Energy Efficient and Reliable Algorithms for Data Gathering in Wireless Sensor and Actor Networks

Submitted in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

Submitted by

Sai Krishna Mothku

(Roll No. 701431)

Under the guidance of

Dr. Rashmi Ranjan Rout

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

TELANGANA - 506004, INDIA

SEPTEMBER 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

THESIS APPROVAL FOR Ph.D.

This is to certify that the thesis entitled, Energy Efficient and Reliable Algorithms for Data Gathering in Wireless Sensor and Actor Networks, submitted by Mr. Sai Krishna Mothku [Roll No. 701431] is approved for the degree of DOCTOR OF PHILOSOPHY at National Institute of Technology Warangal.

Examiner

Research Supervisor	Chairman
Dr. Rashmi Ranjan Rout	**Prof. R.B.V. Subramanyam**
Dept. of Computer Science and Engg.	**Head, Dept. of Computer Science and Engg.**
NIT Warangal	**NIT Warangal**
India	**India**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

CERTIFICATE

This is to certify that the thesis entitled, Energy Efficient and Reliable Algorithms for Data Gathering in Wireless Sensor and Actor Networks, submitted in partial fulfillment of requirement for the award of degree of DOCTOR OF PHILOSOPHY to National Institute of Technology Warangal, is a bonafide research work done by Mr. Sai Krishna Mothku [Roll No. 701431] under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

Research Supervisor
Dr. Rashmi Ranjan Rout
Associate Professor
Dept. of Computer Science and Engg.
NIT Warangal
India

Place: NIT Warangal
Date:

DECLARATION

This is to certify that the work presented in the thesis entitled "*Energy Efficient and Reliable Algorithms for Data Gathering in Wireless Sensor and Actor Networks*" is a bonafide work done by me under the supervision of Dr. Rashmi Ranjan Rout and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sai Krishna Mothku

(Roll No. 701431)

Date:

ACKNOWLEDGMENTS

It is with great pleasure that I acknowledge my sincere thanks and deep sense of gratitude to my supervisor Dr. Rashmi Ranjan Rout, Department of Computer Science and Engineering for his valuable guidance throughout the course. His technical perception, profound knowledge, sincere effort in guiding a student and devotion to work have greatly charmed and constantly motivated me to work towards the goal. He always gave me ample time for discussions, reviewing my work and suggesting requisite corrections.

I extend my gratitude to all my Doctoral Scrutiny Committee members for their valuable suggestions and faculty members of Computer Science and Engineering Department. I wish to express my sincere thanks to Prof. D.V.L.N. Somayajulu, Prof. R.B.V. Subramanyam, Dr. S. Ravi Chandra, Prof. N.V.S Narasimha Sarma and Prof. L. Anjaneyulu for their insightful comments and suggestions during oral presentations. I am also grateful to Prof. T. Ramesh and Dr. Ch. Sudhakar, who has always been supportive and encouraging all through my tenure. I am lucky to attend lectures by Prof. B. B. Amberker during my tenure. I am immensely thankful to Prof. R. B. V. Subramanyam, Head of Dept. of CSE, for providing adequate facilities in the department to carry out the research work. I wish to express my sincere thanks to Prof. N.V. Ramana Rao, Director, NIT Warangal for his official support and encouragement.

I would also like to thank my friends Sudarshan, Sandipan Maiti, Manoj, Abhilash, Hemkumar, Preethi, Sumalatha, Vemireddy, Sanjib, Sunil Joy and Hemanth Nooka for their valuable suggestions and for extending selfless cooperation. My gratitude to my family for their unconditional love, support and prayers for my success in achieving the goal. Lastly I am deeply indebted to the almighty for giving the opportunity to pursue this line.

Sai Krishna Mothku

Dedicated to

My Family & Teachers

ABSTRACT

Wireless Sensor and Actor Networks (WSANs) recently emerged as major information-gathering paradigm due to its wide variety of applications, such as issuing tsunami alerts, chemical attack detection, forests fire detection and intrusion detection in military surveillance. The energy-constrained sensor nodes spend more energy in transmitting data packets than in sensing operation. The data transmission rate and delivery delay increase due to increase in packet dropping rate. Thus, improvement in energy efficiency is a challenging issue while providing reliable data delivery and stringent delivery delay in WSANs. Reduction of packet dropping rate results in improvement of energy efficiency, data transmission reliability and delivery delay. Packet dropping occurs mainly due to unavailability of free buffer and unreliable wireless links. Further, energy harvesting technology extends the lifetime of a sensor network. The volume of harvested energy varies dynamically with the change in weather conditions over time. This may lead to temporary disconnection of nodes from the network. Therefore, survivability of a node till the next recharge cycle improves the reliable data transmission rate in energy harvesting sensor actor networks.

The thesis focuses on the energy efficient and reliable algorithms for data gathering in wireless sensor and actor networks. The issues of energy efficiency and reliable data transmission have been addressed while maintaining stringent data delivery delay. In this thesis, the proposed approaches have achieved reliable data transmission by reducing the packet dropping rate. Firstly, a markov decision process based buffer management mechanism has been designed in tree-based WSAN to reduce the data delivery delay and to improve the energy efficiency. A mathematical model for a mobile actor is presented to analyze buffer occupancy and energy consumption with event-centric traffic. Secondly, a reliable data transmission mechanism using opportunistic encoding has been proposed for a WSAN with faulty nodes. The proposed mechanism analyzes the quality of link states and determines the applicability of network coding to improve the data transmission reliability and to reduce the number of data transmissions. Thirdly, a fuzzy based delay and energy-aware intelligent routing mechanism has been developed to take effective routing decisions. The proposed mechanism computes a routing metric by considering the residual energy, link quality, available buffer and distance. Finally, a fuzzy based adaptive duty cycling algorithm has been designed to achieve network sustainability in harvesting sensor actor networks. Energy consumptions of bottleneck zone has been estimated with the proposed network model. The proposed mechanism takes switching decision for network survivability. Coordinated duty cycle schedule is considered to improve the reliable data delivery. Performances of proposed approaches have been evaluated through simulation.

Keywords: Wireless sensor and actor networks, energy efficiency, delay, reliable data transmission, buffer management, markov decision process, network coding, energy and delay aware routing, energy harvesting nodes, network sustainability, duty cycle, fuzzy logic.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

WSN	Wireless Sensor Networks
WSAN	Wireless Sensor and Actor Networks
MDP	Markov Decision Process
FLS	Fuzzy Logic System
COA	Center of Area
NS-3	Network Simulator-3
FEARM	Fuzzy based Energy-Aware Routing Mechanism
AR	Automatic Retransmission
PRD	Predicted Remaining Deliveries
DCR	Data Collection Region
DR	Determined Rule Set
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance

Chapter 1

Introduction

Recent advances in sensing technologies lead to the emergence of heterogeneous sensor networks including wireless sensor and actor networks (WSANs) [1]. A WSAN is a type of wireless sensor network (WSN) where one or more number of actors gather (i.e. collect) data within the sensor deployment region. A group of sensors and actors are geographically distributed and interconnected by wireless links in a WSAN. They may be deployed randomly or deterministically in a monitoring field where persistent human surveillance is infeasible. Sensors usually communicate with each other using multi-hop paths [2, 3]. In WSANs, sensors gather information about the physical world and transmit the collected data to actors. Actors forward data towards a sink node. From the received information, the actors perform the actions based on the data collected from the physical world. Usually sensor nodes are static, whereas actors may move in the monitoring area for data gathering. Wireless sensor and actor networks (WSANs) and wireless sensor networks (WSNs) share many common considerations concerning network design such as reliability, scalability, connectivity and energy efficiency. The existence of actors along with sensors in WSANs causes significant difference between the two types of networks. In a sensor network, actions may be performed for local decision making for the purpose of enhancing the monitoring and decision making capability in the networks.

Achieving energy efficient and reliable data collection with stringent delay in wireless sensor and actor networks is a challenging task in applications like monitoring the health of chemical plants, nuclear plants and bridges, forest fire detection, intrusion detection in

military surveillance and issuing tsunami alerts. A packet may be dropped due to unreliable link quality, overflow of buffer and low energy levels of a node. Retransmission of dropped packets resolves the issue of reliable data collection in WSANs. However, it increases energy consumption and delivery delay in the network. The above mentioned observations motivate the present work for designing energy efficient and reliable data collection algorithms in WSANs while maintaining stringent data delivery delay. Further, energy harvesting technology extends the lifetime of a WSAN. The energy harvesting sensor nodes harvest energy from various sources, such as sunlight, wind, vibration and thermal. Survivability of a node till next recharging cycle achieves sustainability and reliable data collection in energy harvesting sensor actor networks.
The contributions in this thesis are as follows:

- **An adaptive buffer management mechanism using markov decision process for efficient data gathering in a WSAN** : This work presents a buffer management mechanism to reduce the data delivery delay and to improve the energy efficiency in tree-based wireless sensor and actor networks. In this work, analysis on energy consumptions has been performed with a typical data acquisition model. A mathematical model for an actor is presented to estimate bound on buffer occupancy and energy consumption.

- **A reliable data gathering mechanism using opportunistic encoding in a wireless sensor and actor network**: This work presents an opportunistic encoding mechanism to achieve reliable data gathering (i.e. data collection) in the presence of faulty nodes. In the proposed mechanism, network coding approach encodes packets to reduce the number of transmissions. A markov decision process algorithm is designed for opportunistic network coding decisions. The proposed mechanism improves packet level reliability with stringent delivery delay and energy efficiency.

- **A fuzzy based energy and delay aware routing protocol for a heterogeneous sensor actor network**: This work presents an energy and delay aware routing protocol using fuzzy logic in heterogeneous sensor actor network. This work considered the heterogeneous characteristics, such as battery capacity, link quality and buffer capacity. The proposed protocol computes a routing metric by considering the residual energy, available buffer, quality of link and distance (proximity). The proposed protocol reduces the packet dropping rate and this leads to the reduction in delay and energy consumption.

- **An adaptive duty cycling algorithm for sustainability in energy harvesting sensor actor networks**: This work presents a fuzzy based duty cycling algorithm to achieve sustainability in a tree-based harvesting sensor actor network. Fuzzy logic controller adaptively controls the duty cycle of a sensor node by considering residual energy, predicted harvesting energy and expected residual energy. The proposed mechanism includes a switching decision for survivability.

The rest of this chapter is organized as follows. Motivation behind the work has been presented in Section 1.1. In Section 1.2.1, the requirement of designing an adaptive buffer management mechanism in tree-based wireless sensor actor network has been highlighted. Upper bound on buffer occupancy and energy consumption is also estimated in this section. In Section 1.2.2, the requirement of opportunistic encoding for wireless sensor and actor networks in the presence of faulty regions has been stressed. A learning automata based routing mechanism is also explained in this section. In Section 1.2.3, a discussion on energy and delay aware routing protocol using fuzzy logic is mentioned. Section 1.2.4 describes the usage of adaptive duty cycling algorithm for sustainability in tree-based energy harvesting sensor actor network. Section 1.3 gives experimental setup details. The organization of the thesis has been presented in Section 1.4.

1.1 Motivation and objectives

Sensor nodes are usually small and low cost battery powered devices and having limited sensing, data processing and wireless communication capabilities while actor nodes as compared with sensors have higher computation and communication capabilities and longer battery life [2]. WSAN can be considered as a heterogeneous network by considering capabilities of nodes, link characteristics and node mobility.

WSANs may consist of hundreds or thousands of low-cost sensors which are capable of sensing and collecting data from the environment and push the relevant data to actors. The actors react to the information by performing appropriate actions. Data packets need to traverse many hops to reach the actor. However, it is inefficient for every sensor node to relay sensed data because sensor nodes are often constrained with limited resources in memory, computation, communication, and battery. One of the most important constraints for sensor nodes is the low power consumption because the sensor nodes are typically battery operated devices. Commonly, it is not feasible to replace exhausted batteries once the network is deployed. Therefore, a primary concern in a WSAN is the issue of energy efficiency and network lifetime maximization while providing reliability and stringent delay in data delivery. Recently, energy harvesting sensor nodes have been designed to harvest energy from various natural resources, such as solar light, thermal, wind, vibration and water flow [4]. However, optimal use of this energy till next recharge time is an important issue in a rechargeable wireless sensor actor network.

In a typical sensor network (multi-hop), nodes in immediate vicinity of actor or sink will be overburdened and the energy of the nodes will be depleted quickly when compared with other nodes in the network. This leads to network partitioning and actor cannot receive data from the sensor nodes [5]. Therefore, improvement in energy efficiency can be achieved using mobile actor where actor moves across the network to collect data from sensors. Resource constrained nodes store the packets till a mobile actor collects the data. Therefore, packet dropping occurs when buffers of the nodes are full. The dropped packets are retransmitted by the source node. This leads to more energy consumption and delay in data delivery. Secondly, network coding and duty cycle techniques increase the network lifetime and reliable data delivery in homogeneous networks. However, heterogeneous networks (such as WSAN) need further study. Although these solutions are effective in some scenarios, network lifetime is still determined by the limited battery energy. Further, energy harvesting based WSAN need to be investigated by using energy efficient techniques such as duty cycle and network coding. Thus, the present work focuses on the above mentioned observations to increase the energy efficiency while providing the data reliability and stringent data delivery delay in the heterogeneous networks by considering Random and Tree-based topologies. The above mentioned challenges motivate the present work towards energy efficient and reliable data gathering algorithms in wireless sensor and actor networks. The major objectives of this dissertation are as follows.

