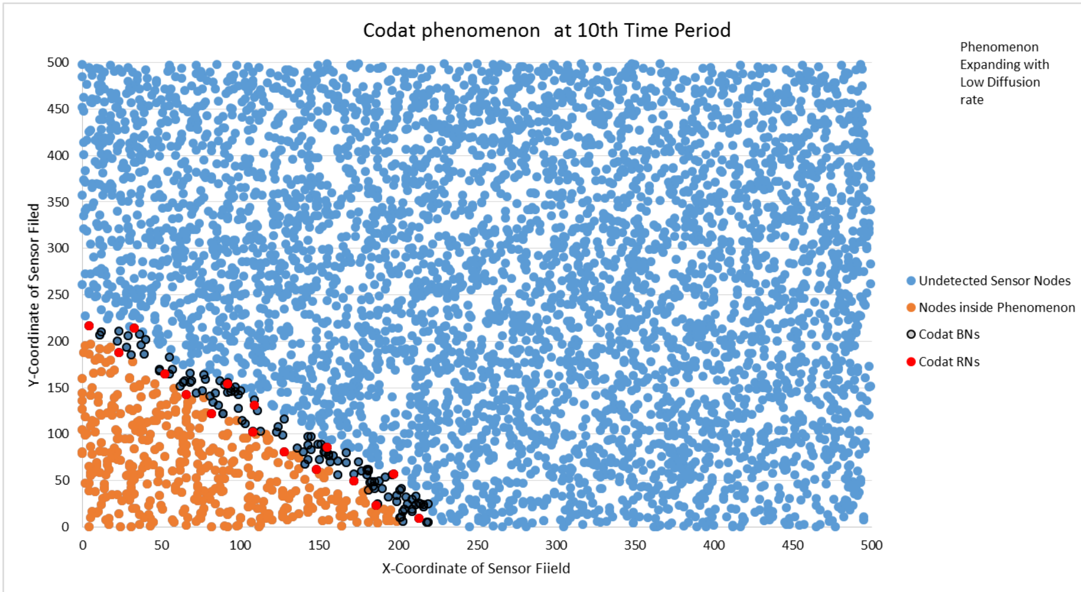


Figure 4-6: CODAT Phenomenon at 10th Time Period

In Figure 4-7 and Figure 4-8, the phenomenon at 20th Time period is shown which has further expanded.

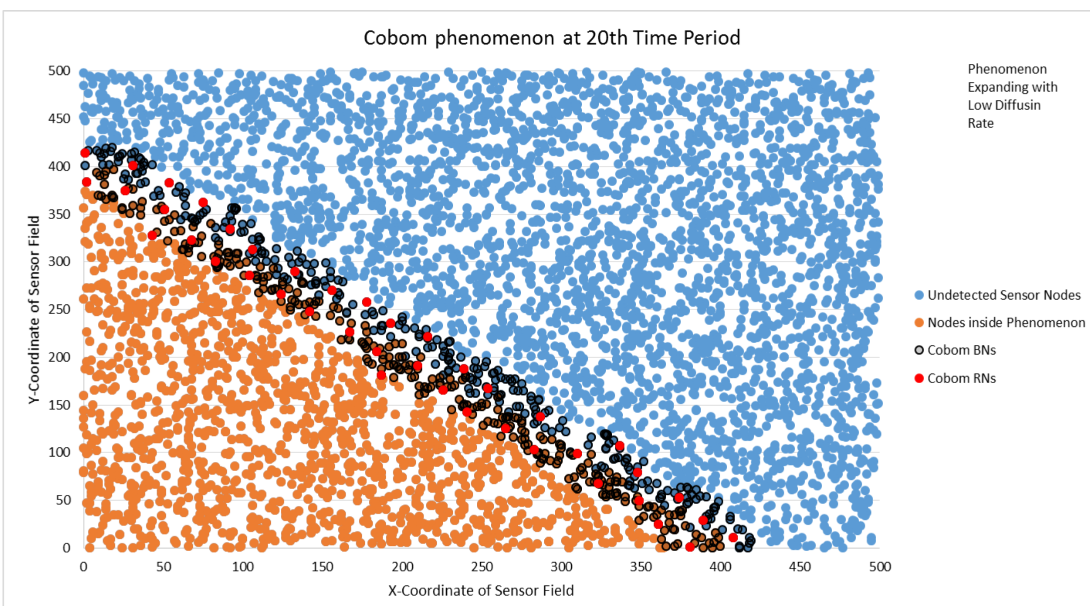


Figure 4-7: COBOM Phenomenon at 20th Time Period

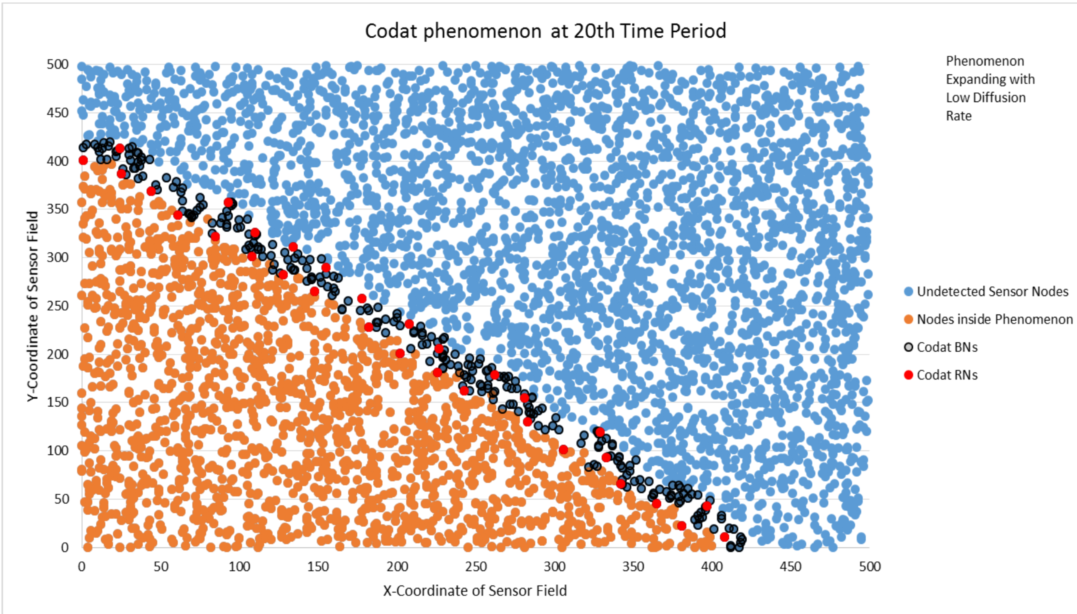


Figure 4-8: CODAT Phenomenon at 20th Time Period

In Figure 4-9 and Figure 4-10, the phenomenon has further expanded up to 25th Time period. At this point, maximum number of Boundary nodes and Reporting nodes are generated because the phenomenon has reached to the diagonal of square sensor filed.

