

0-2557



**ENERGY EFFICIENT DATA COLLECTION IN
WIRELESS SENSOR NETWORKS**

By

P. MANIVANNAN

Reg. No. : 0720108010

of

KUMARAGURU COLLEGE OF TECHNOLOGY
(An Autonomous Institution affiliated to Anna University, Coimbatore)

COIMBATORE – 641 006

A PROJECT REPORT

Submitted to the

**FACULTY OF INFORMATION AND COMMUNICATION
ENGINEERING**

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MAY 2009



BONAFIDE CERTIFICATE

Certified that this project report titled” **ENERGY EFFICIENT DATA COLLECTION IN WIRELESS SENSOR NETWORK**” is the bonafide work of **Mr.P.MANIVANNAN (0720108010)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report of dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.



GUIDE

Mrs.V.Vanitha ME.,
Assistant Professor,
Department of Computer
Science and Engineering



HEAD OF THE DEPARTMENT

Dr. S.Thangasamy, Ph.D
Professor and Dean,
Department of Computer
Science and Engineering

The candidate with **University Register No. 0720108010** was examined by us in Project Viva-Voce examination held on 04/05/2009



INTERNAL EXAMINER



EXTERNAL EXAMINER



Departments of Computer Science and Engineering & Information Technology



CSI COLLEGE OF ENGINEERING

CET '09

Ketti, The Nilgiris- 643 215

THIRD NATIONAL CONFERENCE ON EMERGING TECHNOLOGIES - 2009

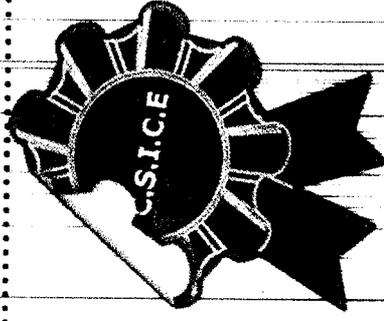
CERTIFICATE

This is to certify that **Dr./Prof./Mr./Ms. MANI.VANNAN. P** of **KUMARAGURU COLLEGE OF TECHNOLOGY** has participated in **CET '09** and presented a paper entitled **ENERGY EFFICIENT DATA COLLECTION IN WIRELESS SENSOR NETWORKS** at the conference held on 11th March 2009.

Authors: **Mr. P. Mannivannan, Mrs. V. Vanitha**

O-ORDINATOR

CONVENER



PRINCIPAL

ABSTRACT

Wireless sensor networks are a trend of the past few years, and they involve deploying a large number of small nodes. The nodes then sense environmental changes and report them to other nodes over flexible network architecture. Sensor nodes are great for deployment in hostile environments or over large geographical areas.

In a traditional reliable data delivery paradigm, hop-by-hop ARQ and end-to-end feedback mechanisms are used comprehensively to guarantee packet delivery. Conventional reliability schemes generally do not take energy efficiency into consideration. However in wireless sensor networks (WSN), the greatest challenge is the development of long-lived sensor networks in spite of energy-constraints of individual nodes.

Energy conservation is one of the primary requirements in WSN. Strict data delivery guarantee for individual packets is not required in many WSN applications. Field deployment of sensor network applications, such as environmental monitoring and animal tracking; show that occasional loss of sensor readings is tolerable as long as the collective information from all the source nodes can be obtained.

An Efficient Reliable Data Collection (eRDC) algorithm is used to reduce the number of retransmission between source and sink, and to balance reliability and energy consumption in wireless sensor network. Dynamic programming concept is used to reduce the number of intermediate nodes. With the help of ARQ mechanism the retransmission is done during packet loss. The performance of this algorithm is compared with the existing algorithm based on the number of retransmission and energy consumption.

ஆய்வுச்சுருக்கம்

கடந்த சில ஆண்டுகளாக கம்பியில்லா உணரி பிணையத்தை பயன்படுத்துகிறோம். அவற்றில் பல சிறிய கணுக்களை ஏதேனும் சூழ்நிலையில் மாற்றம் ஏற்பட்டால் அவற்றை அருகிலுள்ள கணுக்களாக அறுவுக்கிறோம்.

பழங்காலங்களில், முடிவுட்டு முடிவு முறையை பயன்படுத்தி கட்டாயமாக தகவல் பெட்டகம் அனுப்புதல், ஆனால் ஆற்றல் மிச்சப் படுத்தும் மற்றும் தகவல் பெட்டகத்தில் இழப்பீடு ஏற்பட்டால் அவற்றை திரும்ப அனுப்புதல் போன்றவற்றை நா எடுத்து கொள்வதில்லை. அதுமட்டுமல்லது கம்பியில்லா உணரி பிணையத்தில் உணரி பிணையத்தின் வாழ்னாளை அதிகரிப்பது மிகவும் சவாலக்கிறது. இந்த ஆய்வில் நமது நோக்கமானது கம்பியில்லா உணரி பிணையத்தின் ஆற்றலை மிச்சப்படுத்தல் மற்றும் நம்பகத்தன்மையை ஏற்படுத்த வேண்டும்.

ஆற்றல் பயன்படுத்துவது என்பஹ்டு கம்பியில்லா உணரி பிணையத்தின் முக்கிய செயலாகும். கம்பியில்லா உணரி பிணையத்தின் பயன்பாடுகளில் தரவு அனுப்புதல் என்பது அவசியமில்லை. கம்பியில்லா உணரி பிணையம் துறைகளில் பயன்படுத்துகிறோம். அதாவது சூழ்னில்லை மேற்பாற்றவை, மிருகங்களை தேடுதல். எப்போதாவது தரவு இழுப்பீடு ஏற்றுக் கொள்ள கூடியது.

இந்த ஆய்வில் பயன்படுத்தும் படிமுறையை பயன்படுத்தி திரும்ப அனுப்பதலின் எண்ணிக்கை, நம்பகத்தன்மை மற்றும் ஆற்றல் மிச்சப்படுத்துதல் ஆகியவற்றை கம்பியில்லா உணரி பிணையத்தில் அமுல்படுத்த வேண்டும். ப்ய்னமிக் ப்ரொக்ரம்மிங்க் முறையை பயன்படுத்தி இடைகணுக்கள் குறைக்கப்படுகின்றன. ஆற்ஃ பயன்படுத்தி திரும்ப அனுப்புதல் எண்ணிக்கை குறைக்கப்படுகிறது. இந்த படிமுறையை படிங்கால படிமுறைகளுடன் ஒப்பிடுகையில், இந்த படிமுறை திரும்ப அனுப்பதல் எண்ணிக்கை மற்றும் ஆற்றலையும் மிச்சப்படுத்துகிறது.

ACKNOWLEDGEMENT

I express my profound gratitude to our Chairman **Padmabhusan Arutselver Dr. N. Mahalingam B.Sc, F.I.E** for giving this great opportunity to pursue this course.

I express my profound gratitude to our esteemed Correspondent Professor **Dr.K.Arumugam, B.E. (Hons), MS (USA), M.I.E.**, for giving this great opportunity to pursue this course.

I thank **Dr. Joseph V. Thanikal, M.E., Ph.D., PDF, CEPIT** Principal, and **Prof. R. Annamalai**, Vice Prinipal, Kumaraguru College of Technology, Coimbatore, for providing me with the necessary facilities and Infrastructure to work on this project.

I thank **Dr.S.Thangasamy, Ph.D.**, Dean and Head of the Department of Computer Science and Engineering for being my greatest of inspiration and for embedding the quest for innovative ideas.

I express my heartiest thanks to our Project Coordinator as well as my Project Guide **Mrs.V.Vanitha M.E.**, Assistant Professor, Department of Computer Science and Engineering who rendered her valuable guidance and support to perform my project work extremely well.

I would like to convey my honest thanks to all Teaching staff members and Non Teaching staff members of the department for their support. I would like to thank all my classmates who gave me proper light moments and study breaks apart from extending some technical support whenever I needed them most.

My work will be incomplete without expressing my gratitude to various authors whose works were referred to carry out this project.

TABLE OF CONTENTS

TITLE	PAGE NO
Abstract	iv
Abstract (Tamil)	v
List of Figures	ix
List of Abbreviations	x
CHAPTER 1 INTRODUCTION	1
1.1 Overview of wireless sensor networks	1
1.1.1 Sensing process	1
1.2 Components of sensor networks	3
1.3 Applications of sensor networks	3
1.4 Data Collection Methodology	4
1.4.1 Event-based data collection	5
1.4.2 Periodic data collection	5
1.5 Challenges of wireless sensor networks	6
1.6 Relevance and importance of the problem	9
CHAPTER 2 LITERATURE SURVEY	12
CHAPTER 3 PROBLEM DEFINITION	18
CHAPTER 4 PROPOSED METHODOLOGY	19

4.1 Energy Efficiency Analysis	19
4.1.1 Energy Efficiency of ESRT	19
4.1.2 Energy Efficiency of ARQ	19
4.2 Efficient Reliable Data Collection	20
4.3 ERDC Algorithm	21
CHAPTER 5 RESULTS AND ANALYSIS	23
5.1 The Simulation on ns-2	23
5.2 Results and analysis using NAM	23
5.3 Physical/MAC/Network layers Specifications	25
CHAPTER 6 CONCLUSION AND FUTURE OUTLOOK	28
CHAPTER 7 APPENDICES	29
Appendix 1: Source Code	29
Appendix 2: Screen Shots	44
CHAPTER 8 REFERENCES	46

LIST OF FIGURES

FIGURE NO.	NAME	PAGE NO
1.1.1	Sensing Process	1
1.1.2	Sensor nodes scattered in sensor field	2
1.4.1.1	Data Collection in WSN	4
5.1	Comparison in Throughput --ESRT and ARQ	26
5.2	Comparison in Loss Probability -- ESRT and ARQ	26
5.3	Analysis in no of retransmission reduced --ESRT and ARQ	27
5.4	Energy Efficiency Analysis in -- ESRT and ARQ	27

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Networks
ARQ	Automatic Retransmission Request
ESRT	Event to sink reliable Transmission
QoS	Quality of Service
EPB	Energy per Bit
MAC	Media Access Control
NS2	Network Simulator 2
NAM	Network Animator
TCL	Tool Command Language
TTL	Time to live
AODV	Ad hoc on-demand Distance Vector routing

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF WIRELESS SENSOR NETWORKS

Wireless sensor networks are tiny, battery powered sensor nodes with limited on-board processing, storage and radio capabilities deployed in ad-hoc fashion in the area of interest to monitor events and gather data about the environment at a lower cost than traditional wired sensor system. The sensor nodes are capable of collecting, relaying and processing the sensor readings from the monitored physical world. A sensor node made up of four basic major components as shown in figure 1.1.1: a sensing unit, a processing unit, a transceiver unit and a power unit.