1. Analysis of energy efficiency and delay using Markov Decision Process with mobile actor and with event-centric load in a Tree-based sensor and actor network.

2. Design of a network coding based energy efficient communication approach for a wireless sensor and actor network in the presence of faulty nodes.

3. Design of a fuzzy logic based energy and delay aware routing protocol for a heterogeneous sensor actor network.

4. Analyze the energy efficiency and sustainability of a wireless sensor and actor network with energy harvesting nodes.

1.2 Overview of the Contributions of this Thesis

In this section, an overview of chapter-wise contributions of this thesis has been presented. Each subsection presents summary of contributions of the corresponding chapter.

1.2.1 Markov Decision Process Based Adaptive Buffer Management Mechanism

In this work, a tree-based wireless sensor network is considered with a mobile actor. Sensor trees are disconnected and mobile actor collects data. Communication range of root node in a sensor tree is defined as *Data Collection Region*. Data acquisition is done through the mobile actor. Mobile actor collects the data from all data collection regions (from root node of every sensor tree) by traveling around the network. Improvement in energy efficiency is achieved using mobile actor where actor moves across the network to collect data from sensor nodes [6]. Resource constrained nodes need to store the packets till a mobile actor collects the data. Therefore, packet dropping occurs when buffers of nodes are full. The dropped packets are retransmitted by the source node which lead to more energy consumption and delay in data delivery. Delivery delay reduces by reducing the number of retransmissions. Nodes in the network may get high traffic in a given time period. It would be beneficial to drop the packets (which are generated from nearest source node) to improve available buffer space. In this work, an adaptive buffer management technique using *Markov Decision Process* (MDP) is proposed to determine the buffer states at which the packets will be dropped to reduce average delivery delay and energy consumption in the network. Different MDP policies are formulated to achieve lower delivery delay. The focus of this work is to estimate the energy utilization in tree-based WSAN. Further, the work focuses on formulating a mathematical model for mobile actor and analyzing the upper bound on buffer occupancy and energy consumption. The major contributions of this work are as follows:

- Analysis of energy consumptions in tree-based wireless sensor network while considering an event-centric network traffic.
- Analysis of bounds on buffer occupancy and energy consumption in tree-based sensor actor networks.
- Design of a *Markov Decision Process based buffer management mechanism* to reduce the end-to-end delay and to improve the energy efficiency in a tree-based WSAN.

Analysis of Energy Consumption and Network Delay in a Tree-based Sensor Network with a Mobile Actor

In this section, an energy consumption has been estimated for the proposed network model by adopting the model described in [7] and [8]. Buffer of each node is modeled using $M/M/1/b$ queuing model to analyze the end-to-end delay of a packet in a sensor tree. Further, energy consumption and delay for retransmission of dropped packets have been estimated.

Upper Bound on Buffer Occupancy and Energy Consumption

In a Wireless sensor and actor network, evaluation of the traveling schedule of mobile actor is an important issue to reduce the dropping rate of packets. Further, buffer state (buffer occupancy) of a node changes dynamically in event-centric network. The buffer states of tiny sensors are limited due to which some of the packets are dropped. The retransmission of the dropped packets leads to more delay and more energy consumption in the network. Therefore, there is a need to estimate the bounds on the buffer occupancy and energy consumption in a tree-based WSAN. In this section, a mathematical model is formulated for an actor to estimate upper bound on buffer occupancy and energy consumption in the proposed network model.

MDP based Buffer Management Algorithm

In the proposed network model, when the actor enters into the data collection region of a sensor tree, every node transmits the buffered data to its parent node. Further, the actor collects data from the root node during its travel through data collection region of the sensor tree. Nodes in the network may get high traffic at a particular time. The packets are dropped if sufficient buffer space is not available to store the data. Then, the source node has to retransmit (the dropped packets) which leads to increase in average end-to-end delay and energy consumption (for retransmissions of dropped packets). Therefore, it will be beneficial to drop the packets (which are generated at nearest sensor nodes) to increase the available buffer space. However, the number of retransmissions depends on the buffer occupancy (i.e. at which state packets are to be dropped and number of packets to be dropped). Hence, we need to determine the buffer state at which packets need to be dropped so that average delivery delay and energy consumption for retransmissions of dropped packets are minimized. To address the above problems, a *Markov Decision Process* based buffer management is proposed in this section.

The proposed adaptive buffer management mechanism is simulated using *Network Simulator-3* [9]. The performance metrics such as average end-to-end delay, energy consumptions and lifetime are taken to compare proposed MDP based adaptive buffer management mechanism and geographic routing protocol [10]. Simulation results show that the average end-to-end delay reduces significantly using the proposed approach in comparison to a geographic routing protocol. Further, it is observed that there is a reduction of energy consumption upto 19.83% using the proposed MDP based buffer management mechanism in comparison to a geographic routing protocol.

1.2.2 Reliable Data Transmission Mechanism using Markov Decision Process and Network Coding

In this work, a wireless sensor and actor network (WSAN) with n number of sensor nodes is considered and that are deployed randomly in an area. A faulty region is defined as a region which is affected by adverse environmental effects (such as fog, rainfall, high temperature, etc.) for temporary period of time [11]. Nodes in a faulty region are called as faulty nodes. In a WSAN with faulty regions, achieving high degree of data reliability with appropriate end-to-end delay is an inescapable challenge. Data reliability depends on the quality of wireless channel. It will change dynamically with environmental conditions such as fog, rainfall, high temperature [11] and movable obstacles. Therefore, the quality of link varies dynamically in the presence of environmental interferences.

A packet may be lost due to fluctuation of a wireless link (quality) in the presence of adverse environmental conditions. However, high data transmission reliability can be achieved using acknowledgment based retransmission approaches [12, 13]. In a retransmission based approach, a source packet will be retransmitted till reception of an acknowledgment [12, 13]. Therefore, more number of data transmissions (including retransmissions) are required to achieve data transmission reliability. This leads to increase in the number of data transmissions, delivery delay and energy consumption. Therefore, a retransmission based approach is inefficient for delay-constrained and energy-constrained applications in WSAN. Achieving reliable data collection (packet level reliability) with stringent end-to-end delay requirement is a challenging issue in some of the applications (in WSAN with faulty regions) such as, issuing tsunami alerts, chemical attack detection, forests fire detection and intrusion detection in military surveillance [2, 3]. Hence, there is a requirement of an efficient approach for the delay constrained applications in WSAN. Network coding mechanism (is a mechanism by which raw packets are encoded and transmitted by sensor nodes) can be applied to achieve improved data collection reliability and reduction in packet delay when a link is unreliable [12, 13]. Further, the level of packet redundancy influence the number of data transmissions, there by, energy consumption. In order to improve data collection reliability along with energy efficiency, appropriate level of packet redundancy must be chosen adaptively based on the quality of the wireless links.

To address the above mentioned problems, a network coding based reliable data transmission mechanism has been proposed in our work for delay-sensitive applications. Further, a *markov decision process (MDP)* based algorithm has been designed by considering variable link quality in a wireless channel. The *markov decision process* is designed for opportunistic network coding decisions. The MDP algorithm decides: (a) whether to apply coding for relaying packets, and (b) decides the level of packet redundancy. Our algorithm determines the level of packet redundancy (adaptively) in the network coding process for reduction of data transmissions. Further, a *Learning Automata* based mechanism has been designed to route the traffic through alternate links, in case, the present link quality is bad. Major contributions of this work are as follows:

- Design of a network coding based reliable data transmission mechanism to improve the reliable data collection for a wireless sensor and actor network in the presence of faulty regions.
- Design of a learning automata based mechanism to route the traffic through alternate links, when the present link quality is bad.

A Network Coding based Data Transmission Mechanism

In this section, an overview of the proposed mechanism is presented. The proposed mechanism has three major functions, such as, *LINK_QUALITY()*, *DIVERT_TRAFFIC()* and *MDP_REDUNDANCY()*. The *LINK_QUALITY()* function finds the quality of links (the probability of successful transmission) of a node to all its neighbors. The *DIVERT_TRAFFIC()* function will be called by a node if the node has poor quality of link (less than the threshold value (p_{th}) which will be chosen based on application requirements) to the next node. Then, the node selects a new next node which has maximum selection probability as compared with its neighbors using Learning Automata (LA). *MDP_REDUNDANCY()* function decides whether to encode the packets or not and it finds the level of packet redundancy which will be applicable in encoding process (if the decision is *Do Network Coding*). Moreover, *MDP_REDUNDANCY()* returns the optimal policy in terms of number of data transmissions per data collection round (including retransmissions), data collection reliability, packet latency and energy efficiency. Deterministic coding vectors are applied in the coding process.

The proposed mechanism is compared with a redundancy-based mechanism (proposed by Wu *et al.* [12]), the end-to-end erasure coding (EEEC) [14] and automatic retransmission (AR) [12] in terms of average packet delivery delay, number of data transmissions, energy consumptions and lifetime. From the simulation results, it has been observed that the average delivery delay of a packet is reduced using the proposed encoding approach. It has been observed that the number of data transmissions in the network are reduced significantly using the proposed mechanism. Further, network lifetime is improved upto 21.5% and 56.22% with the proposed MDP based encoding mechanism in comparison to the end-to-end erasure coding and automatic retransmission mechanisms, respectively.

1.2.3 Energy and Delay Aware Routing Protocol using Fuzzy Logic System

In this work, a heterogeneous sensor actor network with different heterogeneity characteristics, such as battery capacity, quality of link, buffer capacity is considered. A node (v_i) has different levels of residual energy (E_i), buffer capacity (B_i), associated with link quality (from node j to node i is L_{ji}) and distance (from node j to node i is D_{ji}). Sensor nodes generate data (by sensing operation) and transfer the data to a local actor node through an efficient routing path.

Energy efficiency and reliability in data collection are the major issues in WSANs. In WSANs, a data collection protocol may satisfy the major application requirements such as, high data reliability and less delay along with energy efficiency. These requirements can be achieved by designing an energy efficient routing protocol. The sensor nodes may be deployed randomly in a dynamic environment [11]. This results in dropping of packets due to dynamic changes in quality of wireless links. The major reasons for dropping of packets may be bad wireless link quality, unavailability of free buffer at intermediate nodes and residual energy at nodes. To ensure the data reliability, the dropped packets need to be retransmitted and this leads to more delay and energy consumption. Therefore, an energy efficient routing protocol solves the above issues by taking efficient routing decisions while considering energy of a node, link quality and available buffer.

The key observation that motivates this work is the improper selection of a next hop node (in a routing path) that leads to dropping of packets. That means, selection of a node which has high available buffer, good wireless link quality, more residual energy and close distance (proximity) as next hop node reduces the packet dropping rate. However, reduction of the packet dropping rate increases the data reliability, energy efficiency and reduces the delivery delay [15, 13]. In this work, a fuzzy based delay and energy aware routing mechanism has been proposed to improve the lifetime of the network. A Fuzzy Logic System (FLS) can be used for blending different parameters and fuzzy rule set that may produce an optimal result [16, 17]. Residual energy, available buffer, quality of link and distance (proximity) are the primary considerable parameters to reduce the packet dropping rate at a node. FLS takes these parameters as input and produces the optimal output value (or a routing metric value called *chance of becoming next node*). Further, Dijkstra's algorithm has been adopted to find the routes where the determined routing metric value is considered as cost.

The major contributions of this work are as follows:

- Design of a delay and energy aware routing protocol using fuzzy logic for a heterogeneous sensor actor network.
- A routing metric (i.e. a routing parameter) has been computed by considering the residual energy, link quality, available buffer and distance (proximity).

A Fuzzy based Energy-Aware Routing Protocol

In the proposed fuzzy based delay and energy aware routing mechanism, a node (v_i) finds the efficient routing path by determining the fuzzy output value (p_j) (where, v_j is one of the possible next hop nodes) for next hop node. The fuzzy output value (indicates the value of routing parameter or *chance of becoming next node*) is a function of residual energy (E_j), available buffer (B_j), quality of link (L_{ij}) and distance (D_{ij}). Using the determined fuzzy output value, Dijkstra's algorithm finds the routing paths.

The proposed Fuzzy based Energy Aware Routing Mechanism (FEARM) has been compared with a PRD (*predicted remaining deliveries*) routing mechanism [15] and a retransmission based approach ([12]) in terms of average end-to-end delay, total number of retransmissions, lifetime, total energy consumption, data collection rounds at which half of the nodes die, data collection rounds at which all nodes die and network stability. Simulation results are presented by considering the different combinations of the fuzzy input variables, (i.e. FEARM with four parameters (residual energy, available buffer, link quality and distance), FEARM with three parameters (residual energy, available buffer and link quality), FEARM with residual energy and link quality and FEARM with residual energy and available buffer). Simulation results show that there is a reduction in (average) delay upto 58.78% with the proposed fuzzy based mechanism. Network (average) energy consumption is reduced upto 61.82% with the proposed mechanism. Further, it has been shown that lifetime of the network in case of *fuzzy with four parameters* is better than *fuzzy with three* and *two parameters*. It has been observed that the network stability is improved upto 17.72% in comparison to the retransmission based data forwarding approach

1.2.4 Fuzzy Logic Based Adaptive Duty Cycling Mechanism

In this work, a tree-based wireless sensor and actor network is considered, where the root is a local actor which collects sensing data from a sensor tree. Let, $T_i = (V, E)$ be a sensor tree, where $V = \{v_0, v_1, ..., v_n\}$ is a set of sensor nodes, v_0 is a local actor and E represents a set of communication links between the sensors. Here, duty cycle enabled sensor nodes are considered and a node harvests the energy from the sunlight.