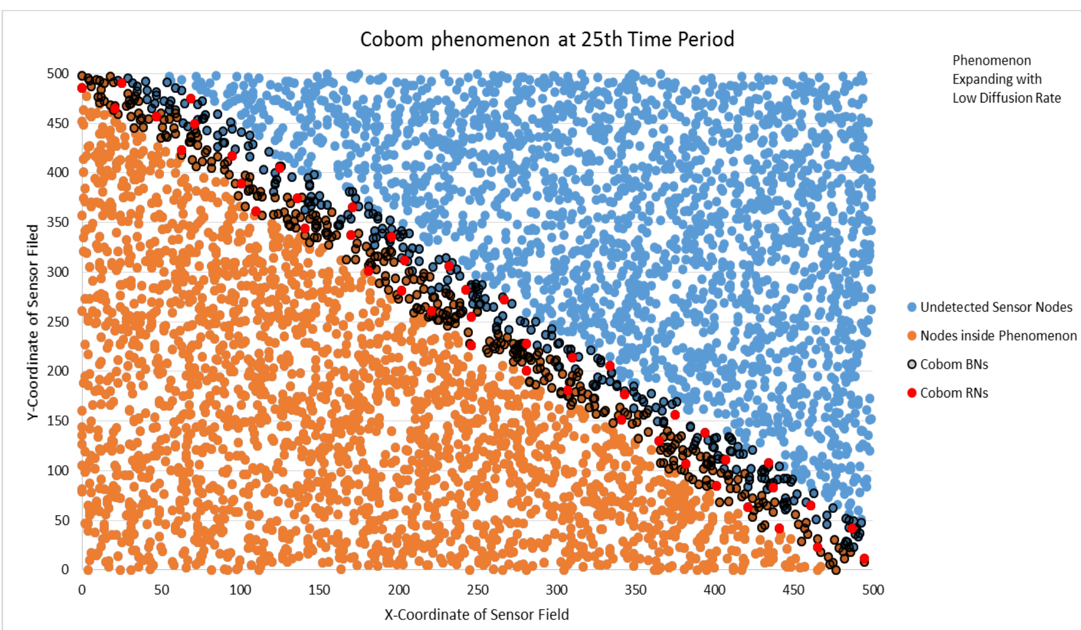


Figure 4-9: COBOM Phenomenon at 25th Time Period

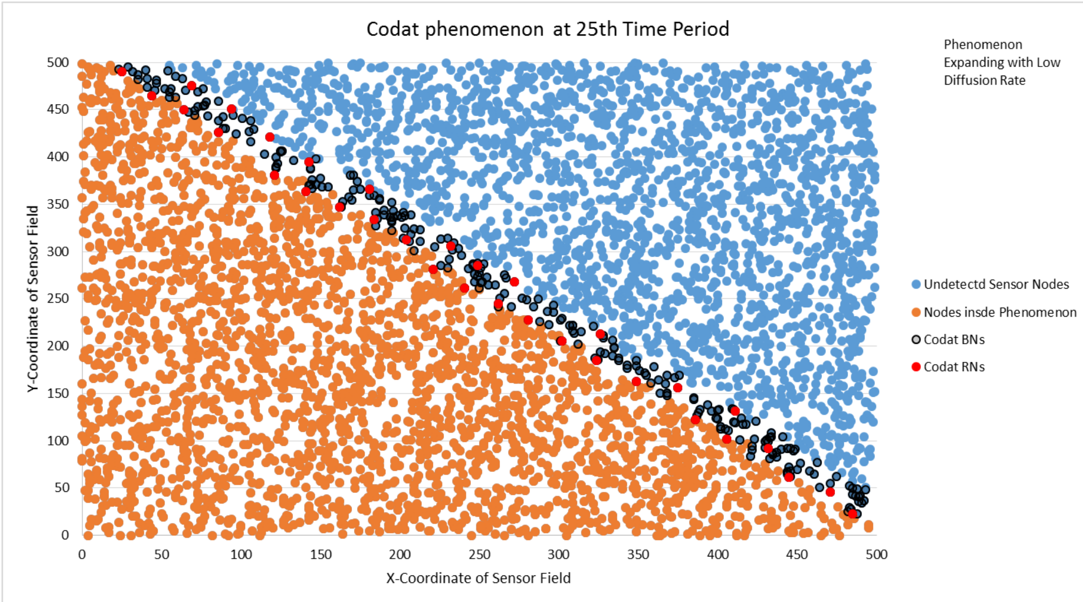


Figure 4-10: CODAT Phenomenon at 25th Time Period

4.1 Comparative Analysis Based on Number of BNs

The number of BNs generated as shown in the Figure 4-11, Figure 4-12, Figure 4-13 and Figure 4-14 with the varying density along with the variation in communication range of the sensor nodes. The number BNs increase with each time period due to expansion of phenomenon up to 25th period. The higher the number of BNs generated by the algorithm, the sensor network consumes more energy.

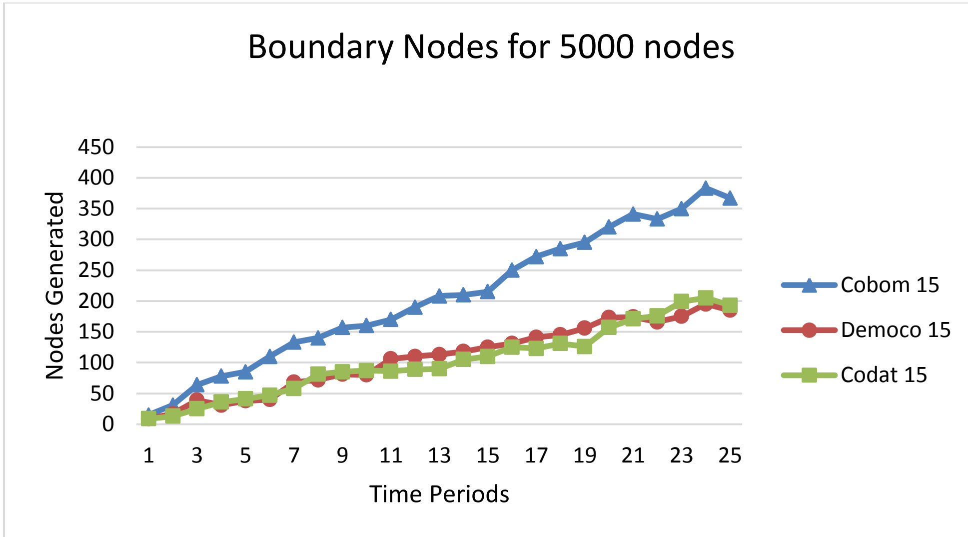


Figure 4-11: BNs for 5000 nodes with 15m Communication Range

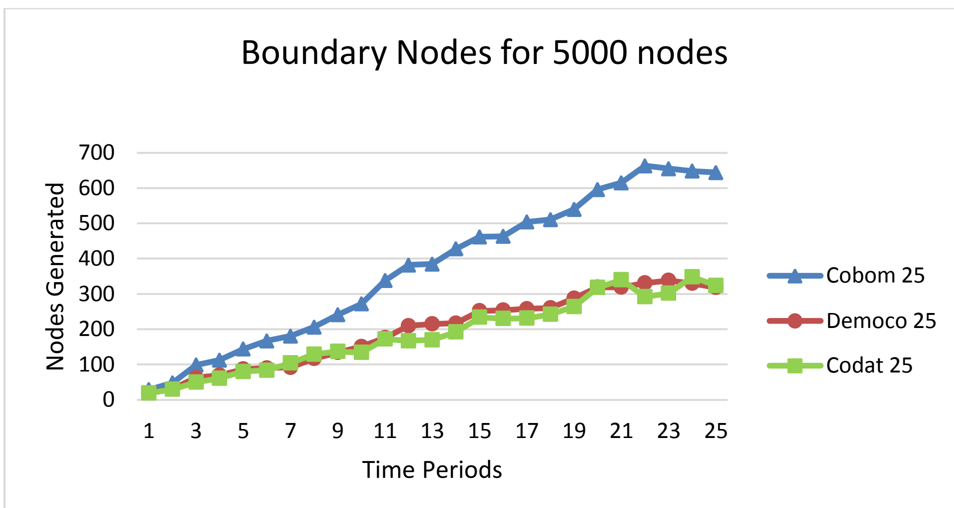


Figure 4-12: BNs for 5000 nodes with 25m Communication Range

The Figure 4-11 and Figure 4-12 show that the number of BNs generated by COBOM are far more compared to CODAT and DEMOCO for both 15m and 25m communication range respectively for dense WSNs.

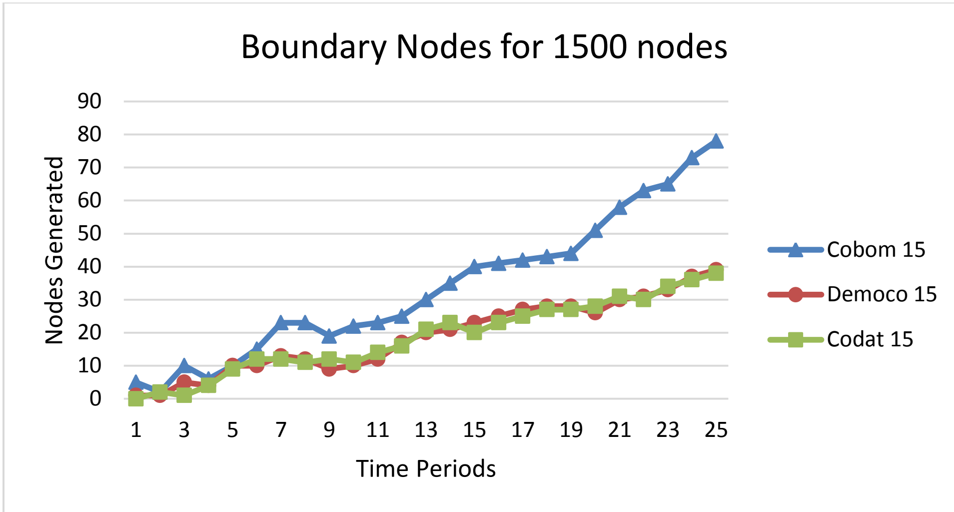


Figure 4-13: BNs for 1500 nodes with 15m Communication Range

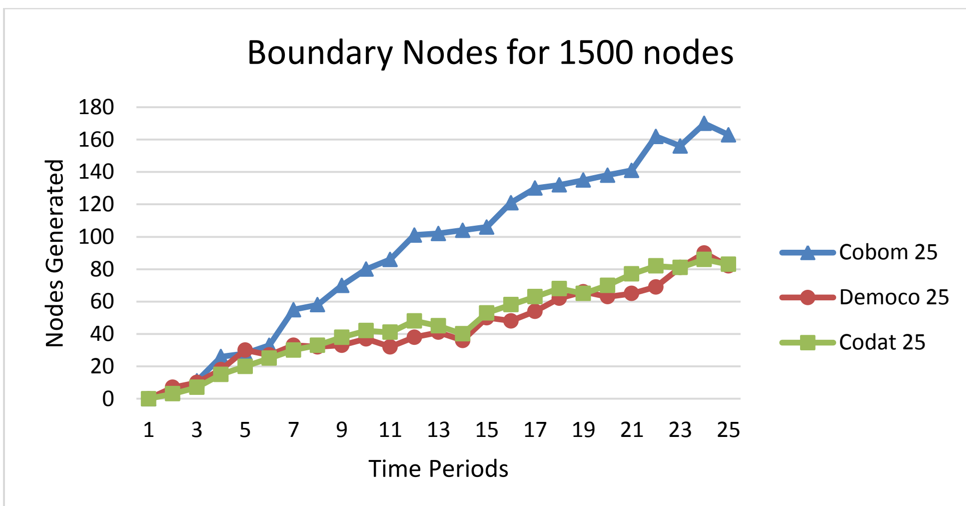


Figure 4-14: BNs for 1500 nodes with 25m Communication Range

Similarly Figure 4-13 and Figure 4-14 show that with the increase in expansion of phenomenon more number of BNs are generated and COBOM produces more number of BNs compared to CODAT and DEMOCO for sparse WSNs.