A wireless sensor network is a collection of nodes organized in a network. Each node consist of processing capability, multiple types of memory, a RF transceiver which is usually an Omni directional antenna, a power source and accommodates various sensors and actuators. The nodes communicate wirelessly and often self organized after being deployed in an ad hoc fashion.

1.1.1 Sensing process

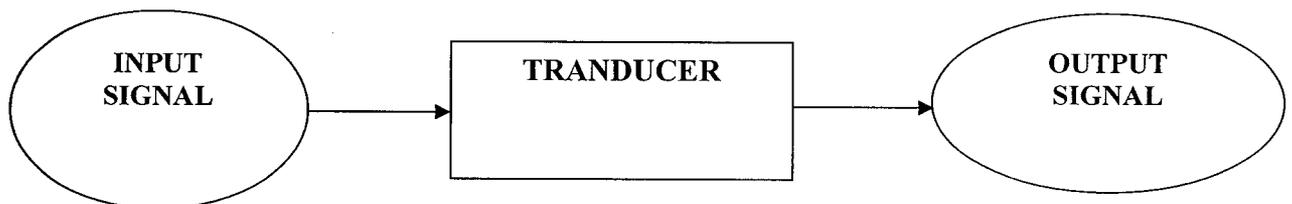


Fig 1.1.1 Sensing Process

Sensor networks, once deployed, are left unattended and expected to work for extended periods of time. Data is being sensed by the sensor nodes in the network must be transmitted to a processing center (base station / sink), where the end user can access the data. A sensor node can communicate directly only with other sensor nodes that are within a small distance. So all the sensor nodes cannot reach the processing center effectively, resulting in more energy consumption to transmit data than the order nodes and hence, exhausts and die sooner. Energy is a scarce resource for sensor systems and has to be managed wisely in order to prolong the life time of the sensor nodes for the duration of a particular mission.

The sensor nodes are usually scattered in a sensor field as shown in Fig 1.1.2. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink. Data are routed back to the sink by multi-hop infrastructure less architecture through the sink. The sink may communicate with the task manager node via Internet and satellite.

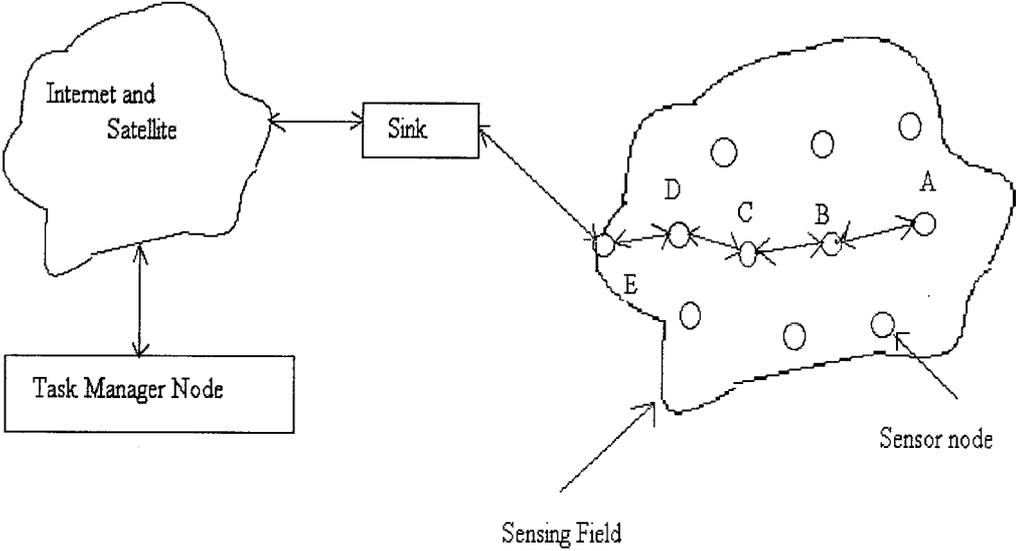


Fig 1.1.2 Sensor nodes scattered in sensor field

Sensor networks have wide application in areas such as environment and habitat monitoring, health care, military, collecting information in disaster-prone areas and surveillance application

1.2 Components of sensor networks

Sensing unit:

Sensing units are usually composed of two sub units: sensors and analog to digital converters. The analog signals produced by the sensors are converted to digital signals by the ADC, and fed into the processing unit.

Processing unit:

The processing unit which is generally associated with a small storage unit manages the procedure that makes the sensor nodes collaborate with the other nodes to carry out the assigned sensing tasks.

Transceiver unit:

A transceiver unit connects the nodes to the networks.

Power unit:

One of the most important components of a sensor node is the power unit. Power units may be supported by a power scanning unit such as solar cells.

1.3 Applications of sensor networks

Wireless sensor networks have a wide range of applications Such as,

1. Military application

- i. Monitoring friendly forces and equipment.
- ii. Battlefield surveillance.
- iii. Nuclear, biological and chemical attack detection.

2. Environmental application

- i. Forest fire.
- ii. Bio complexity mapping of the environment.

- iii. Flood detection.
- iv. Precision agriculture.
- 3. Health application
 - i. Tele monitoring of human physiological data.
 - ii. Tracking and monitoring doctors and patients inside a hospital
 - iii. Drug administration in hospital.
- 4. Home application
 - i. Home automation.
 - ii. Smart environment.

In order to enable reliable and efficient observation and initiate right actions, physical phenomenon features should be reliably detected/estimated from the collective information provided by sensor nodes. Moreover, instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. The intrinsic properties of individual sensor nodes, pose additional challenges to the communication protocols in terms of energy consumption.

1.4 Data Collection Methodology

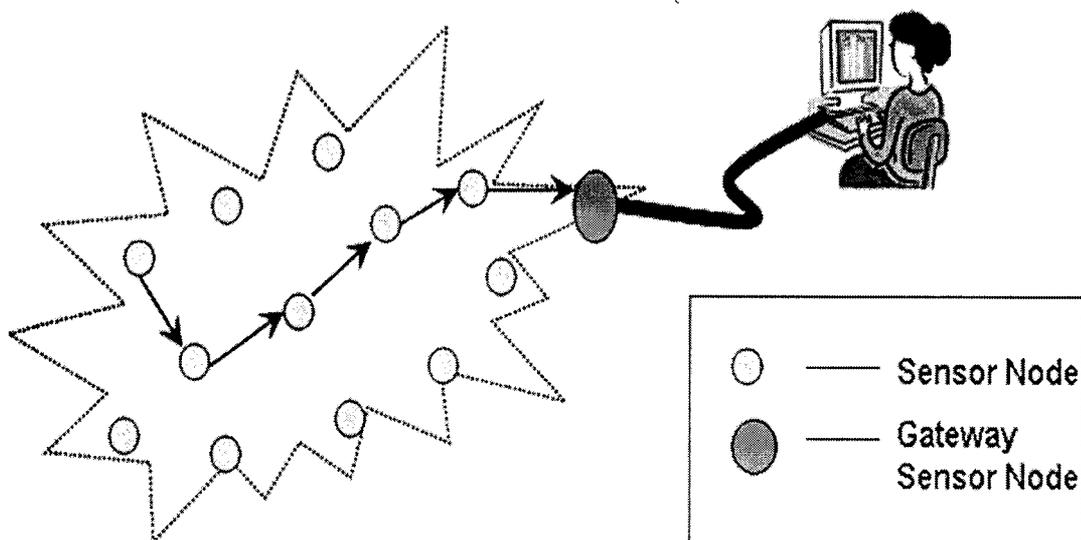


Figure 1.4.1.1 Data Collection in WSN

1.4.1 Event-based data collection

In event-based data collection the sensors are responsible for detecting and reporting events such as spotting moving targets. The event-based data collection is less demanding in terms of the amount of wireless communication, since local filtering is performed at the sensor nodes, and only events are propagated to the base node. In certain applications, the sensors may need to collaborate in order to detect events. Detecting complex events may necessitate nontrivial distributed algorithms that require the involvement of multiple sensor nodes. An inherent downside of the event based data collection is the impossibility of performing an in-depth analysis on the raw sensor readings since they are not extracted from the network in the first place.

1.4.2 Periodic data collection

Periodic updates are sent to the base node from the sensor network based on the most recent information sensed from the environment.

Two approaches are used to classify this,

1. In query-based data collection.
2. Long-standing queries are used to express user or application-specific information interests and these queries are installed “inside” the network. Most of the schemes following this approach support aggregate queries such as minimum, average, and maximum. These types of queries result in periodically generating an aggregate of the recent readings of all nodes. Although aggregation lends itself to simple implementations that enable the complete in-network processing of queries, it falls short in supporting holistic aggregates over sensor readings such as quintiles. Similar to the case of event-based data collection, the raw data is not extracted from the network and a complex data analysis that requires the integration of sensor readings from various nodes at various times cannot be performed with the in-network aggregation

1.5 CHALLENGES OF WIRELESS SENSOR NETWORKS

A sensor network design is influenced by many factors, which include,

- **Energy efficiency/system lifetime**

As sensor nodes are battery-operated, protocols must be energy-efficient to maximize system life time. System life time can be measured such as the time until half of the nodes die or by application-directed metrics, such as when the network stops providing the application with the desired information about the phenomena.

- **Fault Tolerance**

Some sensor nodes may fail or be blocked due to lack of power, have physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures.

- **Scalability**

The number of sensor nodes deployed in studying a phenomenon may be in the order of hundreds or thousands. Depending on the application, the number may reach an extreme value of millions. The new schemes must be able to work with this number of nodes.

- **Production Costs**

Since the sensor networks consist of a large number of sensor nodes, the cost of a single node is very important to justify the overall cost of the networks. If the cost of the network is more expensive than deploying traditional sensors, then the sensor network is not cost-justified. As a result, the cost of each sensor node has to be kept low.