A long operational lifetime of WSAN is required for applications, such as monitoring forest fires, nuclear plants, object tracking and military surveillance [2, 4]. Energy harvesting technology provides a long lasting lifetime to the WSAN. A node recharges its battery from the charging vehicles or from the natural resources, such as solar light, thermal, wind and vibration [4, 18]. A sensor node harvests energy periodically from ambient sources (such as sunlight). The harvested energy varies dynamically with the weather conditions over time [18]. The residual energy may reduce with usages and time. This results in temporary disconnection of the nodes from the network. The temporal-disconnected nodes (dead nodes) may join into the network in the next energy available time slot. Hence, the node should sustain (survive) till next available period of the energy source for improving the lifetime of the network. Achieving sustainability with solar based harvesting technology is one of the challenging issues which has not been addressed adequately in the literature.

To address the temporal death of the nodes, we have proposed a fuzzy based adaptive duty cycling algorithm to achieve the network sustainability in a tree-based energy harvesting sensor network. Moreover, a prediction model has been proposed to estimate energy consumption (i.e. residual energy) for a future interval of time. The fuzzy logic system estimates duty cycle value for a future time period by considering residual energy, predicted harvesting energy and predicted residual energy. Further, a switching decision has been taken based on the predicted duty cycle value of a node for survival. *Active and sleep schedules* are forwarded to the children nodes to reduce the packet drops. The major contributions of the work are as follows:

- Design of a fuzzy based adaptive duty cycling algorithm to achieve sustainability in an energy harvesting sensor actor network.
- Prediction of duty cycle using fuzzy logic by considering current residual energy, futuristic harvesting energy and residual energy.

The performance of the proposed mechanism is compared with a randomized switching algorithm [19] by considering network sustainability metrics, such as number of rounds network is connected, first node disconnected round, number of packets received at actor node and maximum number of dead nodes. From the simulation results, it has been observed that there is an improvement in network connected rounds on an average of 60.44% with the proposed mechanism. Survivability of a node is increased on an average of 47.13% with the proposed mechanism. Further, there is a reduction in the maximum number of dead nodes and a significant improvement in the total number of received packets at actor node with the proposed mechanism in comparison to a randomized switching algorithm.

1.3 Experimental Setup

In this thesis, *Network Simulator-3* (NS3) [9] has been used for simulating the proposed energy efficient and reliable data gathering mechanisms. The network topologies: flat and tree have been studied with the proposed mechanisms. A CSMA/CA based medium access control protocol (IEEE 802.11 standard) is considered in the simulation. E_{sense}, E_{rx} and E_{tx} are the parameters used to quantify energy consumptions by a sensor node in sensing, receiving and transmitting data over a distance $\hat{d}$, respectively. The energy consumptions are $E_{sense} = \alpha_3$, $E_{rx} = \alpha_{12}$, $E_{tx} = \alpha_{11} + \alpha_2 \hat{d}^{\hat{n}}$, where, α_{11} represents energy consumption per bit by the transmitter electronics, α_2 denotes the energy dissipated in the transmit op-amp, α_{12} represents the energy consumption per bit by the receiver electronics, α_3 denotes the energy cost for sensing a bit and $\hat{n}$ represents the path loss exponent. The following simulation parameters are considered in this thesis: the number of nodes is 1000, transmission range is 30 m, the minimum distance between any two sensor nodes in a tree is 5 $meters$, $\alpha_{11} = 0.937 \times 10^{-6}\ Joules/bit$, $\alpha_{12} = 0.787 \times 10^{-6}\ Joules/bit$ [20], $\alpha_2 = 10 \times 10^{-12}\ Joules/bit$, $\alpha_3 = 50 \times 10^{-9}\ Joules/bit$, $\hat{d}$ = 85 $meters$ and path loss exponent $(\hat{n})$ is 2 [7], a data packet size is 960 $bits$. The above parameters are used to evaluate the performance of the proposed algorithms (as discussed in sections 1.2.1, 1.2.2, 1.2.3 and 1.2.4).

1.4 Organization of the Thesis

The main focus of this dissertation is to design and analyze energy efficient and reliable data gathering algorithms in wireless sensor and actor networks by considering dynamic environment conditions. The proposed approaches achieve energy efficiency and data collection reliability with stringent delivery delay requirement. The thesis has been organized into seven chapters.

Chapter 1: In this chapter, a brief introduction to the wireless sensor and actor networks and objectives of the thesis have been presented. It also describes an overview of the major contributions and outline of the thesis.

Chapter 2: In this chapter, energy efficient communication protocols based on duty cycle, network coding and load balancing have been surveyed. A survey on reliable data collection protocols is discussed. The challenges in energy harvesting WSNs have been presented.

Chapter 3: In this chapter, a Markov Decision Process (MDP) based buffer management mechanism has been proposed in tree-based WSAN to reduce the data delivery delay and to improve the energy efficiency. A mathematical model for a mobile actor is presented to analyze buffer occupancy and energy consumption with event-centric network traffic. An energy consumption model is presented to estimate energy utilization of the proposed network model.

Chapter 4: A reliable data transmission mechanism using opportunistic encoding has been proposed for a WSAN with faulty nodes. A Learning Automata based mechanism has been designed to route the traffic.

Chapter 5: In this chapter, a fuzzy logic based delay and energy-aware intelligent routing mechanism has been proposed to select efficient routes. The network performance has been analyzed with different combinations of fuzzy input variables.

Chapter 6: A fuzzy based adaptive duty cycling algorithm has been proposed to achieve the network sustainability in harvesting sensor actor networks. Duty cycle has been predicted using fuzzy logic system by considering current residual energy, futuristic harvesting energy and residual energy. Energy consumption of bottleneck zone in a duty cycled sensor network is estimated in this chapter.

Chapter 7: This chapter summarizes the work, outcomes of the contributions and future scopes for expansion of the work.

Chapter 2

Literature Survey

A Wireless Sensor Network (WSN) is a group of sensor nodes that are connected among themselves through a wireless medium to perform distributed sensing tasks [3]. There exist different kinds of sensors to collect sensory data, such as weather information, intensity of rainfall and light, the speed at which the wind flows etc. There are many application areas in which WSNs are adopted, they include, tracking of objects, monitoring and prediction of natural phenomenon [3, 21]. Latest advances in technology have led to the development of wireless sensor and actor networks (WSANs) [2, 22]. A Wireless Sensor and Actor Network (WSAN) is a type of sensor networks where one or more actors are distributed in the network for data collection and local decision making [1, 2, 3]. In WSANs, sensor nodes collect the data about the physical world and actors take action decisions and perform appropriate actions upon the environment [23]. Sensor nodes are less in cost, have limited sensing power, computation and wireless communication potentiality. Whereas, actor nodes are more powerful with high processing potential, more transmission power and long life battery [2, 24]. Also, the sensors count in a particular target place is high in number compared to the actors count. This is obvious because of high potentiality of actor nodes. Since WSAN is a type of WSN, we have discussed a comprehensive survey on protocols application for WSN. Further, sensor-actor architectures are also discussed in this chapter. Features of WSANs are discussed below.

1. *Real time communication:* There exists some applications where WSANs should

respond very quickly for the sensor input. For example, consider a fire application where the initiation of the action should be done very quickly. Therefore, reliable data transfer protocols are required to transfer the data from source sensor to actor with less delay.

2. *Coordination:* In WSANs, there are two types of coordination. One is sensor-actor coordination and the other is actor-actor coordination [25]. In case of sensor-actor coordination, the event information is being transmitted from sensors to actors. Now, based on this event information, the actors should coordinate with each other and make the most suitable action. This form of coordination among the actors is referred as actor-actor coordination.

2.1 Physical architectures of Wireless Sensor and Actor Networks

The responsibility of sensor and actor nodes in a WSAN is to collect data from the physical world and perform the necessary actions based on the collected data. Considering the physical architecture of WSAN, there exists three components, namely, sensor/actor field, sink and task manager node. The field in which sensors and actors are distributed is known to be sensor/actor field. The node that monitors the network is said to be a sink node and it communicates with both the task manager node and sensor/actor field. Based on the existence of central controller, the physical architecture of WSAN can be divided into two types [3]. One is automated architecture (refer Fig. 2.1) and the other is semi-automated architecture (refer Fig. 2.2). In case of former architecture, the sensors detect the phenomenon and transmits the data to the actors where the processing is done and suitable actions are initiated. There exists no central controller and hence it is called as automated architecture. In case of latter architecture, the sink acts as a central controller and collects the data and coordinates the suitable actions to the actor nodes. The selection of architecture depends on type of applications. The semi-automated architecture is similar to the existing architecture of wireless sensor networks. Protocols can be adopted. In case of au-

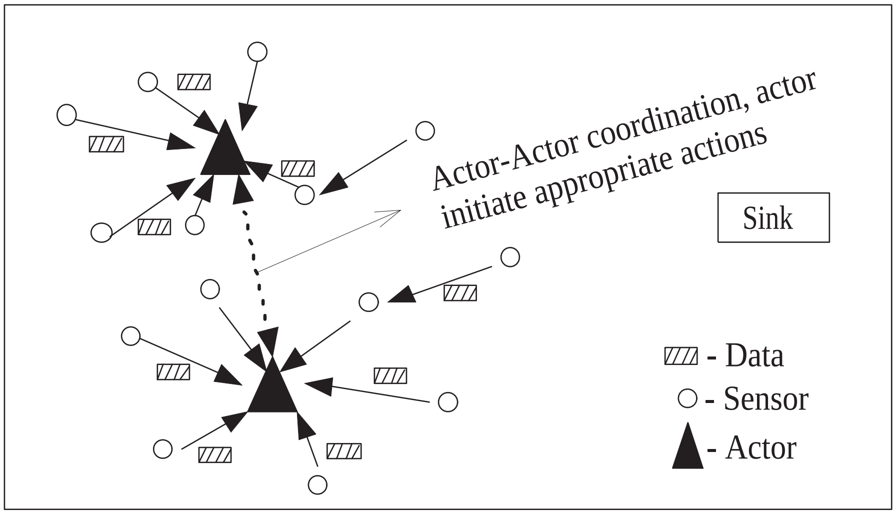


Figure 2.1: Automated Architecture in WSAN

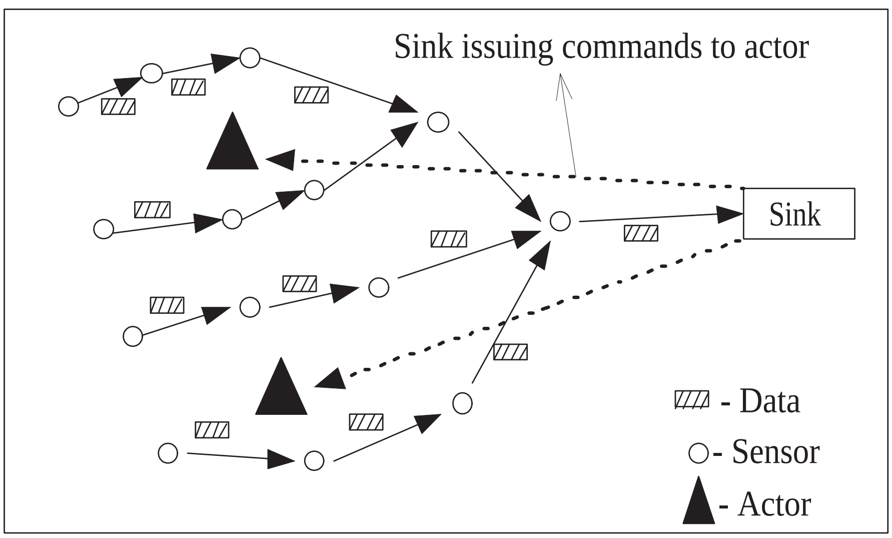


Figure 2.2: Semi-Automated Architecture in WSAN

tomated architecture, the data transfer is from the sensors to actors. This approach reduces latency as compared to the semi-automated architecture where the sink communicates to actors which are separated by a large distance from the sink node.

In semi-automated architecture, the sensed information is passed from the sensor nodes to the sink. Hence, the sensor nodes have to participate more in relaying the data and when one of these nodes fails, it may result in the loss of a connection thereby losing the network connection. There is more of a chance that sensor nodes near the sink have a higher probability of failure than the remaining nodes. In automated-architecture, the sensed information is passed from sensor nodes to the actor nodes. The sensor nodes which are at one hop distance from the actor nodes have a higher load of data relaying. Different actors will be triggered for each event. This implies that relaying sensor nodes will also be different for each event. This has the advantage of an evenly distributed relay load on the nodes. From this, it is obvious that the automated architecture has a longer life time compared to the semi-automated architecture. However, the selection of architecture is application dependent. In this thesis, energy efficient data gathering algorithms are designed for automated architecture.