4.2 Comparative Analysis Based on Number of RNs

Similarly, the number of RNs generated can be shown in Figure 4-15, Figure 4-16, Figure 4-17 and Figure 4-18. Higher the number of RNs generated by the algorithm, more energy will be consumed by the sensor network. It is clearly shown in these figures that the number of BNs and RNs generated by the COBOM is more compared to DEMOCO and our proposed algorithm CODAT. This is because COBOM generates BNs and RNs on both sides of the boundary. Our proposed algorithm CODAT just like DEMOCO generates BNs and RNs only on the OUT region when objects expands and generates BNs and RNs only on the IN region when the object shrinks. The difference in BNs and RNs between COBOM and CODAT becomes apparent when the density of the sensor nodes is 5000 and communication range is 25m shown in Figure 4-12 and Figure 4-16. It has also been noticed that the number of RNs generated with sparse setting and communication range of 25m is more. Similarly the number of RNs generated with dense setting and 15m communication range is more. Apparently no strong relation has been found between communication range and RNs.

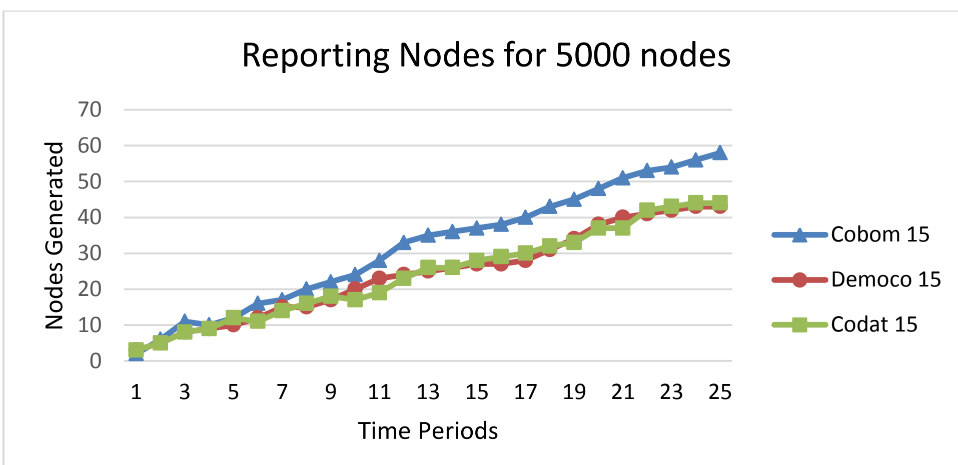


Figure 4-15: RNs for 5000 nodes with 15m Communication Range

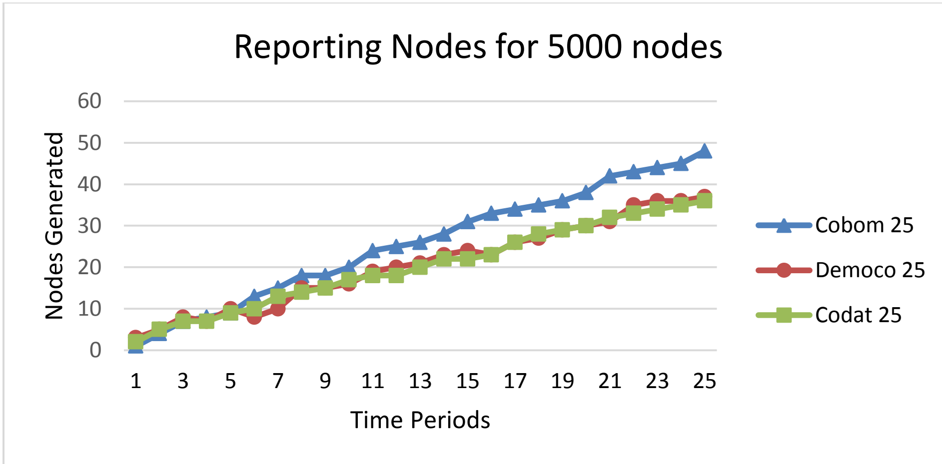


Figure 4-16: RNs for 5000 nodes with 25m Communication Range

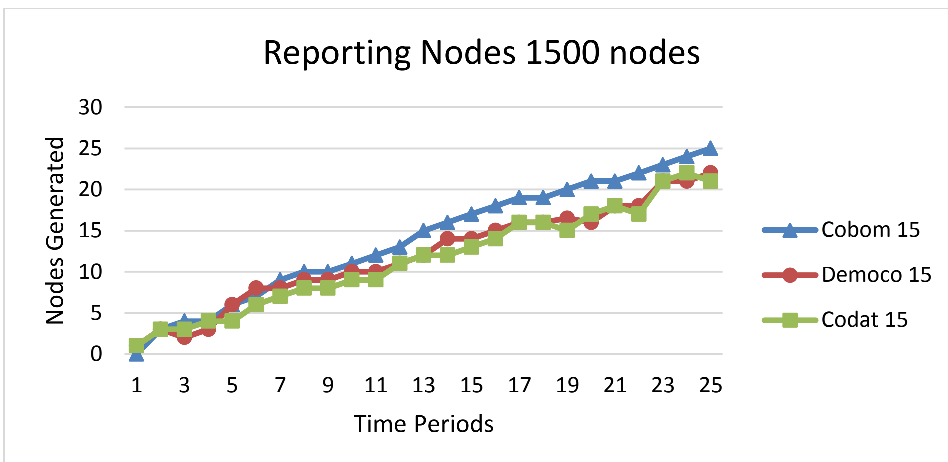


Figure 4-17: RNs for 1500 nodes with 15m Communication Range

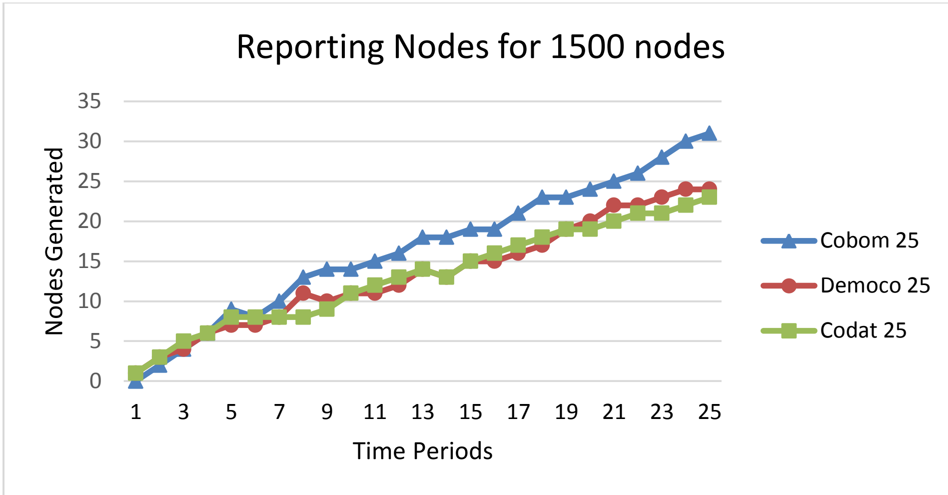


Figure 4-18: RNs for 1500 nodes with 25m Communication Range

4.3 Simulating the Holes along with overall Phenomenon

Although most of the algorithms can differentiate between the holes and overall phenomena. Our algorithm can provide an additional information as an added improvement forwarded to the user along with the reported data with more accuracy and with less number of reporting nodes compared to COBOM. In order to analyze that our proposed algorithm can differentiate at the sink level between the holes and the overall phenomena, we are considering a special case in which Staircase structure and Hole both are expanding. In case of Staircase structure expansion, nodes move from undetected state to detected state and in case of Hole Expansion, nodes move from detected state to undetected state. We initiated a single hole (Perfect circle in java) at 15th time period. The hole is centered at (76,106). The hole expands with each passing period with increase in size of 3 meters in each time period (Each time period of 5 seconds). Along with the hole, the staircase structure is also simulated which expands up to 25th time period. The center of the hole is chosen at (76,106) so that the Hole and the staircase phenomena do not merge into each other. Now both the hole and the staircase structure after reaching to 20th period, the total number of reporting nodes sent to the sink are calculated shown in Figure 4-19 and Figure 4-21. After that both the stair case structure and the hole further expand up to 25th period and again the number of reporting nodes sent to the sink are calculated shown in Figure 4-20 and Figure 4-21.