- **Environment**

Sensor nodes are densely deployed either very close or directly inside the phenomenon to be observed. Therefore, they usually work unattended in remote geographic areas. They may be working in busy intersections, in the interior of large machinery, at the bottom of an ocean, inside a twister, on the surface of an ocean. They

work under high pressure in the bottom of an ocean, in harsh environments such as debris or a battlefield, under extreme heat and cold such as in the nozzle of an aircraft engine or in arctic regions, and in an extremely noisy environment such as under intentional jamming.

- **Hardware Constraints**

A sensor node is made up of four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit. Sensing units are usually composed of two subunits: sensors and analog to digital converters (ADCs). The analog signals produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit, which is generally associated with a small storage unit, manages the procedures that enable the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units may be supported by a power scavenging unit such as solar cells. There are also other subunits, which are application dependent. Most of the sensor network routing techniques and sensing tasks require the knowledge of location with high accuracy.

- **Sensor Network Topology**

Sheer numbers of inaccessible and unattended sensor nodes, which are prone to frequent failures, make topology maintenance a challenging task. Hundreds to several thousands of nodes are deployed throughout the sensor field.

- a) Pre-Deployment Phase**

Sensor nodes can be either thrown in mass or placed one by one in the sensor field. They can be deployed by dropping from a plane, delivering in an artillery shell, rocket or missile, throwing by a catapult, placing in factory, and placing one by one either by a human or a robot. Although the sheer number of sensors and their unattended deployment usually preclude placing them according to a carefully engineered deployment plan, the schemes for initial deployment must reduce the installation cost, eliminate the need for any pre-organization and preplanning, increase the flexibility of arrangement, and promote self-organization and fault tolerance.

b) Post-Deployment Phase

After deployment, topology changes are due to change in sensor nodes position, reach ability (due to jamming, noise, moving obstacles, etc.), available energy, malfunctioning, and task details. Sensor nodes may be statically deployed. However, device failure is a regular or common event due to energy depletion or destruction. It is also possible to have sensor networks with highly mobile nodes. Besides, sensor nodes and the network experience varying task dynamics, and they may be a target for deliberate jamming. Therefore, sensor network topologies are prone to frequent changes after deployment.

c) Re-Deployment of Additional Nodes Phase

Additional sensor nodes can be re-deployed at any time to replace the malfunctioning nodes or due to changes in task dynamics. Addition of new nodes poses a need to re-organize the network. Coping with frequent topology changes in an ad hoc network that has myriads of nodes and very stringent power consumption constraints requires special routing protocols.

- **Transmission Media**

In a multi-hop sensor network, communicating nodes are linked by a wireless medium. These links can be formed by radio, infrared or optical media. To enable global operation of these networks, the chosen transmission medium must be available worldwide. One option for radio links is the use of *Industrial, Scientific and Medical* (ISM) bands, which offer license free communication in most countries.

- **Power Consumption**

The wireless sensor node, being a microelectronic device, can only be equipped with a limited power source. In some application scenarios, replenishment of power resources might be impossible. Sensor node lifetime, therefore, shows a strong dependence on battery lifetime. The malfunctioning of few nodes can cause significant topological changes and might require rerouting of packets and re-organization of the network. Hence, power conservation and power management take on additional importance. It is for these reasons that researchers are currently focusing on the design of power aware protocols and algorithms for sensor networks. In sensor networks, power

efficiency is an important performance metric, directly influencing the network lifetime. Application specific protocols can be designed by appropriately trading off other performance metrics such as delay and throughput with power efficiency. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data. Power consumption can hence be divided into three domains: sensing communication, and data processing,

1.6 RELEVANCE AND IMPORTANCE OF THE PROBLEM

Data collection from neighboring sensors is often redundant and highly correlated. In addition, the amount of data collection in large sensor networks is usually enormous for the base station to process. Hence we need methods for combining data into high-quality information at the sensors or intermediate nodes which can reduce the number of packets transmitted to the base station resulting in conservation of energy and bandwidth.

Data collection is defined as the systematic collection of sensed data from multiple sensors to be eventually transmitted to the base station for processing.

The performance measures of data collection algorithms are

- **Energy Efficiency:**

A data collection scheme is energy efficient if it maximizes the functionality of the network. If we assume that all sensors are equally important, we should minimize the energy consumption of each sensor. This idea is captured by the network lifetime which quantifies the energy efficiency of the network.

- **Network Lifetime:**

Network Lifetime is defined as the number of rounds until the first sensor is drained of its energy.

- **Data Accuracy:**

The definition of data accuracy depends on the specific applications for which the sensor network is designed. For instance, in a target localization problem, the estimate of the target location at the sink determines the data accuracy.

- **Latency:**

Latency can be measured as the time delay between the data packets received at the sink and the data collection at the source nodes.

Data collection is defined as the systematic collection of sensed data from multiple sensors to be transmitted to the base station for processing. Data gathered from neighboring sensors is often redundant and highly correlated. In addition, the amount of data collection in large sensor networks is usually enormous for the base station to process. Hence among we need methods for combining data into high quality information at the sensors or intermediate nodes which can reduce the number of packets transmitted to the base station resulting in conservation of energy and bandwidth. There are different Data Collection techniques based on networks.

- Flat Networks
- Hierarchical Networks

In flat networks each sensor nodes are equipped with approximately same battery power.

Data collection is accomplished by data centric routing where the sink usually transmits the query message to the sensors, and the sensors which have data matching the query send response message back to the sink.

A flat network can result in excessive communication and computation burdens at the sink node, resulting in faster depletion of its battery power. The death of the sink node breaks down the functionalities of the network. Hence, in view of scalability and energy efficiency several hierarchical approaches have been proposed. Hierarchical data collection involves data fusion at special nodes which reduces the number of messages transmitted to the sink. This improves the energy efficiency of the network. There are several hierarchical approaches they are,

- **Cluster Based Approach**

In Energy constrained sensor networks of large size, it is inefficient for sensors to transmit the data directly to the sink. In such scenarios, sensors can transmit data to a local aggregator or cluster head which aggregates data from all the sensors in its cluster

and transmits the data to the sink. This results in significant energy saving for the energy-constrained sensor networks.

- **Chain Based Approach**

In Cluster based sensor networks, sensors transmit data to the cluster head where data aggregation is performed. However if the cluster head is far away from the sensors they might expend excessive energy in communication. Hence further improvements in energy efficiency can be obtained if sensors transmit only to close neighbors. The key idea behind the chain based approach is that each sensor transmits data only to its neighbors. The farthest node from the sink initiates chain formation and at each step the closest neighbor of the node is selected as its successor in the chain.

- **Tree Based approach**

In tree based approach a spanning tree rooted at the sink is constructed. Such a structure is exploited in answering a queries collection by the sink. This is done by performing in-network aggregation along the aggregation tree by proceeding level by level from leaves to its root. Thus as two or more messages get to a given node the aggregation can be computed.



CHAPTER 2

LITERATURE SURVEY

2.1 ESRT: Event to Sink Reliable Transport in Wireless Sensor Networks

Wireless sensor networks (WSN) are event based systems that rely on the collective effort of several micro sensor nodes. Reliable event detection at the sink is based on collective information provided by source nodes and not on any individual report. Hence, conventional end-to-end reliability definitions and solutions are inapplicable in the WSN regime and would only lead to a waste of scarce sensor resources. In order to address this need, a new reliable transport scheme for WSN, the event-to-sink reliable transport (ESRT) protocol, is presented. ESRT is a novel transport solution developed to achieve reliable event detection in WSN with minimum energy expenditure. It includes a congestion control component that serves the dual purpose of achieving reliability and conserving energy. Importantly, the algorithms of ESRT mainly run on the sink, with minimal functionality required at resource constrained sensor nodes. ESRT protocol operation is determined by the current network state based on the reliability achieved and congestion condition in the network. If the event-to-sink reliability is lower than required, ESRT adjusts the reporting frequency of source nodes aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, then ESRT reduces the reporting frequency conservatively in order to conserve energy while still maintaining reliability. This self configuring nature of ESRT makes it robust to random, dynamic topology in WSN.

2.2 RMST: Reliable Data Transport in Sensor Networks

Here analysis and experiments resulting in specific recommendations for implementing reliable data transport in sensor nets. To explore reliability at the transport layer, they present RMST (Reliable Multi Segment Transport), a new transport layer for Directed Diffusion. RMST provides guaranteed delivery and fragmentation/reassembly for applications that require them. RMST is a selective NACK-based protocol that can be configured for in-network caching and repair.

Sensor networks have different constraints than traditional wired nets. First, energy constraints are paramount in sensor networks since nodes can often not be recharged, so any wasted energy shortens their useful lifetime. Second, these energy constraints, plus relatively low wireless bandwidths, make in-network processing both feasible and desirable. Third, because nodes in sensor networks are usually collaborating towards a common task, rather than representing independent users, optimization of the shared network focuses on throughput rather than fairness. Finally, because sensor networks are often deployed by a single organization with inexpensive hardware, there is less need for interoperability with existing standards. The main contribution is an evaluation of the placement of reliability for data transport at different levels of the protocol stack. To consider implementing reliability in the MAC, transport layer, application, and combinations of these. MAC-level reliability is important not just to provide hop-by-hop error recovery for the transport layer, but also because it is needed for route discovery and maintenance.

2.3 Networking issues in wireless sensor networks

Challenges in sensor networks are diverse; the key networking challenges in sensor networks that discussed are:

- (a) Supporting multi-hop communication while limiting radio operation to conserve power.
- (b) Data management, including frameworks that support attribute-based data naming, routing and in network aggregation.
- (c) Geographic routing challenges in networks where nodes know their locations.
- (d) Monitoring and maintenance of such dynamic, resource-limited systems.

For each of these research areas, to provide an overview of proposed solutions to the problem and discuss in detail one or few representative solutions. Finally, they illustrate how these networking components can be integrated into a complex data storage solution for sensor networks. Foremost among these is the development of long-lived sensor networks in spite of energy-constraints of individual nodes. Sensor nodes are

expected to be battery equipped, and deployed in a variety of terrains. In some of these deployments, it may be feasible to harness energy from ambient sources, such as solar power, whereas in others such as climate monitoring in the canopies, sensor nodes may not be able to renew their energy resources. A major energy consumer is radio communication.