2.2 Applications of Wireless Sensor and Actor Networks

The advances in technology allows the applications of WSANs in various systems for monitoring and decision making, such as agriculture micro-climate control, military, home automation, early detection of disaster, monitoring health, chemical and biological attack [2, 22]. An overview of main application areas of WSANs are stated below:

1. *Environmental Monitoring:* Environmental monitoring applications are one of the most important applications of WSANs. It includes, the early prediction of natural disasters, such as flood and earthquake. In the smart cities, WSANs help to monitor the quality of air and water [22]. In case of fire accidents, sensors relay the location and fire intensity to water sprinkler actors so that the fire can be extinguished before spreading.

2. *Smart Home Applications:* Nowadays, home automation demand has increased rapidly [26]. In the smart home, motion and light sensors detect people and then actors execute appropriate actions based on the user preferences. Moreover, Controlling lights, automation in electronic appliances, power controlling, home security are highly recommended applications of WSANs for smart home.

3. *Health Monitoring:* The wireless sensor networks found the way to monitor the health using various types of sensors which are embedded in the body of a patient.

WSANs assist the patients through warnings. Sensors with integrated GPS offer emergency help at a particular place. Advances in this applications would achieve enhancement quality of life for patients and the medical treatment will be more flexible and controllable [27].

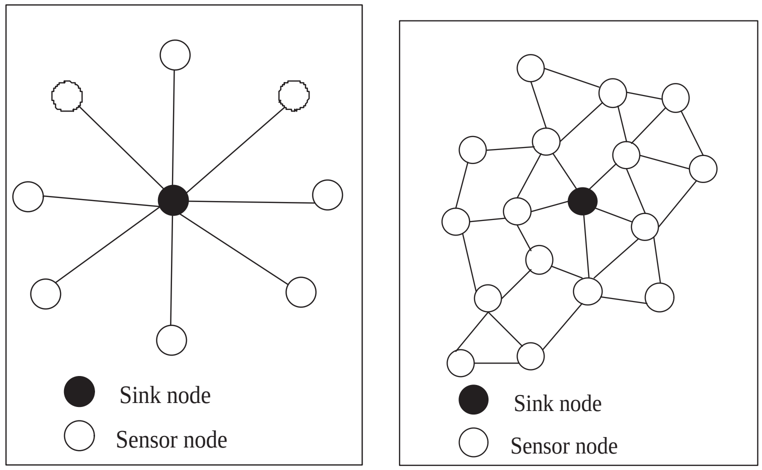
4. *Smart Transportation Applications:* The WSANs provide traffic monitoring, jam reporting, vehicle accident alerts and intelligent traffic management system [22]. Synchronization among components and delay are critical issues in smart transportation systems.

5. *Agriculture and Forest Applications:* Wireless sensors provide many solutions to improve the quality of agricultural products. The major applications include crop management, smart irrigation, soil monitoring, weather prediction and plantations. The WSANs also provide significant solutions for various issues in forest like fire detection, flood detection, animal monitoring, animal tracking and forest health monitoring [28]. Energy efficiency and reliable data transfer are important issues in agriculture and other monitoring applications.

2.3 Communication Architectures in Wireless Sensor Networks

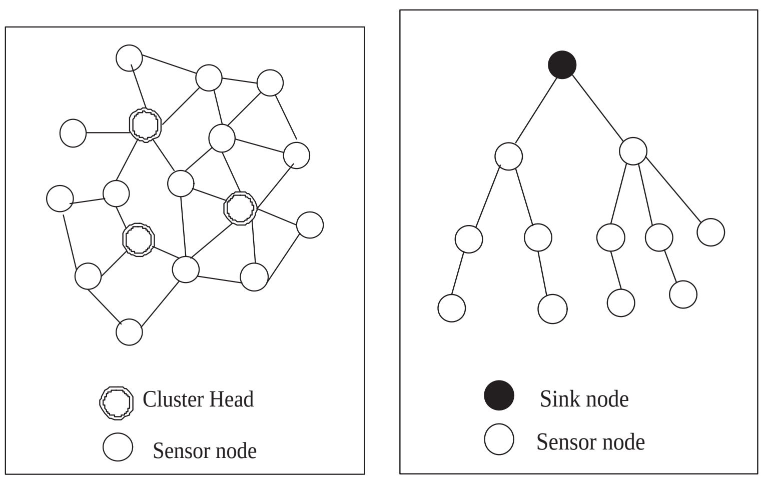
The current network topologies that are best suitable for WSNs are star, mesh, tree and clustered hierarchical architectures (refer Fig. 2.3 and Fig. 2.4). Each topology has its advantages with its own performance metrics. Also, the organization of the network and its connectivity are affected by the network topology. It is also worth noting that the topology plays a major role in the design and construction of a wireless sensor network. The major topologies such as star, mesh, tree and clustered hierarchical architectures are discussed below.

A star topology is the simplest one that can organize the sensor nodes [29]. Each node is connected individually to a central node which is also called the hub or a sink node. The central node can act as a base station or a gateway that directly communicates with the base station. The nodes which are connected to the central node are said to be the remote nodes.



(a) (b)

Figure 2.3: (a) Star topology (b) Mesh topology



(a) (b)

Figure 2.4: (a) Clustered topology (b) Tree topology

In case of star topology, the only communication is between a central node and the remote sensor nodes, but not among the remote sensor nodes. In other words, the remote nodes are capable of sending and receiving the messages to and from the central node and they cannot exchange messages among themselves. This form of communication results in less consumption of energy for any remote node. This minimized energy consumption is the main advantage of the star topology. Also, the failure of any remote sensor node does not affect the overall functioning of the network. However, failure of the central node is the bottleneck which makes the network less reliable. This is because, the central node is the controller of all the remote nodes attached to it. Hence, single point of failure at the central node is treated as the major disadvantage for this type of topology. Further, the central node should be within a particular radio transmission range and may not be a choice for the applications where a robust network is required.

A mesh topology organizes the sensor nodes in such a way that each node is interconnected to as many sensor nodes as possible in a non-hierarchical manner [29]. Basically, it is a local network topology in which each sensor node is capable to send and receive messages. This is opposed to the star topology where the data flow is only between the central node and remote nodes. Unlike star topology, there is no single node that is responsible for data transfer and dependency on one node do not exist in mesh networks. Also, every node in the network acts as a router and relays the information to the neighboring nodes. In order to relay the messages, mesh topology makes use of either of the two techniques, namely, flooding or routing. Moreover, if any node in the network fails, mesh topology has the capability of dynamically distributing the work to other remote node. This is the reason that makes the mesh topology more fault-tolerant. Also, expansion and modification of the network can be done flexibly without disrupting the other nodes in the network. Therefore, the mesh topology has high scalability. However, as the number of nodes increase, the cost of the network also increases. Since it is a multi-hop network, the nodes which are close to the base station may drain fast and this effects the life period of the network. Also, due to the increase in the number of hops, there is a possibility that message delivery time also increased. Hence, the efficient way of organizing the network is to minimize the amount of messages that are being transferred between the nodes. This can be achieved by partitioning the nodes into groups also called as clusters and the topology is named as clustered hierarchical topology.

Clustered hierarchical topology is a multilayer model in which the lowest layer sensor nodes in the WSN are grouped to form a cluster [29]. The local neighborhood information in the sensor nodes is used to create the clusters and a cluster head is being elected as a representative of a particular cluster. Electing a cluster head mainly depends on the factors, such as the total number of sensor nodes that are neighbors to the nodes that participate in the election, the proximity of the nodes to the base station, the total energy resources available etc. Further, the cluster heads of the lower layer are grouped to form the clusters at the higher layer. The election process for a cluster head is repeated among the nodes in this layer too. This process of repetition at each layer results in a hierarchy of clusters. In summary, this hierarchical structure represents a tree in which the root node is called the sink, cluster heads are the internal nodes and the sensor nodes acts as leafs. Even though, it looks like a tree topology, it follows the multi-hop routing which is common in mesh topology. That is, the paths between the lower and higher layers can be multi-hop routes.

The main advantage of clustered hierarchical topology is the efficient distribution of workload among the nodes. Also, the cluster head may vary and it is not fixed. A cluster head is responsible for communication within the cluster nodes, also known as intra cluster communication and communication among the cluster heads which is referred as inter cluster communication. As it is already said that the communication is done using multi-hop routing, the nodes that are closer to the base station reduce their energy faster as compared to the other nodes. This is due to the fact that there will be high traffic at these nodes which are closer to the base station. Also, this effects the network coverage area and may result in hot spot problem [30]. Moreover, there exists few challenging issues, such as selection of appropriate cluster head, rotation process involved in the selection of new cluster head, and the number of nodes to be formed as a cluster.

In this thesis, Chapter 3 and Chapter 6 are based on tree-based WSANs. In Chapter 3 and Chapter 6, flat WSANs are considered.

2.4 Energy Efficiency in Wireless Sensor Networks

Energy efficiency is a major issue in the design of communication protocols in a wireless sensor network. The lifetime of the network depends on the energy efficiency. The lifetime of wireless sensor network is represented in several ways in the literature [31]. Network lifetime is estimated using different energy consumption models based on network topologies. There have been studies on lifetime and energy consumption estimation models in WSN [5, 7, 21, 32, 33]. In [7], Bhardwaj *et al.* have derived upper bounds on lifetime for data gathering in sensor networks. Wang *et al.* [5] have estimated the energy consumptions of bottleneck zone in energy constrained wireless sensor network. In [32], Lee *et al.* have estimated the upper bound on the network lifetime in cluster-based sensor networks. Rout *et al.*[33] have estimated the lifetime of WSN by considering duty cycle and network coding. Pantazis *et al.* [21] have provided an exhaustive survey on energy consumption models in wireless sensor networks.

Bhardwaj *et al.* [7] have demonstrated that lifetime of the network depends on factors like region of observation, the source behavior, base station location, radio energy and path loss parameters characteristics, number of nodes and initial energy. The study considered the following parameters to quantify energy consumptions. E_{sense}, E_{rx} and E_{tx} are the energy consumptions by a sensor node in sensing, receiving and transmitting data over a distance d, respectively. The energy consumptions are given by $E_{sense} = \alpha_3$, $E_{rx} = \alpha_{12}$, $E_{tx} = \alpha_{11} + \alpha_2 d^n$, where, α_{11} represents energy consumption per bit by the transmitter electronics, α_2 denotes the energy dissipated in the transmit op-amp, α_{12} represents the energy consumption per bit by the receiver electronics, α_3 denotes the energy cost for sensing a bit. The authors assumed that the energy model follows $\frac{1}{d^n}$ path loss behavior (n is path loss exponent). The authors have derived an upper bound on the lifetime for a linear multi-hop wireless sensor network. The energy consumption for relaying a bit over distance D is bounded as

$$E_{relay} \geq \alpha_1 \frac{n}{n-1} \frac{D}{d_{char}} - \alpha_{12}$$

where, d_{char} is an optimal hop length and $d_{char} = \sqrt[n]{\frac{\alpha_1}{\alpha_2(n-1)}}$, $\alpha_1 = \alpha_{11} + \alpha_{12}$.

Wang *et al.* [5] have considered a bottleneck zone in a wireless sensor network. The authors have analyzed that the energy resources of the nodes inside the bottleneck zone imposes upper bounds on the network performance. Further, the study derived the performance bounds by considering the network deployment variables, such as the number of nodes and size of deployment area.

2.5 Sink and Actor Mobility in Sensor Networks

In a typical sensor network, a node relays the sensed data to sink node using multi-hop path communication. The nodes (bottleneck zone nodes) which are near to the sink node consume more energy than other nodes in the network due to heavy data traffic towards the sink node [5]. Therefore, the energy of bottleneck zone nodes will be depleted quickly and the network will be partitioned which leads to disruption of the data gathering process to the sink. This problem is called the hotspot problem [34]. Mobility of the sink node relieves data traffic burden of the bottleneck zone nodes and enhances energy efficiency. A mobile sink moves across the network to collect sensed data from the nodes (it spreads the energy drainage of bottleneck zone nodes to the entire network). This results in uniform energy consumption. Thus, mobile sink provides load balancing implicitly. Mobility of sink (or any actor) reduces the data dissemination paths that increases the throughput and reliability with reduction in energy consumption. In sparse and disconnected networks, mobile sink gathers the sensing data from isolated portions of the network whereas a static sink cannot gather data from disconnected regions.

A mobile sink can collect the sensed data either by periodically flooding the fresh position of mobile sink to the network or by traveling through communication range of a sensor node. The first approach suffers from the overhead of flooding. The second approach is energy efficient due to less overhead of control packets. However, the nodes need to wait for the sink to transmit data. A large delay of mobile sink visit may lead to dropping of packets (due to buffer overflow of a node). In recent studies, the mobility has been used not only for gathering the data but also for recharging the nodes [35, 36]. A charging vehicle is equipped with resonant coils that moves across the network to recharge the nodes via wireless energy transfer. In some of the studies, a mobile charger is combined with a mobile base station for data gathering [37, 4].

In the literature, three sink mobility classes are discussed: Mobile Base Stations (MBS), Mobile Data Collectors (MDC), and Rendezvous-Based solutions [38]. The sink mobility is classified into two categories: random mobility and controlled mobility. In random mobility category, a mobile sink moves randomly over the network and collects the data.