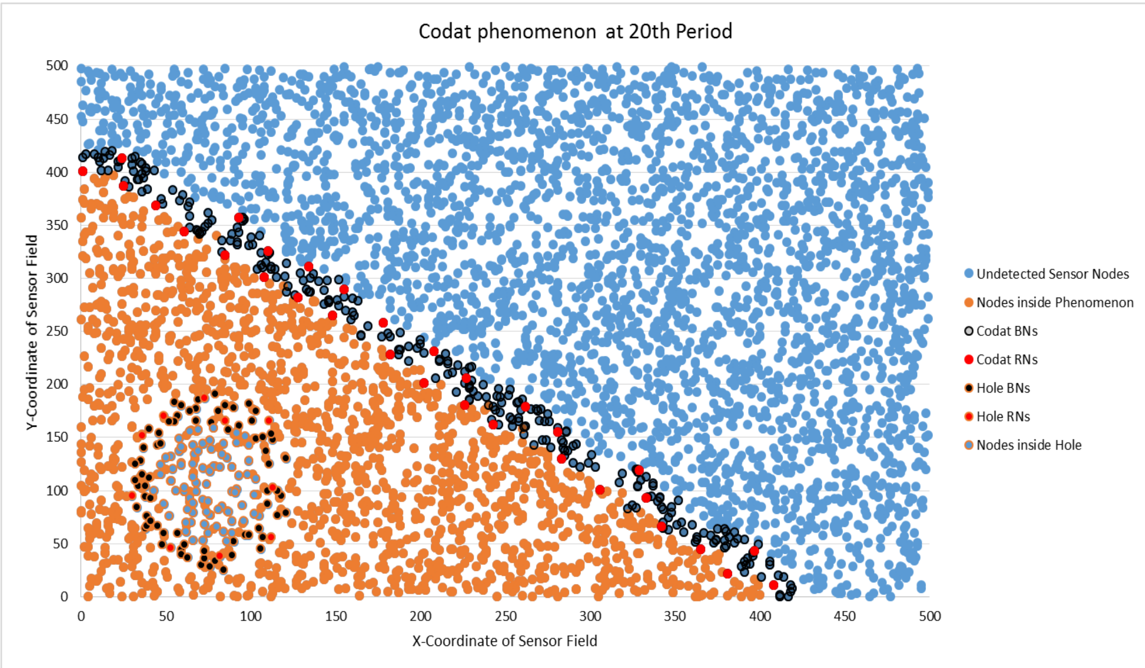


Figure 4-19: CODAT Phenomenon and Hole at 20th Time Period

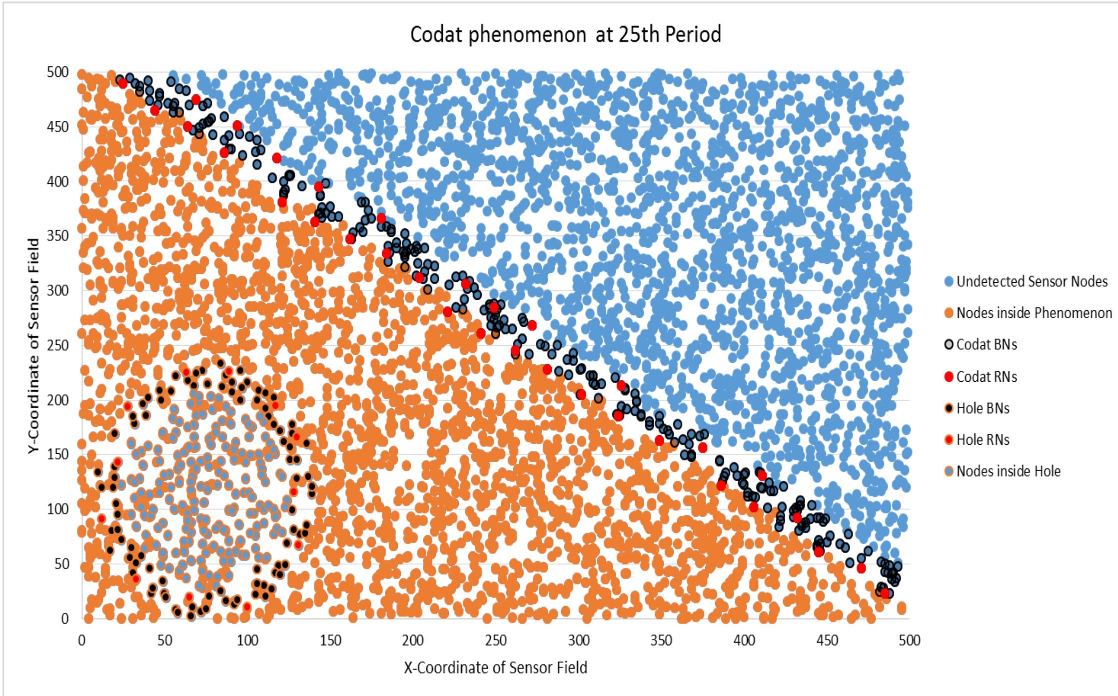


Figure 4-20: CODAT Phenomenon and Hole at 25th Time Period

As shown in the Figure 4-21, our algorithm CODAT can provide additional information as an added improvement forwarded to the user to differentiate between the reported data sent by the Outer Boundary (OB) or Overall Phenomenon and the reported data sent by the Inner Boundary (IB) or Hole due to Phenomena Tags. Similarly same results can also be deduced for Shrinkage of Staircase phenomenon and hole.

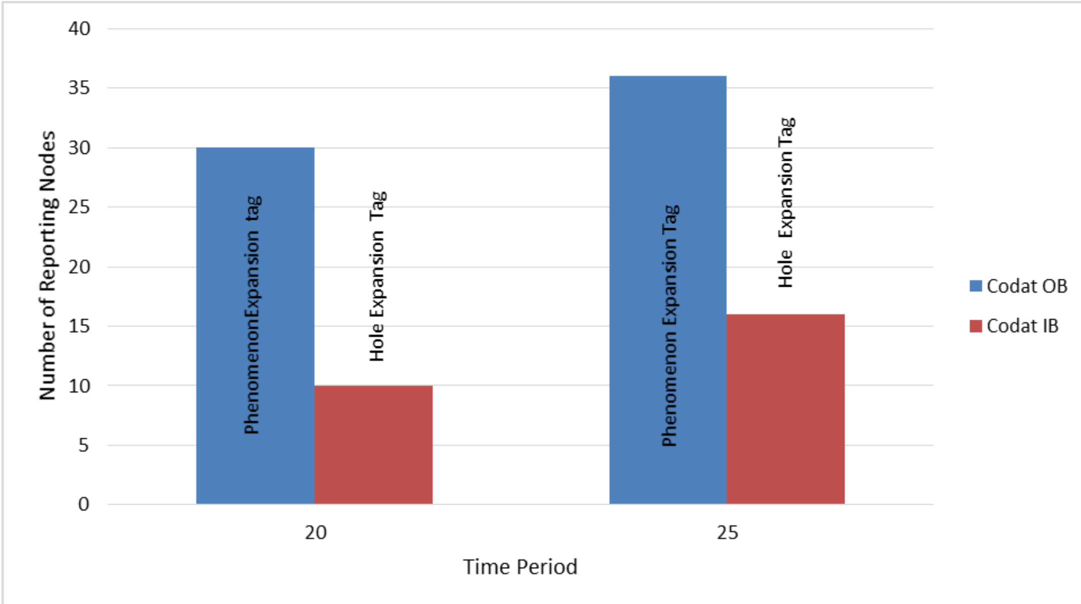


Figure 4-21: Differentiating Between the Inner Boundary and Outer Boundary

4.4 Comparative Analysis Based on Average Report Data size

Before calculating the size of average report data for each algorithm, there are some assumptions that are made for simplicity. The results are collected for both sparse (1500 nodes) and dense settings (5000 nodes) with 25m communication range. We assume the unique ID of each sensor node is 2 bytes. In our proposed algorithm CODAT, the RN sends its own unique ID, first 2 unique IDs received through PCM and a BN-Array (containing status readings and tags). The RN in COBOM sends its own unique ID and BN-Array. Since the size of BN-Array depends on the RN's communication range and the density of the deployed nodes, the greater the density of the sensor field, the larger will be the BN-Array. Since in our algorithm CODAT, the number of BNs and RNs are almost half compared to BNs and RNs in COBOM, we can see in Figure 4-22 and Table 4-2 that average report data size of CODAT does not exceed the COBOM in Dense settings with 5000 nodes. Report data size of CODAT obviously is greater than DEMOCO because DEMOCO does not use BN-Array, however, the main drawback of DEMOCO lies in sending of report data which includes nearest neighbor's ID and this nearest neighbor is estimated based on signal strength which is not very accurate approach. As shown in Figure 4-23 and Table 4-4, the average report data size for CODAT is more compared to DEMOCO and COBOM due to sparse settings with 1500 nodes. In sparse settings, COBOM performs better compared to CODAT because its BN-Array does not contain any neighboring IDs.

The average report data size is calculated based on the following equation 4.1.

$$\tau = \sum_{i=2}^{n} i.U_i \qquad (4.1)$$

Whereas *i* and *U* determines the average report data size i.e *i* is the number of bytes of report data and *U* is the average number of RNs corresponding to the *i*. *τ* is the average report data size for each period. Total report data size is calculated by summing up the average report data size for all periods shown in Table 4-2 and Table 4-4. Also the original results are replicated from the paper [51] and compared with our simulated results shown in Table 4-1 and Table 4-3. Our simulated results for Average Report Data size for 5000 nodes for DEMOCO and COBOM can be compared with original results in Table 4-1. Our simulated results are almost 97 % closed to that of the Original paper for both DEMOCO and COBOM. Similarly for Average Report Data size for 1500 nodes, we compared our simulated results with the original results. We achieved about 98% similarity to that of original results for both DEMOCO and COBOM as shown in Table 4-3.