2.4 Efficiency Centric Communication Model for Wireless Sensor Networks

Recent studies on radio reality provided strong evidence that radio links between low-power sensor devices are extremely unreliable. To identify an optimal bound on energy efficiency for reliable communication, and propose a new communication model in the link layer that asymptotically approaches this bound. This new model indicates a better path metric compared to previous path metrics, and to validate this by establishing a routing infrastructure based on this metric, which indeed achieves a higher energy efficiency compared to other state of the-art approaches. To model this traffic, a new parameter is introduced, Energy per Bit to characterize the energy efficiency aspect of communication. *EPB* represents the average energy consumption for each *delivered* bit from the source to the destination. *EPB* is decided by several factors, such as the link layer packet recovery mechanism, the routing layer path selection, the relative positions of the source and the destination, and the network topology, among others. Some of these factors, such as the relative positioning of the source and the destination, are unique to a particular transport task. Therefore, do not aim to optimize *EPB* across different transport tasks.

Present two corresponding techniques:

- 1) The *lazy* packet loss detection in the link layer
- 2) The use of a stream based path metric in the routing layer. The first optimization technique applies to a particular chosen path, while the second one applies to the path selection process. These two optimization techniques are unified by their consistency.

2.5 A High Throughput Path Metric for Multi Hop Wireless Routing

This paper presents the *expected transmission count* metric (ETX), which finds high-throughput paths on multi-hop wireless networks. ETX minimizes the expected total number of packet transmissions (including retransmissions) required to successfully deliver a packet to the ultimate destination. The ETX metric incorporates the effects of link loss ratios, asymmetry in the loss ratios between the two directions of each link, and interference among the successive links of a path. In contrast, the minimum hop-count metric chooses arbitrarily among the different paths of the same minimum length, regardless of the often large differences in throughput among those paths, and ignoring the possibility that a longer path might offer higher throughput. This paper describes the design and implementation of ETX as a metric for the DSDV and DSR routing protocols, as well as modifications to DSDV and DSR which allow them to use ETX. Measurements taken from a 29-node 802.11b test-bed demonstrate the poor performance of minimum hop count, illustrate the causes of that poor performance, and confirm that ETX improves performance.

One approach to fixing this problem is to mask transmission errors.

Another approach might be to augment minimum Hop-count routing with a threshold that ignores lossy links

2.6 The Design and Evaluation of a Mobile Sensor/Actuator Network for Autonomous Animal Control

A mobile, wireless sensor/actuator network application for use in the cattle breeding industry is investigated. The goal is to prevent fighting between bulls in on-farm breeding paddocks by autonomously applying appropriate stimuli when one bull approaches another bull. This is an important application because fighting between high-value animals such as bulls during breeding seasons causes significant financial loss to producers. Furthermore, there are significant challenges in this type of application because it requires dynamic animal state estimation, real-time actuation and efficient mobile wireless transmissions. An animal state estimation algorithm based on a state-machine mechanism for each animal is designed and implemented. Autonomous actuation is performed based on the estimated states of an animal relative to other animals. A simple, yet effective, wireless communication model has been proposed and

implemented to achieve high delivery rates in mobile environments. To evaluate the performance of our design by both simulations and field experiments

2.7 Shrinking-Horizon Dynamic Programming

To describe a heuristic control policy, for a general finite-horizon stochastic control problem, that can be used when the current process disturbance is not conditionally independent of previous disturbances, given the current state. This approximates the distribution of future disturbances by a product distribution with the same marginal. Then carry out dynamic programming, using this modified future disturbance distribution, to find an optimal policy, and in particular. At the next step this update the conditional distribution, and repeat the process, this time with a horizon reduced by one step. the method can be thought of as an extension of model predictive control.

2.7.1 Two Variations on a Revenue Management Problem

Approximate dynamic programming (ADP) methods, which are meant to handle unwieldy large state spaces, can be applied to find a suboptimal policy, using the augmented system. These methods are based on using an estimate of the optimal value function, or optimal policy. Another general method for finding a suboptimal control policy is model predictive control (MPC) which goes by many other names, including certainty-equivalent model predictive control (CE-MPC), dynamic matrix control rolling horizon planning and dynamic linear programming (DLP) In MPC, the action or control is found as follows. At each step, to solve a deterministic optimal control problem, with the unknown future disturbances replaced with some kind of estimates available at the current time

2.8 Energy-optimal and Reliable Data Collection via Collision-free Scheduling in Wireless Sensor Networks

The problem of supervision or periodic data collection for stationary wireless sensor networks and present a practical, energy-efficient, and reliable solution.

2.8.1 Energy-efficiency is achieved by combining various design methods

(1) Adopting network flow optimization techniques, the optimal scheme for balancing the communication load among all the nodes in the network is calculated. This gives the lower bound for the energy required for data collection process.

(2) Instead of using a fixed network topology (communication tree), a set of optimized trees is constructed and the communication tree varies Over different data collection cycles. To show that this method achieves an average energy consumption rate very close to the optimal value.

(3) The packet exchange procedure is designed asked on collision-free schedules, to minimize the number of ackets and the transmission and reception times for each node. Reliability of the process is guaranteed by including many retransmissions opportunities in the schedules.

2.9 Energy-Efficient Communication Protocol for Wireless Micro sensor Networks

In this paper, the communication protocols, which can have significant impact on the overall energy dissipation of these networks is being explained. LEACH (Low-Energy Adaptive Clustering Hierarchy), a clustering-based protocol that utilizes randomized rotation of local cluster base stations (cluster-heads) to evenly distribute the energy load among the sensors in the network. LEACH uses localized coordination to enable scalability and robustness for dynamic networks, and incorporates data fusion into the routing protocol to reduce the amount of information that must be transmitted to the base station. Simulations show that LEACH can achieve as much as a factor of reduction in energy dissipation compared with conventional routing protocols. In addition, LEACH is able to distribute energy dissipation evenly throughout the sensors and improves the lifetime of the network.

CHAPTER 3

PROBLEM DEFINITION

The wireless sensor networks have recently emerged as an important platform. In reliable data delivery paradigm, hop-by-hop ARQ and end-to-end feedback mechanisms are used comprehensively to guarantee packet delivery. Conventional reliability schemes generally do not take energy efficiency into consideration. However in wireless sensor networks (WSN), the greatest challenge is the development of long-lived sensor networks in spite of energy-constraints of individual nodes. Energy conservation is one of the primary requirements in WSN. Strict data delivery guarantee for individual packets is not required in many WSN applications. Field deployment of sensor network applications, such as environmental monitoring and animal tracking, the occasional loss of sensor readings is tolerable as long as the collective information from all the source nodes can be obtained.

Typical WSN applications involving the reliable detection and/or estimation of event features based on the collective reports of several sensor nodes observing the event. At the end of each decision interval, the sink makes an informed decision based on reports received from sensor nodes during that interval. The specifics of such a decision making process are application dependent. The sink derives a reliability indicator at the end of decision interval. Hence, notions of throughput, which are based on the number of source packets set out, are inappropriate. The transport problem in WSN is to configure the reporting rate of source nodes so as to achieve the required event detection reliability at the sink with minimum resource utilization.

CHAPTER 4

PROPOSED METHODOLOGY

4.1 ENERGY EFFICIENCY ANALYSIS

In this section we derive the energy efficiency associated with some feasible reliability schemes in the context of data collection in WSN. The energy consumption is defined as the expected number of packets transmitted for each successfully delivered data from a source to the sink. This notion of energy efficiency measure has been used in [6] and [7] as path costs for energy efficient routing protocols.

4.1.1 Energy Efficiency of ESRT

We first derive the energy efficiency of ESRT where no acknowledgement or retransmission is required. Over a path of h hops, with p as the loss probability on each hop, the expected number of packets required at the source is

$$n_{ESRT} = \frac{1}{(1-p)^h}$$

The total expected number of transmissions (energy consumption) required to deliver one successful packet from source to sink along the h hop path is:

$$N_{ESRT} = \sum_{i=1}^h \frac{1}{(1-p)^i} = \frac{1}{p} \times \left[\left(\frac{-1}{1-p} \right)^h - 1 \right]$$

4.1.2 Energy Efficiency of ARQ

The energy efficiency of ARQ is derived. ARQ uses acknowledgments and timeouts to achieve reliable data transmission. To complete a transfer of one data packet in one, hop, both data transmission and ACK has to be successful.

The Expected number of transmissions required to complete one data packet delivery in one hop is derived as follows:

$$n_{ARQ} = \frac{1}{(1-p)^2} + \frac{1}{1-p} = \frac{2-p}{(1-p)^2}$$

The first term is the number of data packets transmitted (including retransmissions), and the second term is the number of acknowledgements transmitted. The total expected number of transmissions (energy consumption) required to complete one data packet delivery from source to sink over h hops is:

$$N_{ARQ} = \frac{2-p}{(1-p)^2} \times h$$

4.2 EFFICIENT RELIABLE DATA COLLECTION

In this section, we calculate the number of retransmission needed to identify the safe region. The safe region is the one where we transmit the data without any retransmission between source and sink.

For a link with loss probability p , each time a retransmission is made, the overall loss probability is reduced (pulled down) by a certain amount. The loss probability after r -th retransmission is:

$$p_r = 1 - (1-p) \sum_{i=0}^r p^i = p^{r+1}$$

In sometimes, the number of retransmissions should be limited to 1, and release the channel right after a single retransmission. In most cases a small number of retransmission can reduce the overall loss probability to acceptable level for an application. Hence, it may not be worth the effort of making extra retransmissions to “squeeze the last drop”.

We use the dynamic programming [13] concept to find an optimal solution to this problem. In dynamic programming, before a node forwards a packet, it needs to make the

decision based on the states of the rest of the path, i.e. the number of residual hops in the path, and the next hop link loss probability.

In reality, the number of residual hops is not always available to a source or relaying node and the number may be time varying. Fortunately, the number of hops that a packet has gone through is easily obtainable. By using this number, a symmetric solution to the original dynamic programming solution can be obtained. The benefit of this scheme is that the number of hops a packet has passed can be easily measured with a simple counter. In fact some networking layer protocols, e.g. IP, have this information readily available as TTL (Time to Live) field without any additional cost.

The distributed Efficient Reliable Data Collection (eRDC) algorithm is presented in below:

The eRDC algorithm works as follows: before forwarding a data packet, the relaying node will request the number of hops that the packet has gone through (hop_count) from network layer.