The random mobility schemes are easy to implement but they suffer from uncontrolled behaviors and poor performances. In controlled mobility category, the trajectory of mobile sink will be scheduled in such a way that the delay of data delivery can be reduced. Data gathering protocols based on the random mobility category are presented in [39, 40, 41]. Shah *et al.* [39] have designed an architecture where mobile entities (called *MULEs*) collect data from sensors when they come closer to the sensor nodes. In [40], Tong *et al.* have proposed a network architecture called *SEnsor Networks with Mobile Agents* (SENMA) for low power and large scale sensor networks. In this model, mobile agents communicate with a large field of sensors opportunistically. The problem of energy efficient data collection has been addressed by Jain *et al.* [41]. The authors have presented and analyzed the *MULE* architecture to extend the lifetime of the network by minimizing the communication responsibility of sensor nodes.

The works presented in [42, 43, 44, 45] are based on the controlled mobility models. Gandham *et al.* [42] have proposed a heuristic approach that determines the locations of mobile base stations to increase the lifetime of wireless sensor networks. Further, Wang *et al.* [43] have designed a linear optimization model which determines visiting locations of mobile sink and also waiting time at visiting locations for maximizing the lifetime of a grid network topology. In [44], optimal movement of a mobile base station problem has been provided using a provably optimal algorithm in wireless sensor network. Luo [45] *et al.* have developed an analytical model which jointly considers mobility and routing for characterizing the network lifetime.

Hossain *et al.* [46] have proposed equal energy dissipation condition by exact placement of nodes to ensure equal energy dissipation in wireless image sensor network. Ren *et al.* [47] have considered data collection maximization problem in energy renewable sensor networks in which a mobile sink travels along a predefined path to collect data from sensors and also discussed issues of energy renewable sensors. In [48], Gu *et al.* have discussed delay-bounded sink mobility problem by a mathematical model in which issues, such as sink scheduling, data routing, bounded delay are considered. Umer *et al.* [49] have introduced a hybrid rapid response routing protocol for a wireless body area sensor network to transmit delay sensitive data for patients. A mobile data collector has been introduced by

Rao *et al.* [6] to prolong the lifetime of sensor network. In [6], heuristic traveling paths for a mobile actor have been designed under two constraints, namely, *data overflow on a sensor node* and *timeliness of each data*.

Ma *et al.* [50] have proposed a greedy algorithm where single and multiple mobile data collectors are considered for data gathering. It focuses on the problem of minimizing the length of data gathering tour. The authors also consider multiple collectors to gather data for applications with strict distance and time constraints. In [51], Zhao *et al.* have studied the tradeoff between energy saving and data gathering latency in mobile data gathering. Zhao *et al.* [52] have proposed an efficient data gathering scheme which is a joint design of mobility and space-division multiple access (SDMA) technique in WSNs. The authors focus on minimizing the data gathering time by exploring the trade-off between the shortest moving tour and the full utilization of SDMA.

In this thesis, a model for mobile actor has been formulated and presented in Chapter 3. The bounds on visiting times and contact times are estimated for a mobile actor to mitigate buffer overflow problem.

2.6 Duty Cycling in Sensor Networks

Duty cycling is a widely used mechanism to reduce energy consumption in wireless sensor networks [53]. It aims to reduce the sleep state of sensor nodes and increase the network life time consequently. The two possible states of a sensor node are *active* and *sleep* states. A duty cycle can be defined as the ratio of the time spent in active state to the total time. In other words, the fraction of time a sensor node is active during its entire life time in a wireless sensor network is said to be its duty cycle [54, 33]. Basically, there will be a longer life time by choosing appropriate duty cycle. Generally, a node turns off most of the components when no communication takes place and goes to sleep state [53].

In the process of reducing the time of sleep state, duty cycling mechanism must reduce collision rates and control packet overhead. Moreover, an increase in collision rate and control traffic can cause serious energy consumption [53]. Duty cycling mechanism may also affect the network coverage and connectivity. Hence, redundancy can be introduced during the deployment of sensors so that lower duty cycles do not affect the network performance [54]. Even though redundancy helps in avoiding the degradation of network performance, fixing the redundancy level is not a simple task. Also, the main challenging issue in designing a duty cycle mechanism is to determine the active and sleep state schedule. Based on the above issues, duty cycling can be classified into two types, namely, random duty cycling and coordinated duty cycling [54, 53, 33].

In case of random duty cycling, the manner in which a sensor node is made active and sleep is independent of the other sensor nodes. This fact of sensor nodes being independent of each other leads to zero control packet overhead and also makes the random duty cycling a simpler mechanism. Whereas, in coordinated duty cycling, the sensor nodes communicate with each other and exchange the messages to make them active and sleep. This way of coordinated sleep schedule results in more robust network in terms of network connectivity and coverage. But, the problem is that more energy will be consumed in achieving the coordination among the sensor nodes.

The random and coordinated sleep mechanisms based on the network coverage using low duty cycling have been studied by Hsin *et al.* [55]. In their work, the authors have presented two scheduling algorithms for random sleep and coordinated sleep respectively. They have studied the performance metrics, such as coverage extensity and coverage intensity [55]. Further, a detailed investigation on random duty cycled wireless sensor networks based on the coverage intensity has been discussed in [54]. The problem of enhancing the lifetime of a network using duty cycle in a wireless sensor network has been addressed in [33].

In [56], a distributed delay efficient data aggregation scheduling mechanism has been proposed for duty cycled WSN. Yoo *et al.* [57] have proposed dynamic duty cycle scheduling schemes to reduce sleep latency and achieve the balanced energy consumptions for an energy harvesting WSN. Wu *et al.* [58] have proposed a delay aware energy efficient flooding mechanism for synchronous duty cycled WSNs.

In Chapter 6, a coordinated duty cycle WSAN has been considered. In this work, duty cycle of a sensor node is determined to achieve the sustainability in an energy harvesting sensor actor network.

2.7 Network Coding in Wireless Sensor Networks

The term network coding is firstly introduced by Ahlswede *et al.* [59]. Network coding is a novel technique that has been introduced in multi-hop networks for improving network throughput and reducing packet delay. In coding technique, intermediate nodes encode the data packets before transmission. Network coding reduces the number of transmissions in the network and therefore, improves the network energy efficiency. In WSNs, packets will be dropped due to unreliable wireless links and node failures. In WSNs, network coding has been incorporated to improve the performance of a WSN [59].

In [59], the messages received at the intermediate nodes are being encoded before forwarding and this results in reduction in the network traffic load. The network information flow problem is also introduced in [59]. Ahlswede *et al.* have considered a point-to-point communication network. The scalability, throughput and efficiency of the network have been improved in [59]. Further, in [59], the authors have proposed a mechanism to maximize the network information flow. This mechanism involves grouping of multiple packets and transmits them instead of simply forwarding the received packets. In [60], the number of grid transmissions and circular configurations are decreased in-order to increase the life time of the network nodes. This is achieved because of the network coding mechanism. They also present the interaction among *traffic rate, energy of the nodes, density* and *the number of nodes*. Hong *et al.* [61] have proposed a network coding based distributed algorithm for data gathering to maximize the network lifetime. Wang *et al.* [62] have proposed an algorithm for the 2-hop information exchange in WSN. They have also tested the algorithm on *grid* and *random topologies*.

The encoding technique improves the network bandwidth efficiency in comparison to the traditional store and forward mechanism [63]. In wireless networks, packets are dropped due to the link failures. AI-Kofahi and Kamal *et al.* [64] have designed a network coding based approach to mitigate the effect of packet loss. Erasure coding is a popular technique to increase the data transmission reliability by generating redundant data. Srouji *et al.* [14] have designed a reliable data transfer scheme (RDTS) using erasure coding to reduce data redundancy and to improve the network lifetime. In [14], encoding is performed hop-by-hop manner i.e. an intermediate node determines the number of redundant packets adaptively for next hop. A network coding based communication paradigm has been proposed by Rout *et al.* [33] in a wireless sensor network to reduce traffic load in bottleneck zone. In [8], an adaptive data aggregation strategy using network coding has been proposed to improve the energy efficiency in a clustered sensor network.

2.7.1 Linear Network Coding Technique

In this section, a linear network coding process is explained. Let us assume that n packets ($C_1, C_2, ..., C_n$) are sensed by source nodes in a network. These packets are transmitted to the destination via intermediate encoder nodes. Assume, k is total number of packets that can be encoded into a single packet at a time and therefore, the number of encoded packets generated by a encoder node is n/k. The encoding and decoding process of linear network coding is explained below.

Encoding process: The coded packets are viewed as elements in finite field $GF(2^s)$, where 2^s is the size of the finite field. In the encoding process, a node selects a sequence of coefficients $q = (q_1, q_2, ..., q_n)$, called encoding vector, from finite filed $GF(2^s)$. A single output packet is generated by linear combination of a set of n packets $G_i(i = 1, 2, ..., n)$ arriving at a node. The encoded output is generated by computing the following linear combination:

$$Y = \sum_{i=1}^{n} q_i G_i, q_i \in GF(2^s) \tag{2.1}$$

Every node in the network computes similar linear combination (as given in equation 2.1) for transmitted data. The encoding vector (q) is also transmitted with the coded packet and it is used in the decoding process by the receiver.

Decoding process: A receiver node retrieves the original packets from the set of linear combination of packets. Let, a node receives $(q^1, Y^1), ..., (q^m, Y^m)$ (m is number of received coded packets). The symbols Y^j and q^j denote the information symbol and the encoding vector for the j^{th} received packet, respectively. To decode the coded packets, the following linear equation need to be solved.

$$Y^j = \sum_{i=1}^{n} q^j{}_i G_i, j = 1, ..., m \tag{2.2}$$

The above linear system (Equation 2.2) has m equations and n unknowns. A receiver must receive at least n linearly independent packets for proper decoding of the coded packet. In Equation 2.2, the only unknown is G_i which contains the original data packets transmitted in the network. The n number of original packets can be retrieved by solving the linear system (Equation 2.2) after getting n linearly independent packets. In the next section, XOR coding mechanism is explained. XOR coding is a special case of linear network coding. In XOR coding, the network coded packets (that are elements in $GF(2) = 0, 1$) are forwarded in the network.

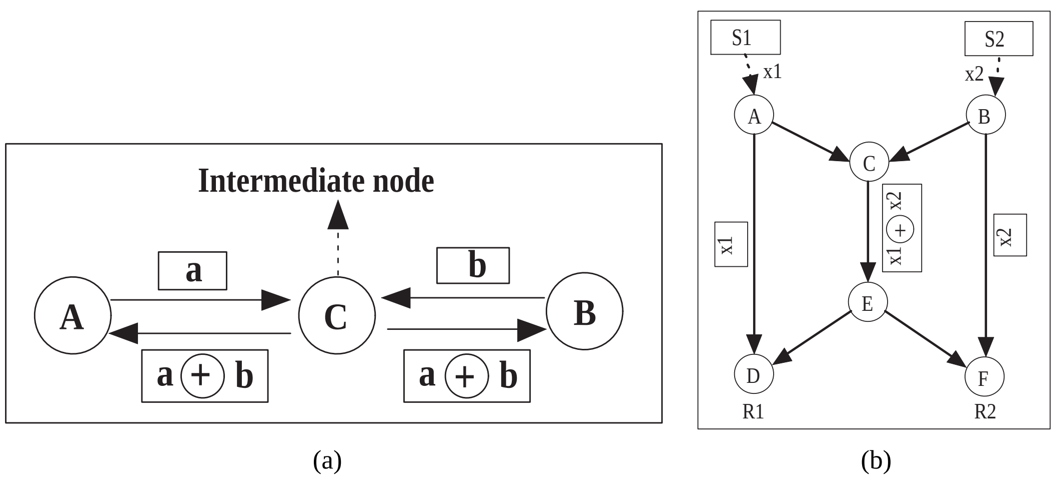


Figure 2.5: (a) Packets exchange between two nodes using XOR network coding (b) The Butterfly Network (Sources S_1 and S_2 multicast their information to receivers R_1 and R_2)

2.7.2 XOR Network Coding

An example of XOR network coding scheme is presented in Fig. 2.5(a). Node A and B are not in the transmission range of each other and intermediate node C falls in the transmission range of both nodes A and B. Thus, nodes A and B will exchange the data packets through the intermediate node C. In XOR coding, node C relays coded packet ($a \oplus b$) to nodes A

and B with a single transmission, after receiving the packets a and b from the nodes A and B, respectively. Then node A decodes the coded packet (i.e. $(a \oplus (a \oplus b))$) to get the packet b. Similarly, node B gets the data packet a. In the absence of XOR coding, relay node C needs to perform two separate transmissions for relaying both data packets. Thus, network coding reduces the number of data transmissions and improves the bandwidth efficiency [65, 66].