Table 4-1 Comparison of Original Results with our Simulated Results Average Report Data Size for 5000 nodes

Object's Time Period	DEMOCO[51] Original Bytes	DEMOCO Bytes	COBOM[51] Original Bytes	COBOM Bytes
1	1.692	2.622	3.653	3.645
2	6.546	6.432	13.999	14.002
3	11.526	13.563	24.631	25.432
4	14.376	15.624	30.736	30.431
5	16.503	17.364	35.209	36.252
6	18.243	19.328	38.782	38.521
7	20.304	21.651	43.304	44.652
8	22.368	22.632	47.602	48.524
9	24.435	25.563	52.308	53.152
10	26.61	27.462	56.673	57.342
11	28.668	29.473	61.052	62.452
12	30.657	31.462	65.293	66.373
13	32.685	33.564	69.606	70.352
14	34.902	35.736	74.289	75.262
15	36.741	37.631	78.148	83.463
16	38.637	39.621	82.185	86.983
17	40.671	41.431	86.596	92.146
18	42.807	43.651	91.123	96.473
19	44.739	44.982	95.137	100.183
20	46.803	46.961	99.52	104.454
21	48.668	49.243	103.432	108.474
22	50.673	51.432	107.617	113.464
23	52.851	53.781	112.305	116.464
24	54.51	56.193	115.869	119.474
25	56.511	57.453	120.06	121.632
Total	803.127	823.855	1708.859	1769.602

Table 4-2: Average Report Data Size for 5000 nodes

Object's Time Period	DEMOCO Bytes	COBOM Bytes	CODAT Bytes
1	2.622	3.645	3.543
2	6.432	14.002	10.736
3	13.563	25.432	18.363
4	15.624	30.431	24.374
5	17.364	36.252	27.873
6	19.328	38.521	31.736
7	21.651	44.652	34.733
8	22.632	48.524	37.373
9	25.563	53.152	40.635
10	27.462	57.342	43.625
11	29.473	62.452	48.732
12	31.462	66.373	52.262
13	33.564	70.352	56.367
14	35.736	75.262	59.362
15	37.631	83.463	63.362
16	39.621	86.983	68.373
17	41.431	92.146	72.273
18	43.651	96.473	76.363
19	44.982	100.183	80.373
20	46.961	104.454	83.272
21	49.243	108.474	87.372
22	51.432	113.464	90.373
23	53.781	116.464	93.272
24	56.193	119.474	97.387
25	57.453	121.632	100.102
Total	823.855	1769.602	1401.236

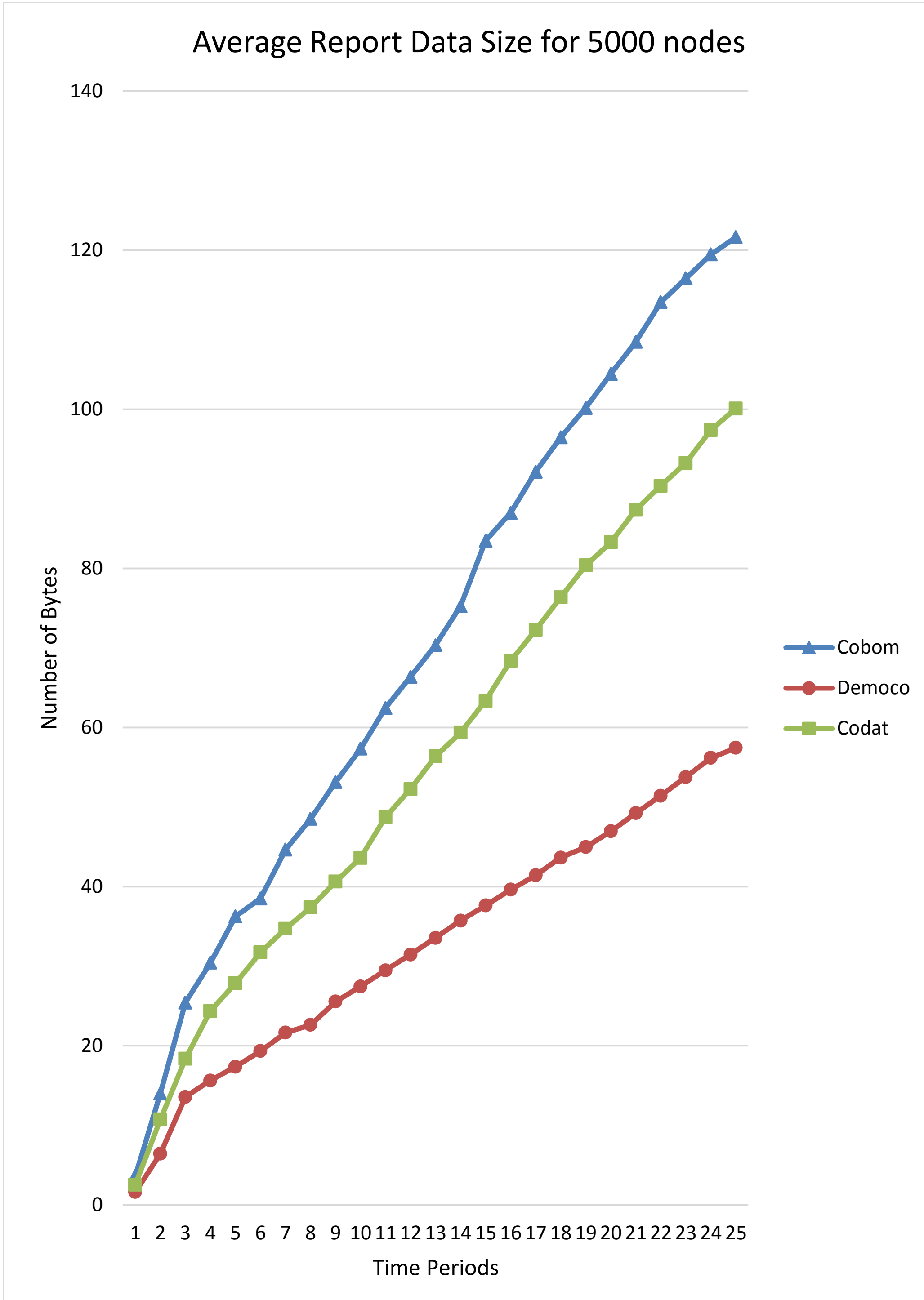


Figure 4-22: Average Report Data Size for 5000 nodes

Table 4-3: Comparison of Original Results with our Simulated Results Average Report Data size for 1500 nodes

Object's Time Period	DEMOCO[51] Original Bytes	DEMOCO Bytes	COBOM[51] Original Bytes	COBOM Bytes
1	0.405	2.404	0.404	2.421
2	1.776	3.767	1.798	3.71
3	4.038	4.04	4.053	4.05
4	6.864	6.922	6.826	6.67
5	9.318	9.55	9.196	9.811
6	11.091	11.095	10.954	10.391
7	12.522	12.571	12.332	12.491
8	14.004	14.244	13.791	13.998
9	15.639	15.537	15.422	15.501
10	17.436	17.692	17.149	17.718
11	19.05	19.127	18.714	18.621
12	20.611	20.73	20.215	20.321
13	22.2	22.491	21.782	21.971
14	24.06	24.81	23.64	23.86
15	25.503	25.525	25.048	25.029
16	27.902	28.13	26.752	26.964
17	28.902	29.3	28.39	28.36
18	30.267	30.393	29.724	29.721
19	31.956	31.965	31.379	31.357
20	33.552	33.635	32.951	32.957
21	35.151	35.602	34.512	34.691
22	36.831	36.846	36.151	36.158
23	38.427	38.018	37.73	37.7
24	39.78	39.8	39.02	39
25	41.352	41.435	40.628	40.621
Total	547.962	551.698	538.566	544.09

Table 4-4 Average Report Data Size for 1500 nodes

Object's Time Period	DEMOCO Bytes	COBOM Bytes	CODAT Bytes
1	2.404	2.421	3.405
2	3.767	3.71	3.772
3	4.04	4.05	4.09
4	6.922	6.67	6.98
5	9.55	9.811	9.89
6	11.095	10.391	11.192
7	12.571	12.491	12.862
8	14.244	13.998	14.465
9	15.537	15.501	15.865
10	17.692	17.718	17.834
11	19.127	18.621	19.432
12	20.73	20.321	20.789
13	22.491	21.971	23.211
14	24.81	23.86	24.258
15	25.525	25.029	25.512
16	28.13	26.964	28.341
17	29.3	28.36	29.262
18	30.393	29.721	30.321
19	31.965	31.357	32.291
20	33.635	32.957	33.532
21	35.602	34.691	35.783
22	36.846	36.158	36.971
23	38.018	37.7	38.957
24	39.8	39	40.761
25	41.435	40.621	41.89
Total	551.698	544.09	556.666