4.3 eRDC algorithm:

```
p_target [] = {1.0, 1.0,.63,.42,.32,...};
hop_count = hopcount (packet);
packet->ack = 1; // ARQ is ON
p_nexthop = link_estimation(nexthop);
p_residue = p_nexthop;
While (p_residue > p_target [hop_count])
{
p_residue *= p_residue;
Send (nexthop, packet);
If receive (ACK) return;
Timer (retransmission timeout);
}
Packet->ack = 0; // ARQ is OFF
Send (nexthop, packet);
```

This information is used to lookup a target loss rate ($p_{\text{target}}[\text{hop_count}]$) from the table. The next hop link loss estimation (p_{nexthop}) is extracted from networking layer routing protocol such as MintRoute [7]. If the next hop link loss is larger than the target loss, ACK flag is set to enable normal ARQ process. Each retransmission, successful or not, is expected to reduce the residual loss rate (p_{residue}) according to equation (4). This retransmission process goes on until either an acknowledgement is received, or when the residual loss probability (p_{residue}) is below the target loss rate ($p_{\text{target}}[\text{hop_count}]$). The last retransmitted packet is sent with its ACK field cleared based on the assumption that the current number of retransmissions is sufficient to achieve the desirable loss probability. Any further retransmission is considered wasteful and unnecessary. The eRDC algorithm uses a dynamically controlled maximum number of retransmission based upon real-time measurement of network conditions rather than a fixed maximum number of retransmission. This ensures that the channel is released as early as possible. Any residual loss probability will be left for the source / sink to handle via ESRT mechanism. By taking advantages of both the ESRT and the ARQ under their respective favorable conditions, eRDC reduces the total number of transmissions, and thus saves energy.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 The simulation on ns-2

The MAMPR and AODV protocols are simulated in NS-2 network simulator, a discrete event simulator that can model and simulate multi-hop wireless ad hoc networks. The operating system used is Red Hat Linux 9.0. The Distribution Coordination Function of the IEEE standard 802.11 for wireless LANs is used as the MAC layer. The radio model used is one similar to Lucent's WaveLAN with nominal bit-rate of 2 Mbps. The simulation is carried out with Constant Bit Rate traffic. Each sender sends data packets of 512 bytes long. The simulation used a send buffer of 64 data packets, containing the data packets waiting for a route. Packets sent by routing layer are queued at the interface queue till MAC layer can transmit them. The size of the interface queue used is 50 packets long. The other considerations made for the simulation environment and protocol settings are shown in table 5.1

5.2 Results and analysis using nam

NAM stands for Network Animator. This tool animates the network elements as described in the tcl script. A complete visualization is available to the user which depicts the networking concepts in the project. The animator takes the tcl file as the input and creates a nam and a trace file as outputs. NAM consists of tools for editing the network topology, navigation bar and a step size controller for time. With all these tools it is possible to vividly view how actually the project works with finer resolution in time. The tools for editing include zoomer and controls for interfacing. A status bar at the bottom of the animator indicates the current status of the network elements.

Network space	1500 * 300 meters
Simulation time	75 seconds
Number of nodes	14
Physical/MAC layer	IEEE 802.11 at 1 Mbps
Transmission range	250 meters
Interface queue	Queue/Drop Tail/Priqueue
IFQ length	50 packets
Node speed	Static configuration to 50 m/sec
Traffic type	CBR
Load	CBR : 25 to 150 kbps with connection varied between 4 to 15
Bandwidth	2e6
Routing protocol	AODV

Table 5.1 Simulation Environment

Node-movement and traffic-pattern files are to be created for simulations. To simulate pedestrian and stationary scenarios, set the max speed of each node at 1.0 meter/sec and 0 respectively, the pause time at 1.0 second and packet size at 512 bytes. The application traffic pattern consists of 20 CBR nodes running on UDP within 1500 m×300 m or 500 m×500 m area and 10 or 15 CBR source-destination connections were collection randomly. The traffic of each of CBR connections begins at randomly collection start time within 1 second from the beginning of the simulation and the simulation stops after 20 seconds. For each pedestrian or stationary type, randomly collection 10 node-movement and traffic-pattern scenarios.

NS-2 can simulate the following:

- 1. Topology:** Wired, wireless
- 2. Scheduling Algorithms:** RED, Drop Tail,
- 3. Transport Protocols:** TCP, UDP
- 4. Routing:** Static and dynamic routing
- 5. Application:** FTP, HTTP, Telnet, Traffic generators

5.3 Physical/MAC/Network layers Specifications

Channel type	Wireless Channel
Antenna type	Omni Antenna
Transmission range	250 meters
Radio Propagation model	Ricean/Rayleigh
MAC protocol	Mac 802_11
Queue type	Priority Queue
Max queue length	50
Ad hoc Routing protocol	AODV

- **Packet Delivery Ratio for 10 seconds**

From the extracted trace file, the number of packets that are send, received and Dropped packets are calculated when the simulation time is 10 seconds.