Fig. 2.5(b) shows the butterfly network where linear network coding increases the network throughput [67]. Two source nodes (S_1 and S_2) have packets x_1 and x_2 that must be transmitted to the two destination nodes (R_1 and R_2), which each want to know both x_1 and x_2. Each edge can carry only a single value at a time. R_1 receives both packets if all the network resources are utilized by it. We could route the packet x_1 from source S_1 along the path AD and the packet x_2 from source S_2 along the path {BC, CE, ED}. Similarly, if receiver R_2 uses all the network resources, then it could also receive both packets (the packet x_2 from source S_2 along the path BF and the packet x_1 from source S_1 along the path {AC, CE, EF}). In multicasting, both receivers should receive the packets simultaneously from both sources. But, we can send only packet through link CE per time slot. Routing is insufficient to transmit both packets simultaneously to both destinations. Using a XOR coding technique, two source nodes can be transmitted to both destinations simultaneously. Node C can take packets x_1 and x_2 and XOR them to create a third packet $x_3 = x_1 \oplus x_2$ and transmitted through edge CE. R_1 receives x_1 and $x_1 \oplus x_2$ and it can decode to retrieve x_1 and x_2. Similarly, R_2 receives x_2 and $x_1 \oplus x_2$ and it can decode to retrieve x_1 and x_2. XOR network coding increases the network throughput when multicasting.

2.8 Markov Decision Process Approaches in Sensor Networks

Wireless sensor network operates in dynamic (random) environments and most of the system parameters are dependent on the environment factors [68]. To achieve long lifetime and low maintenance cost, a WSN requires adaptive and robust mechanisms to address data dissemination, topology formulation, sensing coverage and other resource optimizations. In these problems, a set of optimized decisions or actions must be taken to achieve design goals in WSNs. In such stochastic environment, a mathematical framework called *Markov Decision Processes* (MDPs) helps to model the system dynamics to optimize the desired objectives of the network [68]. According to markov theory, the future state of a sensor node depends upon the current state rather than the past state. MDP is a powerful decision making approach, which deals with the system variation and adaptive policy management (by selecting best actions in each state). The important components of MDP model are *states of the system*, *set of actions or policies*, *state transition probability*, *reward function* and *decision epoch* [68].

Alsheikh *et al.* [68] have discussed applications of *markov decision process* for a wireless sensor network to solve the resource management issues. There are various models (designed based on markov decision process) that deal with the issues of delay-energy tradeoff in WSNs. Lin *et al.* [69] have designed a MDP based algorithm for delay-sensitive WSNs under dynamic environments. The algorithm enables the nodes to determine their appropriate transmission power to maximize the network utility. A node's propagation gain of wireless channel and queue size are considered to define the MDP's states. Similarly, Hao *et al.* [70] have analyzed the energy consumption and delay tradeoff in sensor networks using markov decision process framework. Guo *et al.* [71] have developed a MDP based opportunistic routing protocol with controlled transmit power level in WSN. This scheme reduces the end-to-end delay and consumes less energy. In [72], a markov decision process model has been designed by Mohaptra *et al.* to solve the energy-delay trade-off issue for network coding decisions in a two-way relay network. In [73], the MDP approach achieves an optimal tradeoff between energy efficiency and communication performance.

MDP has been used to solve neighbor discovery problems in WSNs [74, 75]. In [74], Madan and Lall have considered the problem of neighbor discovery for a randomly deployed sensor network. A MDP model is designed to solve the neighbor discovery problem to minimize energy consumption. In [75], Stabellini extended [74] and developed an MDP-based neighbor discovery algorithm using dynamic programming for WSNs. In [74], the authors have not considered the contention windows and collisions in dense networks.

This limitation has been addressed in [75].

In energy harvesting sensor networks, various MDP based models are proposed to balance the tradeoff between the energy consumption and harvesting. Murtaza [73] *et al.* have proposed an optimal data transmission and battery charging policy for solar powered sensor networks. Markov decision process has been incorporated to model the behavior of a node. A markov decision process approach has been presented to analyze the optimal recharge policies in a rechargeable sensor network by Misra *et al.* [76]. Kar *et al.* [77] have addressed an optimal sensor activation problem in a rechargeable sensor network to maximize the network performance. Fernandez-Bes *et al.* [78] have proposed MDP based censoring policy to optimize the expected sum of transmitted messages in harvesting sensor network. Rout *et al.* [79] have proposed a *markov decision process* based switching algorithm for a rechargeable sensor network to increase sustainability in a data collection tree.

In Chapter 3, an adaptive buffer management using markov decision process has been designed to reduce end-to-end delay and to improve energy efficiency in tree-based wireless sensor and actor networks. In Chapter 4, a markov decision process is designed for opportunistic network coding decisions and determining the level of packet redundancy in the network coding process for reduction of data transmissions.

2.9 Fuzzy Logic in Wireless Sensor Networks

In the literature, several studies are performed on fuzzy based approaches to analyze the network lifetime [80, 81, 82]. Fuzzy logic is a suitable technique for making real time decisions and modeling uncertainties of a system. Moreover, fuzzy logic controller can be used by combining different parameters and rules and that produce a result. Fuzzy logic is less complex and flexible and it requires less computational resources. In classical set, an element is associated with zero or one. However, in fuzzy set, every element is associated with some degree of membership. The fuzzy logic system (as shown in Fig. 2.6) has four modules, namely, fuzzifier, fuzzy rules set, fuzzy inference system and defuzzifier. In the fuzzifier module, membership functions are used to determine the membership values

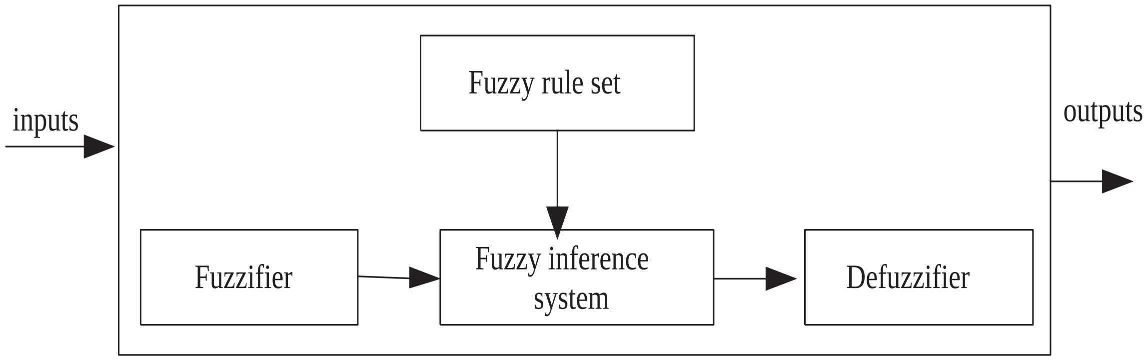


Figure 2.6: Block Diagram of Fuzzy Logic System

and membership levels (linguistic levels) for given crisp input parameters. Fuzzy rules can be generated using experimental data or heuristic data. Fuzzy inference system infers the fuzzy output by applying the fuzzy rule set to the membership functions. Finally, defuzzifier transforms the fuzzy output value to a crisp value. Centroid and maximum methods are well used techniques for defuzzification [83, 84, 17].

In the existing works [85, 86, 87, 88, 17], fuzzy logic systems have been used in wireless sensor networks for cluster head selection, efficient decision making, network lifetime maximization, energy consumption minimization and efficient routing protocols. In wireless sensor networks, fuzzy logic approaches are designed to handle uncertainties in clustering algorithms [85, 86, 89]. Generally, fuzzy logic helps in blending different clustering parameters for selecting a cluster head. Gupta et al. [85] have proposed a novel mechanism based on fuzzy logic for cluster head selection in WSN. Three fuzzy descriptors (residual energy, concentration, and centrality) are considered for cluster head selection. In this centralized approach [85], each sensor node forwards its clustering information to the base station and cluster heads are selected centrally. Kim et al. [86] have developed a distributed cluster head election approach based on fuzzy. In [86], cluster head (CH) is selected by using two fuzzy descriptors (residual energy and local distance). Moreover, Anno et al. [87] have desgined a cluster formation and cluster head selection mechanism in WSN. The fuzzy parameters, remaining battery power, number of neighbor nodes, distance from cluster centroid and network traffic, are employed in the process of cluster head selection. They have shown that remaining energy and number of neighbor nodes are more important parameters than the distance. Bagci et al. [89] have proposed a fuzzy unequal clustering algorithm which adjusts the cluster head radius by considering the distance to the base station and residual energy as fuzzy parameters. This approach solves the hotspot problem by decreasing the intra-cluster work of the sensor nodes which are closer to the base station (or have less residual energy). Ran et al. [88] have improved LEACH protocol using Fuzzy Logic by considering battery, distance and node density as fuzzy parameters. The above fuzzy based clustering algorithms are designed based on the LEACH protocol and considered residual energy as fuzzy parameter for the selection of cluster head. However, Lee *et al.* [17] have considered predicted remaining energy in a fuzzy logic based clustering approach. Also, the authors introduce a model to estimate the predicted energy consumption.

In [90], Ni *et al.* have proposed a cluster head selection approach based on fuzzy clustering and particle swarm optimization (PSO). The fuzzy clustering has been proposed for initial clustering of sensor nodes. In the improved PSO, the fitness function is designed by considering the energy consumption and distance factors (of sensor network). Neamatollahi *et al.* [80] have addressed the energy wastage overhead which is imposed from consecutive reclustering of nodes. In [80], a fuzzy-based hyper round policy sets the clustering task dynamically. A fuzzy inference system is used to determine reclustering time by considering two fuzzy descriptors: *energy budget of a node* and *distance to the sink*. Arjunan *et al.* [81] have proposed a fuzzy logic based Unequal Clustering and Ant Colony Optimization (ACO) based routing to eliminate hotspot problem and to extend lifetime of network. This hybrid protocol proposed for cluster head selection, inter-cluster routing and cluster maintenance. In the fuzzy inference system, residual energy, distance to BS, distance to its neighbors, node degree and node centrality are considered as input variables and probability of becoming cluster head and cluster size are considered as output variables. Collatta *et al.* [16] have proposed a fuzzy logic based mechanism to compute the sleeping time of sensor nodes in an industrial wireless sensor network to improve the energy efficiency. In [16], a particle swarm optimization (PSO) algorithm is presented to obtain the optimal values and parameters of the fuzzy logic controller. The mechanism defines the sleeping time of sensor nodes based on battery level and to the ratio of throughput to workload. Nayak *et al.* [82] have proposed an energy efficient clustering algorithm using fuzzy logic for WSNs. In [91], Yousaf *et al.* have proposed a fuzzy logic based power allocation scheme for a cooperative and delay constraint energy harvesting wireless sensor network.

In this thesis, fuzzy logic system based mechanisms are designed in Chapter 5 and Chapter 6. In Chapter 5, routing decisions are taken using a fuzzy logic system by considering network resources, such as residual energy, quality of link, available buffer size and distance. In Chapter 6, the proposed fuzzy logic system determines the expected duty cycle value by considering the residual energy, predicted harvesting energy and expected residual energy.

2.10 Reliability in Wireless Sensor Networks

Reliable data collection is a primary goal for various applications of wireless sensor network [13]. In WSNs, the sensor nodes collect information from the environment and relay to the sink or actor node using mutli-hop communication. The wireless channel can be affected by the adverse environmental conditions, such as fog, rainfall, wind, high temperature and movable obstacles. Thus, the quality of wireless link varies dynamically with the environmental conditions. Due to lossy nature of wireless communication, packets may be lost during the data transmission. Retransmission and redundancy are the two techniques that are used in WSNs to enhance the reliable data transmission. Both techniques can be performed based on either hop-by-hop or end-to-end. Mahmood *et al.* [13] have discussed several reliability protocols which are based on retransmission and redundancy.

The existing schemes (in [92, 93, 94, 95, 96]) provide the reliable data transmission based on the retransmission technique. Reliable Multi-Segment Transport (RMST) [92] is a transport layer protocol. It provides reliable data transmission using selective negative acknowledgments (NACKs). RMST performs loss detection and it has been developed for multimedia applications in WSNs. Iyer *et al.* have proposed a Sensor Transmission Control Protocol (STCP) in [93]. STCP uses ACK and NACK for retransmission of a lost packet. It is based on end-to-end retransmission mechanism in which a source node caches the packet until it gets an ACK from the sink. The key features offered by STCP are controlled variable reliability and congestion detection and avoidance. Wan *et al.* [94] have proposed Pump Slowly Fetch Quickly (PSFQ) protocol which is based on a hop-by-hop downstream reliability mechanism. It is mainly based on three operations (pump, fetch and report). It performs the loss detection and data recovery using retransmission. *Reliable Bursty Convergecast* (RBC) has been proposed by Zhang *et al.* [95] to improve channel utilization and to reduce ACK losses. RBS is a windowless block acknowledgment scheme and it considers adaptive timer mechanism in time based retransmissions to reduce end-to-end delay. Tunable Reliability with Congestion Control for Information Transport (TRCCIT) has been proposed by Shaikh *et al.* [96] to proved reliable data transmission for WSNs. It uses hybrid acknowledgment and retransmission timers to achieve reliable data transmission in dynamic network conditions. Further, STCP and TRCCIT protocols are suitable for continuous flow applications. RBC is developed for high-volume bursty traffic applications and RMST is for multimedia applications. However, these protocols [92, 93, 94, 95, 96] are based on retransmission based mechanisms which provide the reliability at high cost of communication overhead, energy consumption and memory utilization.