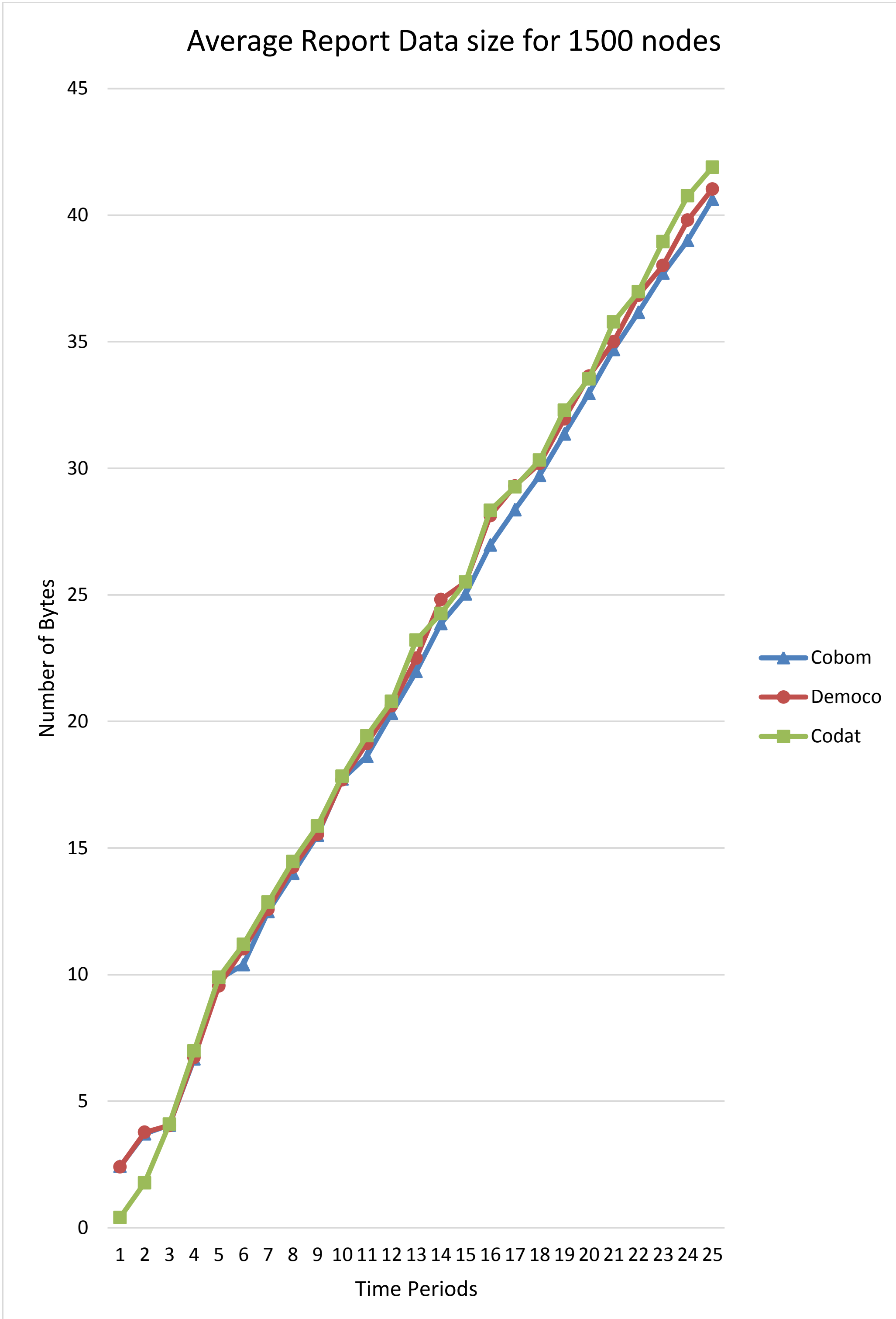


Figure 4-23: Average Report Data Size for 1500 nodes

4.5 Boundary Accuracy with different Diffusion rates of object

In order to analyze the accuracy of our algorithm in detecting the boundary of the continuous object, we simulated about 5000 sensor nodes uniformly distributed over 500 x 500 m^2 with 25m communication range for each sensor node. Accuracy is computed by comparing the reported data at the boundary with that of sink at same time so that we can know that either the actual boundary of the object exists at the specific location or not, when the sink receives the reported data. We set three different speeds for the continuous object (Staircase Structure) with Low diffusion rate, Average diffusion and High diffusion rate. Each Time period is of 5 seconds and the reported data reaches to the sink in 4 seconds. For Low diffusion rate the object expands about 3 meters in each time period. For Average diffusion rate, the object expands about 7 meters in each time period and for High diffusion rate, the object expands about 11 meters in each time period. The first 10 time periods are used in order to calculate the preciseness of the boundary. At the end of each time period, boundary preciseness is calculated.

4.5.1 Boundary Accuracy with Low Diffusion rate of object

The Figure 4-24 shows the accuracy of the boundary detection of the object expanding at low diffusion rate i.e. expands 3 meters in each time period. The x axis of the Figure 4-24 represents the number of time periods and the y axis represents the accuracy in percentage. As we can see that CODAT has high accuracy of about 86 % because of more number of IDs compared to COBOM and DEMOCO. COBOM's and DEMOCO's accuracies are almost 80 %.

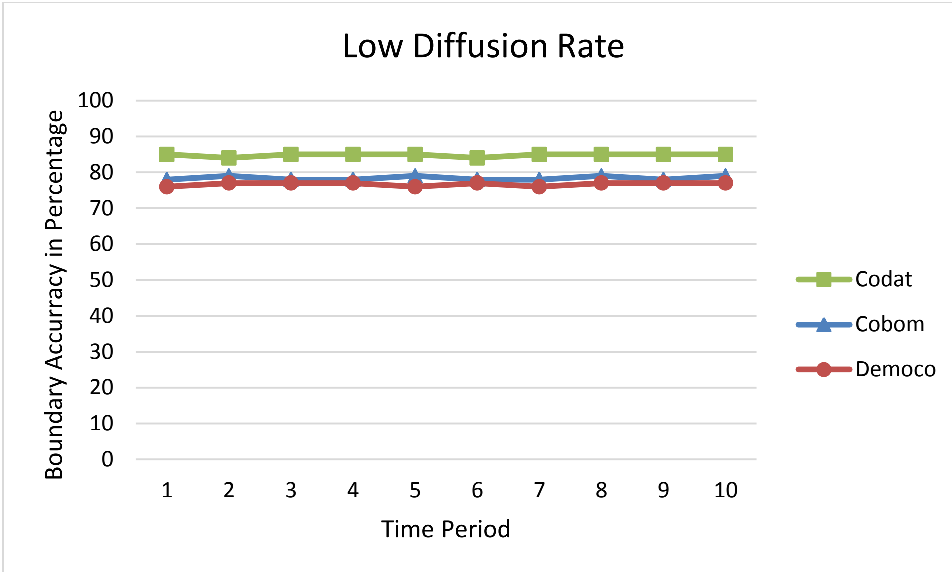


Figure 4-24: Boundary Accuracy with Low Diffusion Rate of Object

4.5.2 Boundary Accuracy with Average Diffusion rate of object

The Figure 4-25 shows the accuracy in boundary detection of the object expanding with average diffusion rate i.e. 7 meters in each time period. The x axis represents the number of periods and the y axis represents the boundary accuracy in percentage. It can be noticed in the Figure 4-25 that accuracy for all the algorithms have decreased compared to the low diffusion rate. Our proposed approach CODAT's accuracy has decreased to 76% compared to Low Diffusion rate. COBOM's and DEMOCO's accuracies have decreased to 68%.

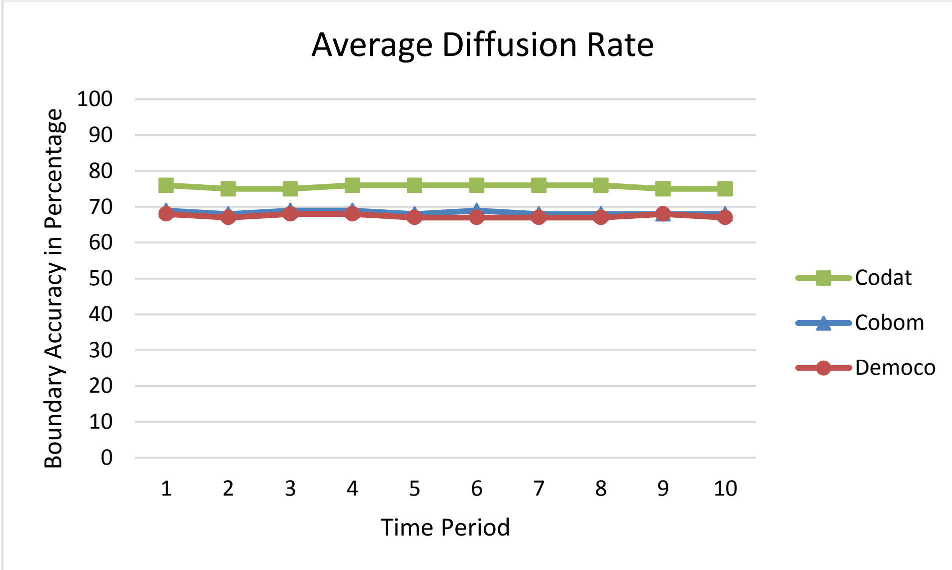


Figure 4-25: Boundary Accuracy with Average Diffusion rate of Object

4.5.3 Boundary Accuracy with High Diffusion rate of object

The Figure 4-26 shows the accuracy in boundary detection of the object expanding with high diffusion rate i.e. object expanding 11 meters in each time period. The x axis represents number of time periods and the y axis represents the boundary accuracy in percentage. Accuracy for all the three algorithms COBOM, CODAT and DEMOCO is calculated at each time period. It is noticed that with the increase in diffusion rate, the overall accuracy of all the algorithms have decreased. Our proposed approach CODAT's accuracy has decreased to 60% due to high diffusion rate of phenomenon. COBOM's and DEMOCO's accuracies have decreased to 52%.

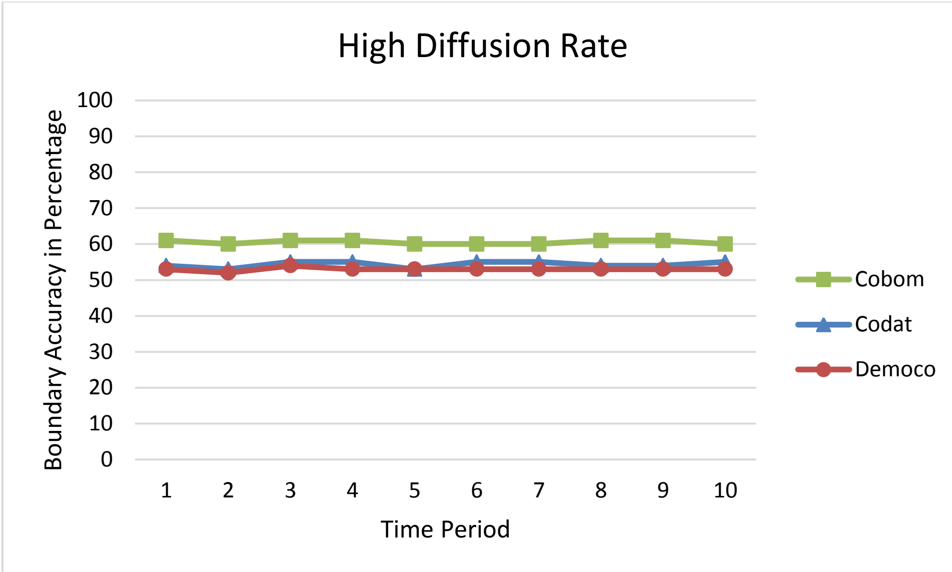


Figure 4-26: Boundary Accuracy with High Diffusion Rate of Object

4.5.4 Boundary Accuracy of CODAT with different Diffusion rates of object

The Figure 4-27 shows the overall picture of the performance of CODAT with different diffusion rates. The x axis represent the time periods and the y axis represents the boundary accuracy in percentage. If we notice the results in detail we can see that with low diffusion rate of object, CODAT's accuracy is up to 86 %, with average diffusion rate of object, CODAT's accuracy is up to 76% and with high diffusion rate of object, CODAT's accuracy is up to 60 %. This shows that the diffusion rate of object has direct impact on the accuracy. Higher the diffusion rate of the phenomenon, lower will be the accuracy. Similarly lower the diffusion rate of the phenomenon, higher will be the accuracy.