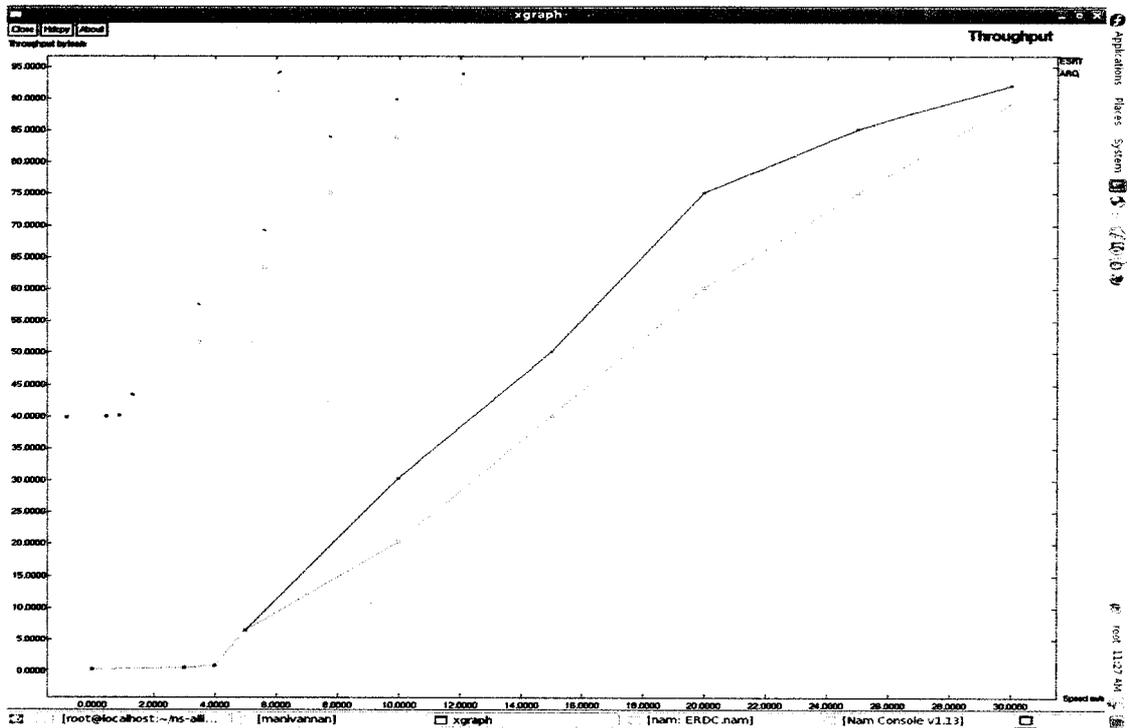


Figure 5.1 Comparisons in Throughput --ESRT and ARQ

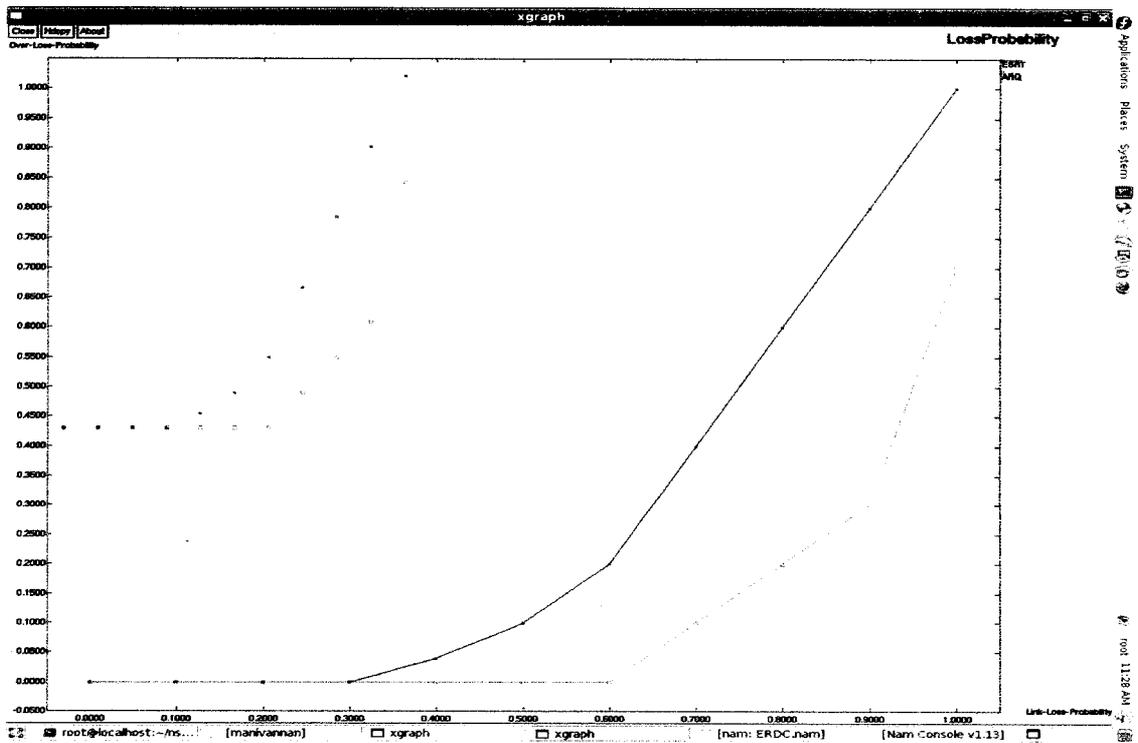


Figure 5.2 Comparisons in Loss Probability -- ESRT and ARQ

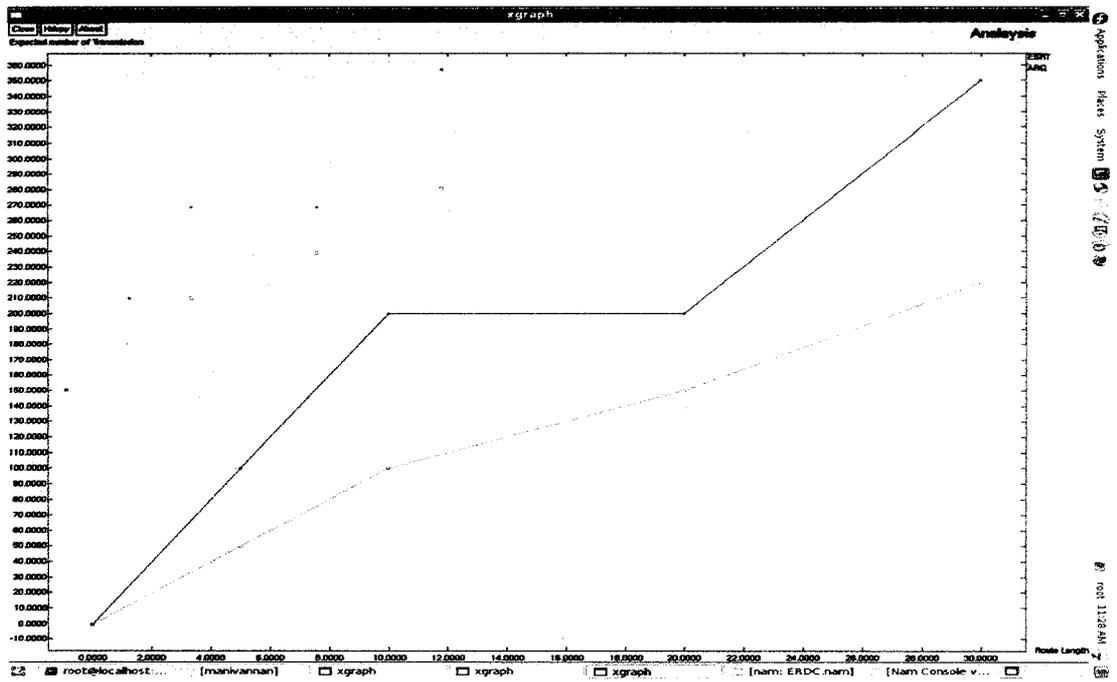


Figure 5.3 Analysis in no of retransmission reduced –ESRT and ARQ

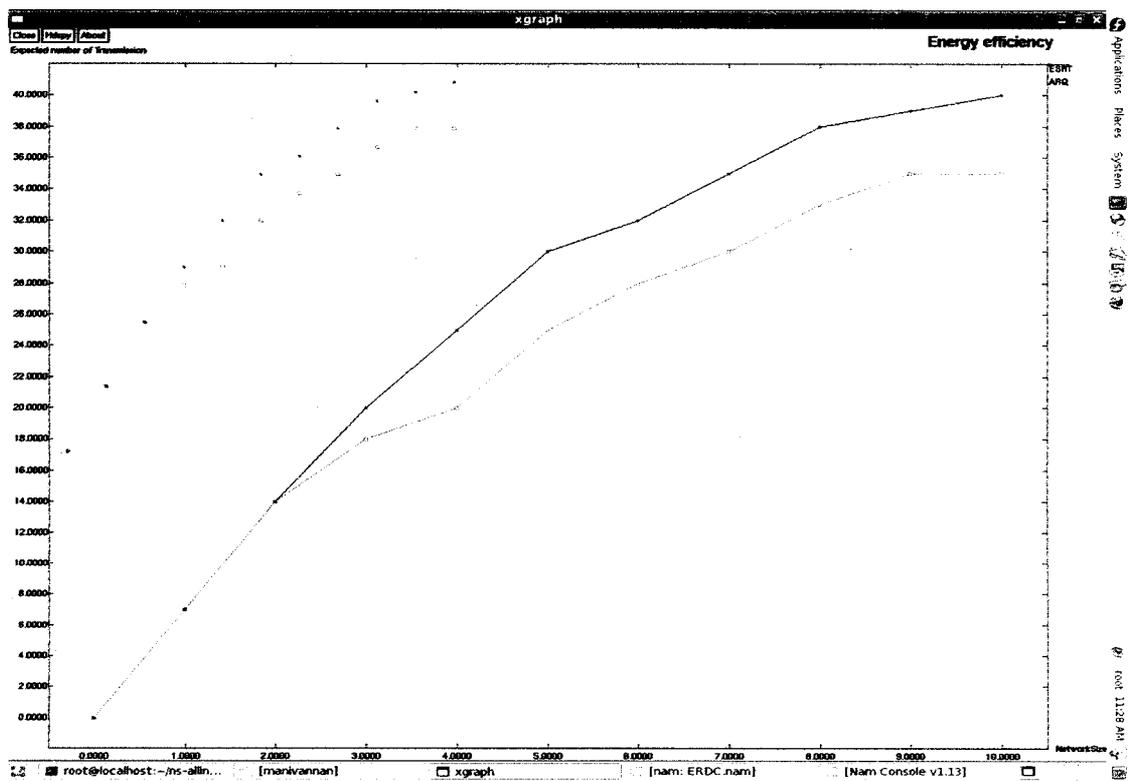


Figure 5.4 Energy Efficiency Analyses in -- ESRT and ARQ

CHAPTER 6

CONCLUSION AND FUTURE OUTLOOK

6.1 CONCLUSION

Sensor node energy is the balancing the tradeoff between reliability and energy consumption, the number of transmissions required to achieve certain degree of reliability is derived. This analysis provides a guideline for the optimal number of retransmissions at link layer. An Efficient Reliable Data Collection (eRDC) algorithm to dynamically control the maximum number of retransmissions based on these guidelines was proposed. Dynamic programming method is used to find the optimal solution. A distributed implementation of the eRDC algorithm was presented. Simulation results confirmed the findings.

6.2 FUTURE OUTLOOK

Currently a process of evaluating the distributed eRDC algorithm in large scale simulations and field trials over random topology networks used. Also developing a statistical bound on the coarse level of reliability for data collection are used.

CHAPTER 7

APPENDIX 1

Coding for eRDC Algorithms

/* IMPLEMENTATION OF DATA COLLECTION ALGORITHM */

7.1 SOURCECODE

=====

Define Node Configuration paramaters

=====

```
set val(chan)      Channel/WirelessChannel;      #Channel Type
set val(prop)      Propagation/TwoRayGround ;    # radio-propagation model
set val(netif)     Phy/WirelessPhy;             # network interface type
set val(mac)       Mac/802_11;                  # MAC type
set val(ifq)       Queue/DropTail/PriQueue;     # interface queue type
set val(ll)        LL;                           # link layer type
set val(ant)       Antenna/OmniAntenna;         # antenna model
set val(ifqlen)    50;                           # max packet in ifq
set val(nn)        15;                           # number of mobilenodes
set val(rp)        AODV;                         # routing protocol
set val(x)         1000
set val(y)         1000
set opt(energymodel) EnergyModel ;
set opt(radiomodel) RadioModel ;
```

```

set opt(initialenergy) 100;                                # Initial energy in Joules

Phy/WirelessPhy set bandwidth_ 2e6

Phy/WirelessPhy set Pt_ 0.2818;                          # for 250.0

Phy/WirelessPhy set freq_ 914e+6

Mac/802_11 set dataRate_ 2.0e6;                          # 1Mbps

# Initialize Global Variables

set ns_ [new Simulator]

=====

# Initialize trace file descriptors

set tracefd [open ERDC.tr w]

$ns_ trace-all $tracefd

set namtrace [open ERDC.nam w]

$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object

set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God

create-god $val(nn)

# Create channel #1 and #2

set chan_1_ [new $val(chan)]

set chan_2_ [new $val(chan)]

```

```

# Create node(0) "attached" to channel #1

# configure node, please note the change below

$ns_ node-config -adhocRouting $val(rp) \

    -llType $val(ll) \

    -macType $val(mac) \

    -ifqType $val(ifq) \

    -ifqLen $val(ifqlen) \

    -antType $val(ant) \

    -propType $val(prop) \

    -phyType $val(netif) \

    -topoInstance $topo \

    -agentTrace ON \

    -routerTrace ON \

    -macTrace ON \

    -movementTrace ON \

    -idlePower 1.0 \

    -rxPower 1.0 \

    -txPower 1.0 \

    -sleepPower 0.001 \

    -transitionPower 0.1 \

    -transitionTime 0.003 \

```

```

        -channel $chan_1_

# node_(1) can also be created with the same configuration, or with a different
# channel specified.

# Uncomment below two lines will create node_(1) with a different channel.

$ns_ node-config \           channel $chan_2_

set node_(0) [$ns_ node]

set node_(1) [$ns_ node]

set node_(2) [$ns_ node]

set node_(3) [$ns_ node]

set node_(4) [$ns_ node]

set node_(5) [$ns_ node]

set node_(6) [$ns_ node]

set node_(7) [$ns_ node]

set node_(8) [$ns_ node]

set node_(9) [$ns_ node]

set node_(10) [$ns_ node]

set node_(11) [$ns_ node]

set node_(12) [$ns_ node]

set node_(13) [$ns_ node]

set node_(14) [$ns_ node]

$node_(0) random-motion 0

```

```

$node_(1) random-motion 0
$node_(2) random-motion 0
$node_(3) random-motion 0
$node_(4) random-motion 0
$node_(5) random-motion 0
$node_(6) random-motion 0
$node_(7) random-motion 0
$node_(8) random-motion 0
$node_(9) random-motion 0
$node_(10) random-motion 0
$node_(11) random-motion 0
$node_(12) random-motion 0
$node_(13) random-motion 0
$node_(14) random-motion 0

for {set i 0} {$i < $val(nn)} {incr i}
{
    $ns_ initial_node_pos $node_($i) 20
}

# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
# define color index

$ns_ color 0 blue

$ns_ color 1 red