In the redundancy based retransmission approach, a lost packet is recovered through coding techniques. Redundancy approach improves both reliability and energy efficiency. In [97], an energy-efficient and reliable transport protocol has been proposed by Le *et al.* to balance energy consumption and reliability in WSNs. Park *et al.* [98] have proposed a downstream reliability mechanism which ensures the reliability from sink-to-sensors in wireless sensor networks. A network coding based adaptive redundant protocol has been proposed by Wu *et al.* [12] to determine the level of redundancy in a link to achieve high data reliability with low delay. Keller *et al.* [99] have proposed a data collection protocol called Sensecode to balance the energy efficiency and end-to-end packet error rate using network coding. Zhang *et al.* [100] have proposed a unequal redundancy level data collection scheme for a wireless sensor network to achieve high reliability. Basagni *et al.* [101] have considered mobile sink and delay issues. The authors have defined *Mixed Integer Linear Programming* analytical model to find routes to the sink node for maximizing the network lifetime.

In this thesis, energy efficient and reliable data gathering algorithms are designed for wireless sensor and actor networks. Primarily, reliable data collection depends on packet dropping rate. A packet may be dropped due to buffer overflow, unreliable link quality and low energy levels of a sensor node. In Chapter 3, the problem of packet loss due to buffer overflow has been considered. The problem of packet drops due to unreliable link quality is addressed in Chapter 4. In Chapter 5, packet loss has been reduced due to an efficient routing metric which is computed based on the link quality, available buffer and energy level. Further, Chapter 6 addresses network sustainability issue in energy harvesting sensor and actor networks. In this work, reliable data delivery is improved by achieving survivability of nodes till next recharge period.

2.11 Energy Harvesting Wireless Sensor Networks

Wireless Sensor Actor Networks (WSANs) remain operational for limited amount of time due to limited battery capacity of a sensor node [2]. A long operational lifetime of WSAN is required for applications, such as monitoring forest fires, nuclear plants, object tracking and military surveillance. Energy harvesting technology provides a long lasting lifetime in WSAN [102]. A node recharges battery in two ways: (i) through mobile charging vehicles and (ii) from the natural resources. In the first case, charging vehicles move close to the nodes and recharge the batteries via wireless energy transfer technology [37, 4, 35, 36]. In the second case, a node recharges its battery from the natural resources, such as solar light, thermal, wind and vibration [102, 18, 79].

The process of energy harvesting allows the sensors to collect the energy from the surroundings that include solar and kinetic energy. Later, this energy can be converted to electrical energy and saved for future purpose or consumed at that moment. In an energy harvesting technique, nodes' energy consumptions and harvesting energy predictions need to be analyzed to know the power consumption behavior of nodes. This allows to determine the next recharge cycle. Energy saving mechanisms are required because energy of a node is limited in between the recharge cycles. Also, the variation in the amount of harvested energy among different sensor nodes could lead to uneven distribution of residual energy.

Hence, energy balance is an important issue for designing routing protocols in harvesting sensor networks.

Some of the studies [37, 4, 36, 35] are based on mobile charger models. Zhao *et al.* [37] have studied joint design of energy replenishment and data gathering in WSNs using mobile collectors. However, they have considered energy consumption of data transmission only (but not consumption of data receiving and sensing). The recharging rates are also considered as constant instead of time-varying. Further, Guo *et al.* [4] have proposed a joint wireless energy replenishment and mobile data gathering framework by considering all types of sensor energy consumptions and time-varying nature of recharging time. In [4], a multifunctional SenCar is presented for charging sensor nodes via wireless energy transfer. Data packets are collected via multi-hop transmissions. Guo *et al.* [4] have proposed an anchor point selection algorithm to determine the sensor nodes for recharging and sequence of anchor points. Here, the anchor point represents position of a sensor node. Shu *et al.* [36] have proposed a joint energy replenishment and scheduling mechanism to maximize the lifetime for rechargeable sensor networks. Most of the previous works have not considered the moving energy consumption of the charging vehicle and charging capacity in their models. Wang *et al.*[35] have proposed a recharge scheduling problem under such important constraints. Further, a mathematical model is also proposed to calculate the minimum number of charging vehicles needed.

Tan *et al.* [18] have proposed a markov model to trace the energy harvesting process and discussed the rechargeable node's performance aspects, such as temporal death occurrence probability, probability density of the residual energy, stationary energy consumption, packet blocking probability and queue length distribution in data buffer. Zhang *et al.* [103] have addressed the problem of system design of energy harvesting devices in terms of energy and data buffer. Further, Misra *et al.* [104] have studied constrained relay node placement problem in an energy harvesting WSN to achieve connectivity and sustainability. Dynamic activation of sensor nodes to maximize system performance has been presented by Kar *et al.*[77]. A markovian model has been designed for calculating sensor discharge and recharge periods. Active time scheduling mechanisms are proposed for rechargeable sensor networks by Pryyma *et al.*[105]. Dynamic activation policies for event capture in rechargeable sensor network have been proposed by Ren *et al.*[106]. Rout *et al.* [79] have proposed a markov decision process based switching algorithm to achieve sustainable data collection in a rechargeable sensor network. In [79], a node recharges the battery from the sunlight.

In this thesis, energy harvesting nodes are considered in Chapter 6. The nodes harvest energy from the sunlight. Chapter 6 addresses the sustainability issue in energy harvesting sensor and actor networks. In Chapter 6, switching decisions are taken based on the duty cycle value to achieve the network sustainability. Further, coordinated duty cycle schedule has been considered to improve the reliable data delivery.

2.12 Summary

In this chapter, different types of network topologies in wireless sensor networks are discussed. A survey on energy efficiency and energy consumption model has been presented. Energy efficient techniques, such as mobile sink, duty cycle, network coding based communication have been discussed. An exhaustive suvey on markov decision process and fuzzy logic based approaches is performed. Further, reliable data transmission approaches for WSNs have been presented. Energy harvesting techniques in wireless sensor networks are discussed. In this thesis, both energy efficient and reliable data transmission approaches have been designed for wireless sensor and actor networks. Further, the work presented in this thesis has been compared with existing approaches in the literature. In the next chapter, an adaptive buffer management mechanism using MDP has been presented to address the problem of packet loss due to buffer overflow.

Chapter 3

Adaptive Buffering Algorithm using Markov Decision Process in Tree-based WSANs

WSANs are a group of sensors and actors that are geographically distributed and interconnected by wireless links. Usually, sensor nodes are static where as actors may move in the monitoring area with self-awareness to fulfill the tasks [2]. Nodes in immediate vicinity of actor (or sink) will be overburdened and the energy of the nodes will be depleted quickly while compared with other nodes in the network. This leads to network partitioning and an actor cannot receive data from the sensor nodes [5].

In a WSAN, evaluation of the traveling schedule of mobile actor is an important issue to reduce the dropping rate of packets. Further, buffer state (buffer occupancy) of a node changes dynamically in event-centric network. The buffer states of tiny sensors are limited due to which some of the packets are dropped. The retransmission of the dropped packets leads to more delay and more energy consumption in the network. Therefore, there is a need to estimate the bounds on the buffer occupancy and energy consumption in a tree-based WSAN. Nodes in the network may get high traffic at a particular time. It will be beneficial to drop the packets (which are generated at nearest sensor nodes) to increase the available buffer space. However, the number of retransmissions depends on the buffer occupancy (i.e. at which packets are to be dropped and number of packets to be dropped). Therefore, an efficient mechanism is required to find an accurate buffer state to drop minimum number of packets so that average end-to-end delay and energy consumptions will be minimized.

In this chapter, a Markov Decision Process (MDP) based buffer management mechanism has been designed with varying available buffer levels of sensor nodes. The novelty of the approach lies on applying MDP in a tree-based sensor (mobile) actor network for efficient buffer management. Further, energy efficiency of the system is enhanced due to efficient buffer management in dynamic traffic conditions. A node determines the optimal policy using the proposed algorithm to reduce the number of retransmissions, there by, minimizing delay and energy consumption. In this work, energy consumptions have been estimated using an event-centric network traffic. A mathematical model for actor has been formulated to estimate bound on buffer occupancy and energy consumption.
The major contributions of this chapter are as follows:

- Analysis of energy consumptions in tree-based wireless sensor network while considering an event-centric traffic.
- Design of a *Markov Decision Process based buffer management mechanism* for efficient data gathering by reducing the end-to-end delay and improving the energy efficiency in a tree-based WSAN.
- Analysis of bounds on buffer occupancy and energy consumption in tree-based sensor actor networks.

The rest of the chapter is organized as follows. Initially, motivation and problem formulation is discussed in section 3.1. The proposed network model is introduced in section 3.2.1. In section 3.2.2, energy utilization model has been presented. A mathematical model has been formulated in section 3.2.3 for actor to formulate bounds on buffer occupancy and energy consumption. In section 3.3, a *Markov Decision Process* based buffer management mechanism has been presented to reduce the delivery delay and to improve energy efficiency in tree-based sensor network. Performance evaluation has been done through simulation studies in section 3.4. The concluding remarks are given in section 3.5.

3.1 Motivation and Problem Formulation

Improvement in energy efficiency can be achieved using mobile actor where actor moves across the network to gather data from sensor nodes [6]. Resource constrained nodes store the packets till a mobile actor collects the data. Therefore, packet dropping occurs when nodes buffers are full (Packet dropping due to channel noise is not considered in this work). The dropped packets are retransmitted by the source node which lead to more energy consumption and delay in data delivery. Delivery delay reduces by reducing the number of retransmissions. Nodes in the network may get high traffic at any time, it would be beneficial to drop the packets generated from nearest source node to improve available buffer space. In our work, an adaptive buffer management technique using *Markov Decision Process* is proposed to determine the buffer states at which the packets will be dropped to reduce average delivery delay and energy consumption in the network. Different MDP policies are formulated to achieve lower delivery delay. The focus of this work is to estimate the energy utilization in tree-based WSAN. Further, the work focuses on formulating a mathematical model for mobile actor and analyzing the upper bound on buffer occupancy and energy consumption.

3.2 Estimation of Energy Consumptions and Network Delay in a Tree-based Sensor Network with a Mobile Actor

In this section, a network model is introduced. An energy consumption model has been presented to estimate the energy utilization for the network model. Further, a mathematical model is formulated to estimate upper bound on the buffer occupancy and energy consumption in the sensor tree. Buffer of each node is modeled using $M/M/1/b$ queuing model [107] to analyze the end-to-end delay in a sensor tree. Energy consumption and delay for retransmission of dropped packets have been estimated.

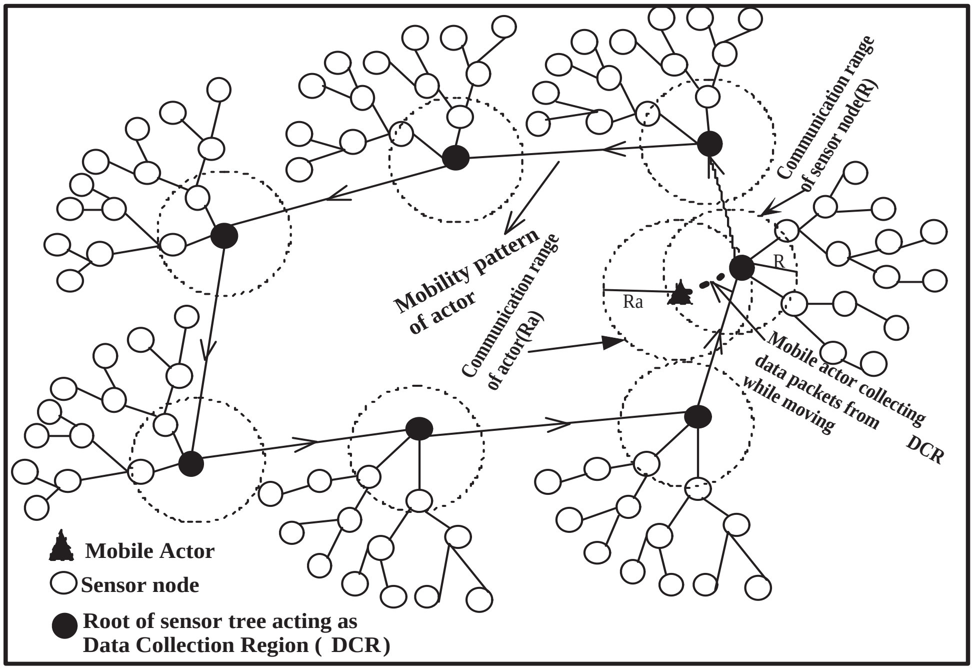


Figure 3.1: Tree-based Sensor Network with Mobile Actor

3.2.1 Network Model and Preliminaries

A tree-based wireless sensor network is considered with a mobile actor (shown in Fig. 3.1). Tree based topology avoids flooding and data can be sent using unicast instead of broadcast. The topology saves energy than flat topology as flooding is not necessary for data communication (for establishing the data forwarding paths). Sensor nodes get more opportunity to aggregate or pre-process the data in tree-based topologies. In a tree-based topology, scalability is limited up to a certain number of depths of the tree. The topology works well for medium sized networks, but has scalability limitations that degrade performance for larger sensor network. However, negative effects of dense deployment may be reduced using the transmission and duty cycle scheduling of the sensor nodes.