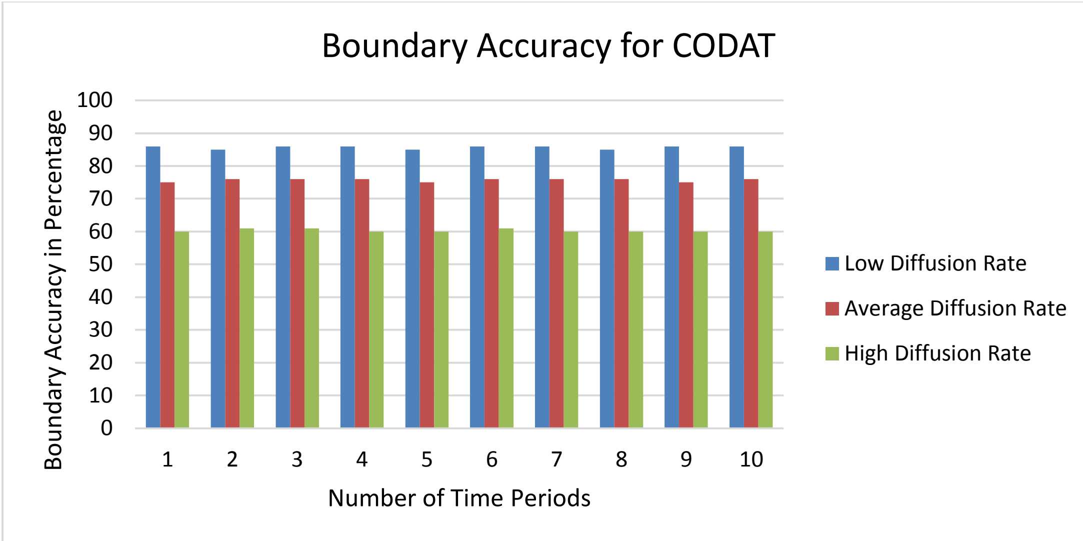


Figure 4-27: Boundary Accuracy of CODAT with different Diffusion Rates of Object

CHAPTER 5

CONCLUSION

In this thesis we concentrated on detection and tracking of continuous phenomenon due to its specific nature such as continuous expansion and shrinkage of phenomenon, splitting and merging of phenomenon and holes inside the phenomenon. Since wireless sensors have limited battery capacity, detection and tracking algorithms have to be energy efficient to increase the overall life time of the sensor network. We proposed an energy efficient algorithm CODAT (Continuous Object Detection and Tracking) with a suitable data structure that is sent by the Reporting nodes to the sink in addition of unique IDs and Phenomenon Tags. Moreover we used the concept of Phenomenon Tags to differentiate between phenomenon changes and holes inside the phenomenon as an added information forwarded to the user while maintaining the high accuracy and efficiency. CODAT is an improved hybrid algorithm integrating the detection techniques of both previous algorithms COBOM and DEMOCO. CODAT outperforms COBOM in terms of efficiency and outperforms both COBOM and DEMOCO in terms of accuracy.

In future we plan to test our algorithm in a real sensor environmental setup in order to analyze it under different environmental conditions.

References

[1] Akyildiz, Ian F., et al. "A survey on sensor networks." Communications magazine, IEEE 40.8 (2002): 102-114.

[2] Chong, Chee-Yee, and Srikanta P. Kumar. "Sensor networks: evolution, opportunities, and challenges." Proceedings of the IEEE 91.8 (2003): 1247-1256.

[3] Guide—Accelerometers, Crossbow Product. "Crossbow Technology." Inc., webpages: 1-3.

[4] Frantz, Gene. "Digital signal processor trends." Micro, IEEE 20.6 (2000): 52-59.

[5] Trimberger, Stephen. Field-programmable gate array technology. Springer, 1994.

[6] Einspruch, Norman, ed. Application specific integrated circuit (ASIC) technology. Academic Press, 1991.

[7] Instruments, Texas. "Chipcon cc2531 datasheet." Online] http://focus. ti.com/lit/ds/symlink/cc2531. pdf [Last accessed 11th June 2011] (2007).

[8] Datasheet, MICAz. "Crossbow Technology Inc." San Jose, California (2006).

[9] Atmel ATMega128L, A. V. R. "Microcontroller Datasheet."

[10] Sudevalayam, Sujesha, and Purushottam Kulkarni. "Energy harvesting sensor nodes: Survey and implications." Communications Surveys & Tutorials, IEEE 13.3 (2011): 443-461.

[11] Levis, Philip, et al. "Tinyos: An operating system for sensor networks." Ambient intelligence. Springer Berlin Heidelberg, 2005. 115-148.

[12] Shapiro, Marc, et al. "SOS: An Object-Oriented Operating System – Assessment and Perspectives." Computing Systems 2.4 (1989): 287-337.

[13] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. 2004. Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04). IEEE Computer Society, Washington, DC, USA, 455-462.

[14] AtmelAVRDevices. Availableonline: http://www.atmel.com/products/avr/ (accessed on 17 April 2011)

[15] Shah Bhatti, James Carlson, Hui Dai, Jing Deng, Jeff Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, and Richard Han. 2005. MANTIS

OS: an embedded multithreaded operating system for wireless micro sensor platforms. Mob. Netw. Appl. 10, 4 (August 2005), 563-579.

[16] Eswaran, A.; Rowe, A.; Rajkumar, R., "Nano-RK: an energy-aware resource-centric RTOS for sensor networks," Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International , vol., no., pp.10 pp.,265, 8-8 Dec. 2005.

[17] Rowe A, Mangharam R, Rajkumar R. FireFly: A Time Synchronized Real-Time Sensor Networking Platform. Proceedings of the 28th IEEE Real-Time Systems Symposium; Tucson, AZ, USA. 3–7 December 2007.

[18] Cao Q, Abdelzaher T, Stankovic J, He T. The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks. Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008); St Louis, MO, USA. 22–24 April 2008.

[19] Howitt, Ivan, et al. "Wireless industrial sensor networks: Framework for QoS assessment and QoS management." ISA transactions 45.3 (2006): 347-359.

[20] Hill, Jason, et al. "System architecture directions for networked sensors." ACM SIGOPS operating systems review. Vol. 34. No. 5. ACM, 2000

[21] Martinez, Kirk, et al. "Deploying a sensor network in an extreme environment." Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on. Vol. 1. IEEE, 2006.

[22] Aberer, Karl, et al. "OpenSense: Open community driven sensing of environment." Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming. ACM, 2010.

[23] Yang, Guang-Zhong. Body sensor networks. Vol. 6. London: Springer, 2006.

[24] www.streetline.com/ - Cached

[25] M. Salamah, F. Zawaideh, and F. Zawaideh, “Optimal object tracking via wireless sensor networks,” in 2013 International Conference on Electronics, Computer and Computation (ICECCO), 2013, pp. 273–276.

[26] Chauhdary, S. H., et al. "EOATR: Energy Efficient Object Tracking by Auto Adjusting Transmission Range in Wireless Sensor Network." Journal of Applied Sciences 9.24 (2009): 4247-4252.

[27] Chaudhary, Sajjad Hussain, Ali Kashif Bashir, and Myong-Soon Park. "[ETCTR] Efficient Target Localization by Controlling the Transmission Range in Wireless

Sensor Networks." Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on. Vol. 1. IEEE, 2008.

[28] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. 2002. Wireless sensor networks for habitat monitoring. In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02). ACM, New York, NY, USA, 88-97.

[29] L.M. Kaplan, Q. Le, and P. Molnar, Maximum likelihood methods for bearings-only target localization, Proceedings of IEEE ICASSP, PP.554-557, Salt Lake City, UT, May 2001.

[30] J.S. Scholl, L.P. Clare, and J.R. Agre, Wavelet packet-based target classification schemes, Meetings of IRIS Specialty Group on Acoustic and Seismic Sensing, Laurel, MD, September, 1998.

[31] K. Yao, R.E. Hudson, C.W. Reed, T.L. Tung, D. Chen, and J. Chen, Estimation and tracking of an acoustic/seismic source using a beamforming array based on residual minimizing methods, In the Proceedings of IRIA-IRIA, pp. 152-163, January 2001.

[32] B. Krishnamachari, S.B. Wicker, and R. Bejar, Phase transition phenomena in wireless ad-hoc networks, In the Proceedings of IEEE GLOBECOM, San Antonio, TX, November 2001.