```

\$ns_color 2 chocolate

\$ns_color 3 red

\$ns_color 4 brown

\$ns_color 5 tan

\$ns_color 6 gold

\$ns_color 7 black

\$node_(0) set X_ 5.0

\$node_(0) set Y_ 5.0

\$node_(0) set Z_ 0.0

\$node_(1) set X_ 490.0

\$node_(1) set Y_ 285.0

\$node_(1) set Z_ 0.0

\$node_(2) set X_ 150.0

\$node_(2) set Y_ 240.0

\$node_(2) set Z_ 0.0

\$node_(3) set X_ 20.0

\$node_(3) set Y_ 80.0

\$node_(3) set Z_ 0.0

\$node_(4) set X_ 520.0

\$node_(4) set Y_ 330.0

\$node_(4) set Z_ 0.0

\$node_(5) set X_ 200.0

\$node_(5) set Y_ 400.0

\$node_(5) set Z_ 0.0

\$node_(6) set X_ 330.0

\$node_(6) set Y_ 580.0

\$node_(6) set Z_ 0.0

\$node_(7) set X_ 210.0

\$node_(7) set Y_ 100.0

\$node_(7) set Z_ 0.0

\$node_(8) set X_ 420.0

\$node_(8) set Y_ 270.0

\$node_(8) set Z_ 0.0

\$node_(9) set X_ 560.0

\$node_(9) set Y_ 165.0

\$node_(9) set Z_ 0.0

\$node_(10) set X_ 450.0

\$node_(10) set Y_ 75.0

\$node_(10) set Z_ 0.0

\$node_(11) set X_ 700.0

\$node_(11) set Y_ 280.0

\$node_(11) set Z_ 0.0

\$node_(12) set X_ 110.0
\$node_(12) set Y_ 700.0
\$node_(12) set Z_ 0.0
\$node_(13) set X_ 610.0
\$node_(13) set Y_ 175.0
\$node_(13) set Z_ 0.0
\$node_(14) set X_ 430.0
\$node_(14) set Y_ 120.0
\$node_(14) set Z_ 0.0
node_(1) color red
\$node_(2) color red
\$node_(3) color blue
\$node_(9) color blue
\$node_(4) color tan
\$node_(7) color tan
\$node_(11) color brown
\$node_(10) color brown
\$node_(12) color green
\$node_(14) color green
\$node_(13) color chocolate
\$node_(8) color chocolate

\$node_(5) color black

\$node_(0) color black

\$ns_ at 1.4 "\$node_(1) color red"

\$ns_ at 1.4 "\$node_(1) label S1"

\$ns_ at 1.4 "\$node_(2) color red"

\$ns_ at 1.4 "\$node_(2) label D1"

\$ns_ at 1.0 "\$node_(3) color blue"

\$ns_ at 1.0 "\$node_(3) label S2"

\$ns_ at 1.0 "\$node_(9) color blue"

\$ns_ at 1.0 "\$node_(9) label D2"

\$ns_ at 2.0 "\$node_(4) color tan"

\$ns_ at 2.0 "\$node_(4) label S3"

\$ns_ at 2.0 "\$node_(7) color tan"

\$ns_ at 2.0 "\$node_(7) label D3"

\$ns_ at 3.0 "\$node_(11) color brown"

\$ns_ at 3.0 "\$node_(11) label S4"

\$ns_ at 3.0 "\$node_(10) color brown"

\$ns_ at 3.0 "\$node_(10) label D4"

\$ns_ at 1.0 "\$node_(12) color green"

\$ns_ at 1.0 "\$node_(12) label S5"

\$ns_ at 1.0 "\$node_(14) color green"

\$ns_ at 1.0 "\$node_(14) label D5"
\$ns_ at 2.0 "\$node_(13) color chocolate"
\$ns_ at 2.0 "\$node_(13) label S6"
\$ns_ at 2.0 "\$node_(8) color chocolate"
\$ns_ at 2.0 "\$node_(8) label D6"
\$ns_ at 3.0 "\$node_(5) color black"
\$ns_ at 3.0 "\$node_(5) label S7"
\$ns_ at 3.0 "\$node_(0) color black"
\$ns_ at 3.0 "\$node_(0) label D7"
\$ns_ at 0.0 "\$node_(0) setdest 250.0 250.0 3.0"
\$ns_ at 3.0 "\$node_(1) setdest 45.0 285.0 5.0"
\$ns_ at 10.0 "\$node_(0) setdest 480.0 300.0 5.0"
\$ns_ at 8.0 "\$node_(3) setdest 420.0 180.0 3.0"
\$ns_ at 4.0 "\$node_(4) setdest 100.0 256.0 3.0"
\$ns_ at 9.0 "\$node_(5) setdest 120.0 355.0 5.0"
\$ns_ at 17.0 "\$node_(3) setdest 600.0 200.0 5.0"
\$ns_ at 2.0 "\$node_(6) setdest 700.0 400.0 3.0"
\$ns_ at 4.0 "\$node_(7) setdest 500.0 150.0 5.0"
\$ns_ at 6.0 "\$node_(6) setdest 580.0 120.0 5.0"
\$ns_ at 5.0 "\$node_(5) setdest 490.0 25.0 3.0"
\$ns_ at 3.0 "\$node_(2) setdest 310.0 300.0 5.0"

```

$ns_ at 2.0 "$node_(8) setdest 550.0 120.0 3.0"
$ns_ at 4.0 "$node_(9) setdest 620.0 240.0 5.0"
$ns_ at 7.0 "$node_(10) setdest 430.0 190.0 5.0"
$ns_ at 4.0 "$node_(11) setdest 760.0 410.0 5.0"
$ns_ at 4.0 "$node_(12) setdest 545.0 280.0 5.0"
$ns_ at 7.0 "$node_(13) setdest 324.0 330.0 5.0"
$ns_ at 4.0 "$node_(14) setdest 740.0 10.0 5.0"
$ns_ at 1.0 "$node_(12) setdest 230.0 90.0 5.0"
$ns_ at 2.0 "$node_(11) setdest 60.0 241.0 5.0"
$ns_ at 5.0 "$node_(10) setdest 545.0 280.0 5.0"
$ns_ at 8.0 "$node_(13) setdest 450.0 130.0 5.0"
$ns_ at 4.0 "$node_(14) setdest 360.0 118.0 5.0"

#DATA TRANSMISSION

# Setup traffic flow between nodes

set tcp [new Agent/TCP]

set sink [new Agent/TCPSink]

$ns_ attach-agent $node_(1) $tcp

$ns_ attach-agent $node_(2) $sink

$ns_ connect $tcp $sink

set ftp [new Application/FTP]

$ftp attach-agent $tcp

```

```
$ns_ at 15.000000000000000 "$ftp start"

set tcp1 [new Agent/TCP]

set sink1 [new Agent/TCPSink]

$ns_ attach-agent $node_(3) $tcp1

$ns_ attach-agent $node_(9) $sink1

$ns_ connect $tcp1 $sink1

set ftp1 [new Application/FTP]

$ftp1 attach-agent $tcp1

$ns_ at 16.0 "$ftp1 start"

set tcp2 [new Agent/TCP]

set sink2 [new Agent/TCPSink]

$ns_ attach-agent $node_(4) $tcp2

$ns_ attach-agent $node_(7) $sink2

$ns_ connect $tcp2 $sink2

set ftp2 [new Application/FTP]

$ftp2 attach-agent $tcp2

$ns_ at 4.0 "$ftp2 start"

set tcp3 [new Agent/TCP]

set sink3 [new Agent/TCPSink]

$ns_ attach-agent $node_(11) $tcp3

$ns_ attach-agent $node_(10) $sink3
```

```
$ns_ connect $tcp3 $sink3

set ftp3 [new Application/FTP]

$ftp3 attach-agent $tcp3

$ns_ at 1.0 "$ftp3 start"

set tcp4 [new Agent/TCP]

set sink4 [new Agent/TCPSink]

$ns_ attach-agent $node_(12) $tcp4

$ns_ attach-agent $node_(14) $sink4

$ns_ connect $tcp4 $sink4

set ftp4 [new Application/FTP]

$ftp4 attach-agent $tcp4

$ns_ at 5.0 "$ftp4 start"

set tcp5 [new Agent/TCP]

set sink5 [new Agent/TCPSink]

$ns_ attach-agent $node_(13) $tcp5

$ns_ attach-agent $node_(8) $sink5

$ns_ connect $tcp5 $sink5

set ftp5 [new Application/FTP]

$ftp5 attach-agent $tcp5

$ns_ at 10.0 "$ftp5 start"

set tcp6 [new Agent/TCP]
```

```
set sink6 [new Agent/TCPSink]

$ns_ attach-agent $node_(5) $tcp6

$ns_ attach-agent $node_(0) $sink6

$ns_ connect $tcp6 $sink6

set ftp6 [new Application/FTP]

$ftp6 attach-agent $tcp6

$ns_ at 10.0 "$ftp6 start"

set tcp7 [new Agent/TCP]

set sink7 [new Agent/TCPSink]

$ns_ attach-agent $node_(12) $tcp7

$ns_ attach-agent $node_(14) $sink7

$ns_ connect $tcp7 $sink7

set ftp7 [new Application/FTP]

$ftp7 attach-agent $tcp7

$ns_ at 8.0 "$ftp7 start"

set tcp8 [new Agent/TCP]

set sink8 [new Agent/TCPSink]

$ns_ attach-agent $node_(13) $tcp8

$ftp8 attach-agent $tcp8

$ns_ at 12.0 "$ftp8 start"
```

```

*****
Tell nodes when the simulation ends

for {set i 0} {$i < $val(nn)} {incr i} {

    $ns_ at 20.0 "$node_($i) reset";}

$ns_ at 75.0 "stop"

$ns_ at 75.01 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {

    global ns_ tracefd

    $ns_ flush-trace

    close $tracefd

    puts "running nam..."

    exec xgraph g1 &

    exec xgraph g2

    & exec xgraph g3

    & exec xgraph g4

    & exec nam ERDC &

}

puts "Starting Simulation..."

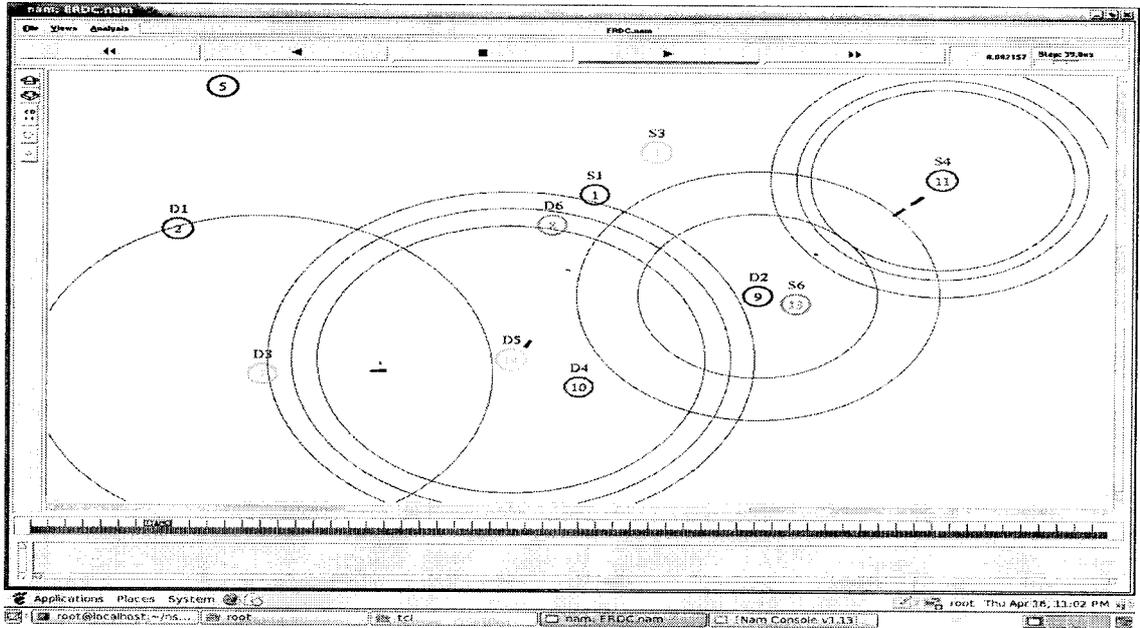
$ns_ run

```

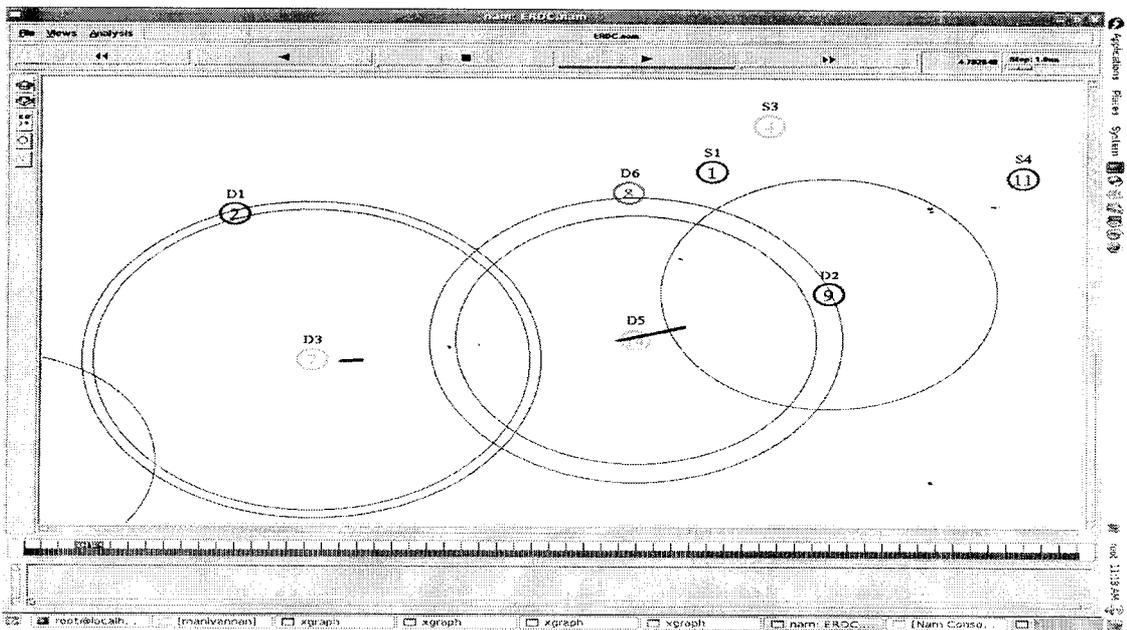
APPENDIX - 2

7.2 SCREEN SHOTS

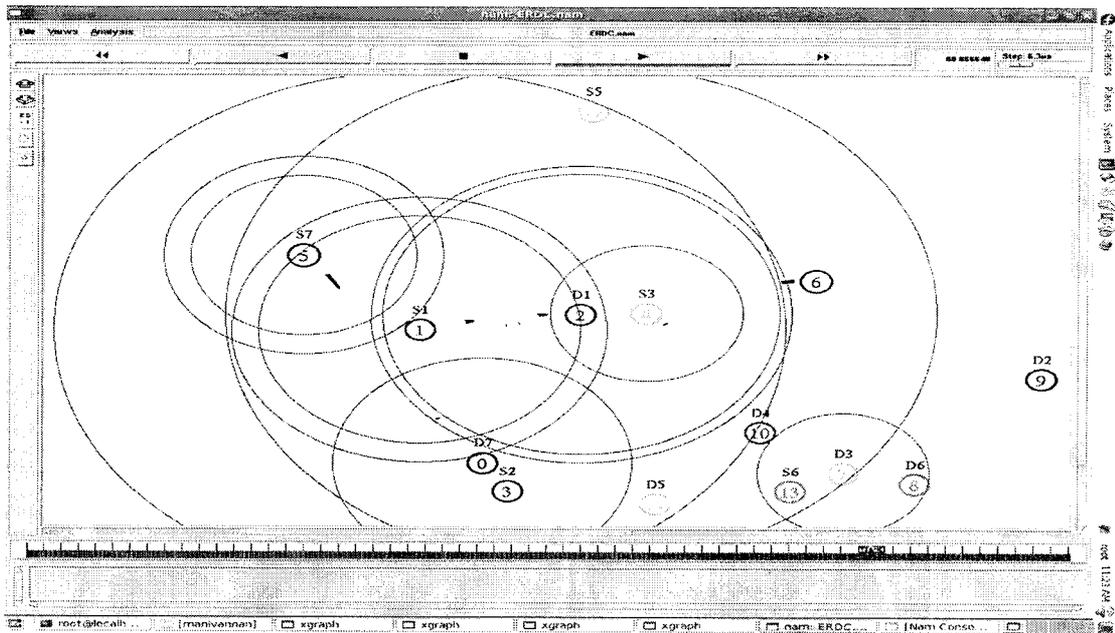
- Node Creation



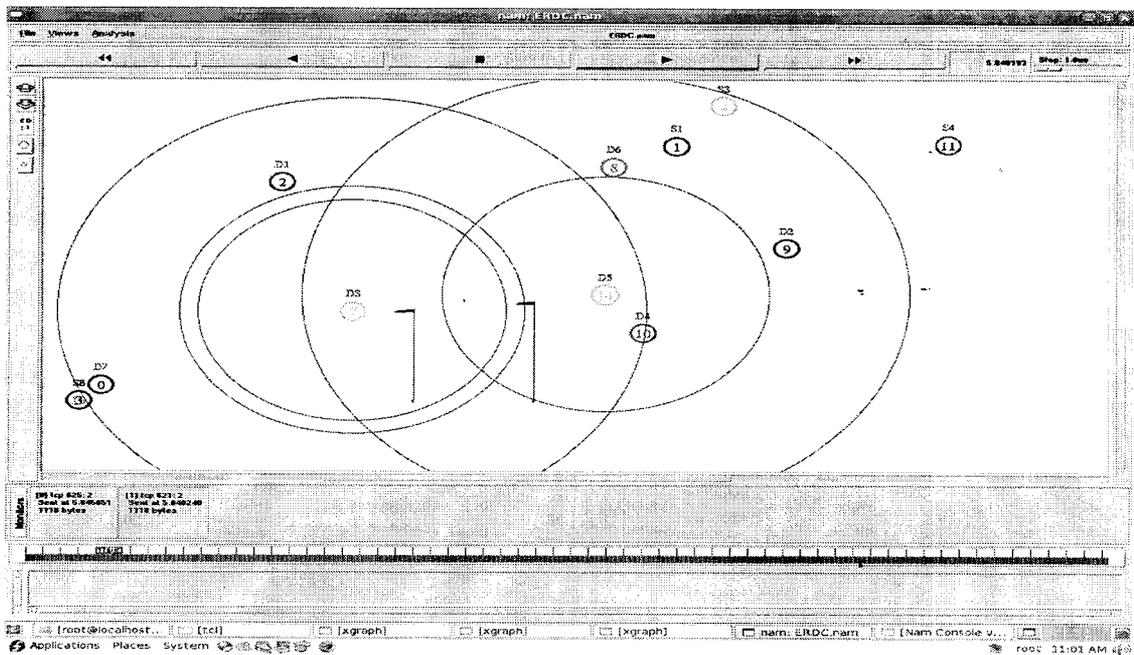
- Data Collection from source to sink



- o Acknowledgement Receiving at source



- o Transmission Monitoring System



CHAPTER 8

REFERENCES

- [1] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, D. Estrin, "Networking Issues in Wireless Sensor Networks," Elsevier Publishers, July 2004.
- [2] O. Akan, I. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks", 2005.
- [3] R. Stann, J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks", May 2003.
- [5] Q. Cao, T. He; L. Fang, T. Abdelzaher, J. Stankovic and S. Son, "Efficiency Centric Communication Model for Wireless Sensor Networks," in *Proceedings of INFOCOM '06*, Apr. 2006
- [6] D.S.J.De Couto, D.Aguayo, J.Bicket, R.Morris "A High-Throughput Path Metric for Multi-Hop Wireless Routing", In Proceedings of MOBICOM 2003.
- [7] A. Woo, T. Tong, and D. Culler. Taming "The Underlying Challenges of Reliable Multihop Routing in Sensor Networks", 2003.
- [8] T. Le Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, S. Brosnan, "Design And Deployment of a Remote Robust Sensor Network: Experiences from An Outdoor Water Quality Monitoring Network," 2007.
- [9] T. Wark, C. Crossman, W. Hu, C. Lee, J. Henshall, J. O. Grady, M. Reed, and A. Fisher, "The design And evaluation of a mobile sensor/actuator network for autonomous Animal control," 2007
- [11] ns-2, <http://www.isi.edu/nsnam/ns/>
- [12] R. Bellman, "Dynamic Programming", Courier Dover Publications, 2003.