[33] D. Li, K. Wong, Y. Hu, and A. Sayeed, Detection, classification and tracking of targets in distributed sensor network, IEEE Signal Processing Magazine, vol 19, no.2, pp.17-29, March 2002.

[34] F. Zhao, J. Shin, and J. Reich, Inforamtion-driven dynamic sensor collaboration for target tracking, IEEE Signal Processing Magazine, Col 19, no. 2, pp.61-72, March 2002.

[35] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," IEEE Transaction on Wireless Communication, vol. 3, no. 5, Sept. 2004.

[36] W. Zhang and G. Cao,"Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks" in Proc. IEEE INFOCOM, vol. 4, pp. 2434-2445, Mar. 2004.

[37] Yingqi Xu, Winter J, and Wang-Chien Lee, "Prediction-Based Strategiesfor Energy Saving in Object Tracking Sensor Networks" in Proc. IEEE MDM, pp. 346-357, Jan. 2004.

[38] Yingqi Xu, Winter J, and Wang-Chien Lee, “Dual Prediction-Based Reporting for Object Tracking Sensor Networks,” in Proc. IEEE MOBIQUITOUS, pp. 154-163, Aug. 2004.

[39] Wei-Peng Chen, Jennifer C. Hou, and Lui Sha, “Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks,” IEEE Trans. On Mobile Computing, vol. 3, issue. 3, pp. 258-271, Aug. 2004.

[40] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Toward sophisticated sensing with queries. In IPSN, pages 63–79, 2003.

[41] I. Solis and K. Obraczka. Efficient continuous mapping in sensor networks using isolines. In Proc. of the 2005 MobiQuitous, pages 325–332, 2005.

[42] I. Solis and K. Obraczka. Isolines: Energy-efficient mapping in sensor networks. In ISCC, pages 379–385, 2005.

[43] X. Meng, L. Li, T. Nandagopal, and S. Lu. Event contour:an efficient and robust mechanism for tasks in sensor networks. Technical Report, UCLA, 2001.

[44] A. Silberstein, R. Braynard, and J. Yang. Constraint chaining: on energyefficient continuous monitoring in sensor networks. In Proc. of the 2006 ACM SIGMOD Intl. Conf. on Management of Data, pages 157–168, 2006.

[45] K. Chintalapudi and R. Govindan. Localized edge detection in sensor fields. Ad Hoc Networks, 1(2-3):273–291, 2003.

[46] M. Ding, D. Chen, K. Xing, and X. Cheng. Localized fault-tolerant event boundary detection in sensor networks. In INFOCOM, pages 902–913, 2005.

[47] Y. Kotidis. Snapshot queries: Towards data-centric sensor networks. In Proc. of the 2005 Intl. Conf. on Data Engineering, pages 131–142, 2005.

[48] Zha, Hongyuan, J. J. Metzner, and George Kesidis. "Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks." Communications, 2004 IEEE International Conference on. Vol. 7. IEEE, 2004.

[49] Chang, Wang-Rong, Hui-Tang Lin, and Zong-Zhi Cheng. "CODA: a continuous object detection and tracking algorithm for wireless ad hoc sensor networks." Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE. IEEE, 2008.

[50] Zhong, Cheng, and Michael Worboys. "Energy-efficient continuous boundary monitoring in sensor networks." Technical Report (2007).

[51] Jung-Hwan, K. I. M., et al. "DEMOCO: Energy-efficient detection and monitoring for continuous objects in wireless sensor networks." IEICE transactions on communications 91.11 (2008): 3648-3656.

[52] Hosung Park; Seungmin Oh; Euisin Lee; Soochang Park; Sang-ha Kim; Wanseop Lee, "Selective wakeup discipline for continuous object tracking in grid-based wireless sensor networks," Wireless Communications and Networking Conference (WCNC), 2012 IEEE , vol., no., pp.2179,2184, 1-4 April 2012

[53] Hyungseop Hong; Seungmin Oh; Jeongcheol Lee; Sang-ha Kim, "A Chaining Selective Wakeup Strategy for a Robust Continuous Object Tracking in Practical Wireless Sensor Networks," Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on , vol., no., pp.333,339, 25-28 March 2013

[54] J. Shen, G. Han, J. Jiang, N. Sun, and L. Shu, "An energy-efficient tracking scheme for continuous objects in duty-cycled wireless sensor networks," in 2015 IEEE International Conference on Consumer Electronics - Taiwan, 2015, pp. 150–151

[55] Seung-Woo Hong, Ho-Yong Ryu, Soochang Park, and Sang-Ha Kim, "Reliable continuous object tracking with cost effectiveness in wireless sensor networks," in 2015 Seventh International Conference on Ubiquitous and Future Networks, 2015, pp. 672–676.

[56] C. S. Hussain, M. S. Park, A. K. Bashir, S. C. Shah, and J. Lee, "A Collaborative Scheme for Boundary Detection and Tracking of Continuous Objects in Wsns," Intell. Autom. Soft Comput., vol. 19, pp. 439–456, 2013.

Vitae

Name	: SHEHR YAR KHAN
Nationality	: Pakistani
Date of Birth	: 17/02/1989
Email	: engr.shehryar89@gmail.com
Address	: Pakistan Aeronautical Complex, Kamra, Pakistan
	: Postal Code 43570

Education

2013-2015: MS Computer Networks CGPA 3.43
Computer Engineering Department
King Fahd University Of Petroleum and Minerals, KFUPM , Saudi Arabia

Thesis Title: *Continuous Objects Detection and Tracking in Wireless Sensor Networks*
Thesis Supervisor: ***Dr. Tarek R. Sheltami***

2008-2012: BS in Telecommunication Engineering with CGPA 3.80 (GOLD MEDALIST)
National University of Computer & Emerging Sciences, FAST, Pakistan

Publications

Journals

1. Basem Almadani, **Shehryar Khan**, Muhammad Naseer Bajwa, Tarek R. Sheltami, and Elhadi Shakshuki, "AVL and Monitoring for Massive Traffic Control System over DDS," *Mobile Information Systems*, vol. 2015, Article ID 187548, 9 pages, 2015. doi:10.1155/2015/187548
Impact Factor: 0.949 **(ISI Indexed)**

2. Tarek R. Sheltami, **Shehryar Khan**, Elhadi Shakshuki "Continuous Objects Detection and Tracking in Wireless Sensor Networks", *Personal and Ubiquitous Computing*. ***Submitted***

Conferences

1. Basem Almadani, **Shehryar Khan**, Tarek R Sheltami, Elhadi M Shakshuki, Muhammad Musaddiq, Bilal Saeed, "Automatic Vehicle Location and Monitoring System based on Data Distribution Service, Elsevier Procedia Computer Science Vol. 37, pp. 127-134, 2014

2. Muhammad Musaddiq, **Shehryar Khan**, Uthman Baroudi, and Bilal Saeed. 2014. Performance Evaluation of IEC 61850 under Wireless Communication Networks. In *Proceedings of the 6th International Conference on Management of Emergent Digital EcoSystems* (MEDES '14). ACM, New York, NY, USA, pp. 90-94

Research Experience (Sept 2013-Present)

- **Continuous Objects Detection and Tracking in Wireless Sensor Networks (MS *Thesis)***

Proposed an energy efficient algorithm for the detection and tracking of continuous phenomenon. Detection of continuous objects such as wildfire, oil spills require extensive communication among nodes which in turn affects the life time of sensor networks. Our goal was to reduce the communication cost while maintaining the accuracy and precision for the detection and monitoring of continuous objects. Simulation tool **JAVA**

Supervisor: Dr. Tarek R. Sheltami

- **Automatic Vehicle Location and Monitoring System based on DDS**

Proposed a Real-Time Automatic Vehicle Location (AVL) and Monitoring system for traffic control of pilgrims coming towards the holy city of Makkah in Saudi Arabia based on data distribution service (DDS) specified by the Object Management Group (OMG), which is a Real-Time Publish/Subscribe Middleware.

Supervisor: Dr. Basem Almadani

- **Performance Evaluation of IEC 61850 under Wireless Communication Networks**

The standard IEC 61850 is considered as a candidate for communication standard for smart grid applications. The delay performance is one of the critical issues that were specified by IEC 61850. This project involved the modeling and simulation of IEC 61850 substation automation system (SAS) under wireless and hybrid networks. **OPNET** is used to model and simulate the Intelligent Electronic Device (IEDs).

Supervisor: Dr. Uthman Baroudi

Industry Experience

Internee at National Network Operation Center, PTCL Islamabad
(Jan 2013-June 2013)

- Managing the nationwide operations & maintenance of IP/MPLS, IP DCN & IP Enterprise Network.
- Ensuring the connectivity within the Core network having IP MPLS backbone.
- Network monitoring and troubleshooting using CA Spectrum, Cacti
- Ensuring through reporting node availability, port/link utilizations and IP SLA based QoS Parameters to plan and maintain a consistent network.

Programming and Simulation Skills

- Java ,C++, Python
- NS2,OPNET Modeler, Packet Tracer

Graduate Courses

1. Client Server Programming
2. Computer Networks
3. Computer Networks Design
4. Computer and Network Security
5. Modeling and Simulation
6. Heterogeneous Computing
7. Pervasive and Ubiquitous Computing

Professional Certifications

Cisco Certified Network Professional (Routing and Switching)

June 2015

Cisco Certified Network Associate (Routing and Switching)

May 2013

Honors and Awards

- Awarded fully funded Scholarship for Master's studies at KFUPM Saudi Arabia
- Overall First Position awarded **GOLD Medal** in Bachelor of Science
- Remained in Dean's List of Honors' for 7 times
- Awarded Tuition Scholarship from PAF Shaheen Foundation Pakistan for BS