Sensor trees are disconnected and mobile actor (A_m) collects data (shown in Fig. 3.1). Data acquisition is done through the mobile actor. A sensor tree (T_i) is modeled as $T_i = (V, E)$, where $V = \{v_0, v_1, ..., v_n\}$ denotes set of sensor nodes, v_0 is a root node and E represents the set of the communication links. Each sensor node has maximum of $\bar{n}$ children in a sensor tree. In the proposed network model, total number of disconnected sensor trees is assumed as m and the communication ranges of a sensor node and an actor node are R and R_a, respectively. Communication range of root node in a sensor tree is defined as *Data Collection Region* (DCR). In a sensor tree, sensors (are stationary) store their generated data (by monitoring the events) and forward to their parent node. Mobile actor collects the data from all data collection regions (from root node of every sensor tree) by traveling around the network (refer Fig. 3.1). Contact time (t_i) is defined as the amount of time the actor moves in the data collection region of the sensor tree (T_i). Depending on the contact time (not constant), the mobile actor adjust its speed to collect maximum number of packets which are available at a sensor tree. In our work, due to limited transmission range of sensor nodes and limited velocity of mobile actor, the system is considered as collection of sensor trees. In the proposed network model, when mobile actor enter into the data collection region of a tree, every sensor node in same tree relays its data to the respective parent nodes. Further, mobile actor informs the contact time to nodes in the sensor tree to adjust their data transfer rate. Sensor nodes manage their buffers to reduce the dropping rate of packets based on the next visiting time of the actor.

In this chapter, a packet at node-A is called as lower priority if it is generated by nearest source node-S. Sensor nodes store the data till a mobile actor collects the data. Nodes in the network may get high traffic at any time. Packet dropping occurs when node's buffer is full. The dropped packets are retransmitted by the source node. If the dropped packets are not lower priority packets, then consumes more energy and cause delay in a data delivery. It would be beneficial to drop the lower priority packets to improve available buffer space so that higher priority packets should not be dropped. The notations are mentioned in Table 3.1.

3.2.2 Estimation of Energy Consumptions

In this section, an energy consumption model is presented to estimate energy utilization of a sensor tree in the proposed network architecture (see Fig. 3.1).

Energy consumption in the proposed architecture has been estimated by using the model described in [7, 8]. E_{sense}, E_{rx} and E_{tx} are the parameters used to quantify energy consumptions by a sensor node in sensing, receiving and transmitting data over a

Table 3.1: Summary of the Used Notations

Symbol	Description
T_i	Sensor tree-i
$T_i = (V, E)$	V is set of sensor nodes and E is set of edges
A_m	Mobile actor
m	Number of sensor trees
n	Number of sensor nodes in a sensor tree
$\bar{n}$	Maximum number of children of a sensor node
R	Communication range of a sensor node
R_a	Communication range of an actor node
b	Sensor node's buffer capacity
DCR	Data Collection Region
t_i	Contact time of a sensor tree (T_i)
α_{11}	Energy consumption per bit by the transmitter electronics
α_2	Energy consumption per bit in the transmit op-amp
α_{12}	Energy consumption per bit by the receiver electronics
α_3	Energy consumption for sensing a bit
α_1	$\alpha_{11} + \alpha_{12}$
$\hat{n}$	Path loss component
l	Size of generated data by a sensor node per event
β	Average rate of occurrence of events
R_b	Radius of a bottleneck zone
H	Radius of a sensor tree
k	Maximum number of hops in a bottleneck zone
h	Height of a sensor tree (in terms of hops)
ρ_t	Node density in a sensor tree
ΔC_{oh}	Energy consumption by sensor nodes for the overhearing of data
A_i	Area of a sensor tree T_i
Δt_i	Time between the last visiting time and current visiting time of the mobile actor at sensor tree (T_i)
T_d^i	Upper bound on Δt_i
$ec(v_j)$	Energy consumption node v_j for sending and receiving data in one unit of time
$eg(v_j)$	Energy consumption node v_j for sensing operation in one unit of time
$E(v_j)$	Residual energy of node v_j
f_{cj}	Size of incoming data from node v_c to the node v_j

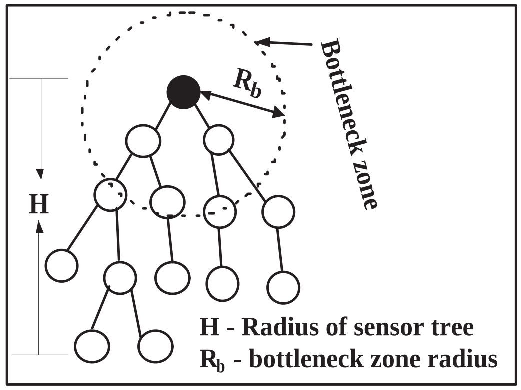


Figure 3.2: Bottleneck zone in a sensor tree

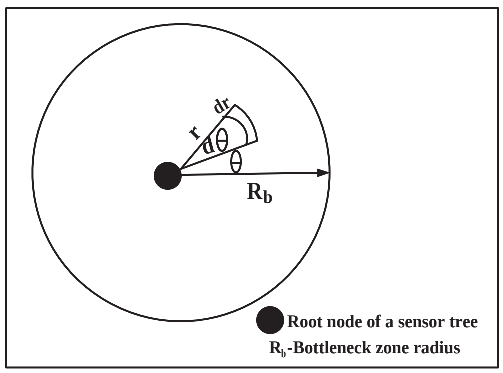


Figure 3.3: A bottleneck zone of radius R_b at root of sensor tree

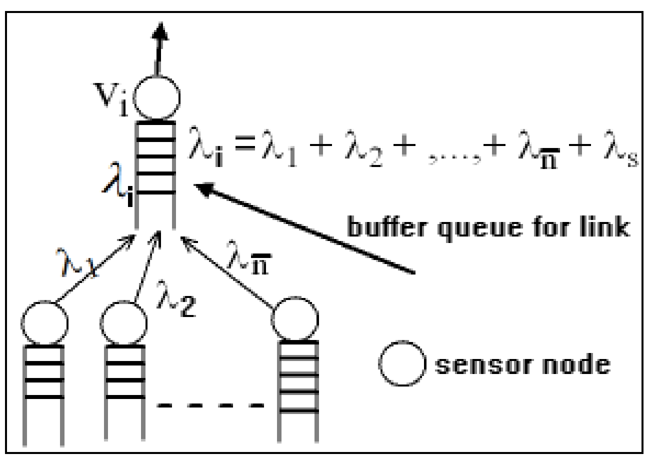


Figure 3.4: Buffering model with $\bar{n}$ children

distance $\hat{d}$, respectively. The energy consumptions are given by $E_{sense} = \alpha_3$, $E_{rx} = \alpha_{12}$, $E_{tx} = \alpha_{11} + \alpha_2 \hat{d}^{\hat{n}}$, where, α_{11} represents energy consumption per bit by the transmitter electronics, α_2 denotes the energy dissipated in the transmit op-amp, α_{12} represents the energy consumption per bit by the receiver electronics, α_3 denotes the energy cost for sensing a bit and $\hat{n}$ represents the path loss component. The energy consumptions in our model are estimated by considering an event-centric application [8]. Further, l is the size of generated data by each sensor node per event and β is the average rate at which events occur per unit time [32]. Therefore, energy consumption for sensing is represented as $\alpha_3 lt\beta$ in time t.

Nodes around the root in a sensor tree form a bottleneck zone [5] (refer Fig. 3.2). R_b is the radius of bottleneck zone and H is the radius (longest distance from root node to any leaf node in a sensor tree) of a sensor tree . Maximum number of hops presented in a bottleneck zone is k and height (h) of a sensor tree is defined as the longest path (number of hops) from root node to any leaf node in a sensor tree. Energy consumption of the sensor tree (T_i) till time t is given by EC_i

$$EC_i = (\bar{n}^{h+1} - 1)\alpha_3 lt\beta + \sum_{j=1}^{(\bar{n}^{h+1}-\bar{n}^{k+1})lt\beta} C_j + \int_0^{R_b}\int_0^{2\pi} p(x_1)lt\beta\rho_t r d\theta dr + \int_{R_b}^{H}\int_0^{2\pi} p(x_2)lt\beta\rho_t r d\theta dr + \Delta C_{oh} \quad (3.1)$$

Where, C_j is the energy consumption of a node inside the bottleneck zone to relay j^{th}-bit from outside of the bottleneck zone to the root node [5, 7] and is estimated as follows

$$C_j \geq \alpha_1 \frac{\hat{n}}{\hat{n}-1}\frac{R_b}{d_m} \quad (3.2)$$

According to ref.[7], d_m is the length of one hop and is as follows

$$d_m = \sqrt[\hat{n}]{\frac{\alpha_1}{\alpha_2(\hat{n}-1)}} \quad (3.3)$$

where, $\alpha_1 = \alpha_{11} + \alpha_{12}$.

Node density in the sensor tree is $\rho_t = \frac{\bar{n}^{h+1}}{A_i}$, where, A_i is the area of sensor tree (T_i). The term $\rho_t r d\theta dr$ is the number of nodes in differential area (shown in Fig. 3.3) [8]. $p(x_1)$ is the energy consumption of a node (in bottleneck zone which is x_1 distance away from the root node) to transmit a bit [7], i.e.

$$p(x_1) \geq \alpha_1 \frac{\hat{n}}{\hat{n}-1} \frac{x_1}{d_m} - \alpha_{12} \tag{3.4}$$

where, $p(x_2)$ is the energy consumption (of a node in $(H - R_b)$ region which is x_2 distance away from bottleneck zone) to transmit a bit [7] is as follows

$$p(x_2) \geq \alpha_1 \frac{\hat{n}}{\hat{n}-1} \frac{x_2}{d_m} - \alpha_{12} \tag{3.5}$$

The first term in *Equation* (3.1) is the energy consumption of the sensor tree due to sensing operation till time t. The second term in *Equation* (3.1) represents the energy consumption of nodes in bottleneck zone to forward the data which is generated in $(H - R_b)$ zone. In *Equation* (3.1), third term is the energy spent by bottleneck zone nodes to forward its own generated data to root node. The fourth term in *Equation* (3.1) is energy spent by $(H - R_b)$ zone nodes to forward its own generated data upto bottleneck zone. In *Equation* (3.1), the last term (ΔC_{oh}) is the energy spent by the sensor nodes in the sensor tree due to overhearing in wireless transmission.

3.2.3 Upper Bound on Buffer Occupancy and Energy Consumption

In this section, a mathematical model is presented for an actor to estimate upper bound on buffer occupancy and energy consumption in the proposed architecture (as shown in Fig. 3.1).

The mobile actor collects data from root node of every sensor tree (when actor enters into data collection region) while moving. The velocity of the actor in data collection region (DCR) depends on contact time of respective sensor tree. Let, Δt_i is the time between last visiting time and current visiting time of the mobile actor at sensor tree (T_i). Packet dropping rate increased more at bottleneck zone nodes if there is no upper bound on Δt_i. Therefore, maximum time bound (T_d^i) is defined as an upper bound on Δt_i and is minimize the dropping rate of the packets when mobile actor is not visiting the sensor tree (T_i) within the stipulated time. Hence, $\Delta t_i \leq T_d^i$. It is estimated as follows

$$lT_d^i\beta(\bar{n}^{h+1} - 1) \leq (\bar{n}^{k+1} - 1)b$$

$$T_d^i \leq \frac{b(\bar{n}^{k+1} - 1)}{l\beta(\bar{n}^{h+1} - 1)} \tag{3.6}$$

where, b is buffer capacity (in terms of bits) of a sensor node. *Equation* (3.6) denotes the maximum time to full the buffers of bottleneck zone nodes and it has been used in *Equation* (3.9) to reduce the dropping rate of packets at bottleneck zone nodes.

The objective of the model is to estimate the bound on buffer occupancy and energy consumption subject to the following constraints (i) Total energy consumption of a node in sensor tree should not exceed its residual (current) energy (ii) Minimizing the dropping rate of packets due to buffer overflow (iii) Every node in the tree should satisfy the data flow constraint i.e the amount of data a node is sending is equal to the sum of the receiving data (from its children) and generating data (by sensing operation). Mobile actor contact time at sensor tree T_i is denoted as t_i. The term $ec(v_j)$ is the energy consumption of node v_j for sending and receiving data in one unit of time. The term $eg(v_j)$ is the energy consumption for sensing operation in one unit of time. Mobile actor collects the maximum data available at sensor nodes in a sensor tree by estimating the optimal contact times and the time spent by the actor traveling from one sensor tree to the next one has not been considered in this work.

$$\textit{Maximize } (T = \sum_{i=1}^{m} t_i) \tag{3.7}$$

subject to:

$$ec(v_j)t_i + (eg(v_j)(\Delta t_i + t_i)) \leq E(v_j), \quad \text{for } j \in n \text{ and } i \in m \tag{3.8}$$

$$T - t_i \leq T_d^i, \text{for } i \in m \tag{3.9}$$

$$\sum_{c=1}^{\bar{n}} f_{cj} + (\Delta t_i + t_i)l\beta = f_{jp}, \text{for } j \in n \text{ and } i \in m \tag{3.10}$$

where, m is the number of sensor trees in the proposed network model (refer section 3.2.1) and n is the number of sensor nodes in each sensor tree. The actor collects the values of the required parameters (as mentioned in the above *Equation* (3.8), *Equation* (3.9) and *Equation* (3.10)) when it visits the sensor trees.

The first term in *Equation* (3.8) is the energy consumption of the node v_j for transmission and reception operations during the contact time (t_i) of an actor at sensor tree (T_i). The second term in *Equation* (3.8) denotes the energy consumption of node v_j for sensing operation in time $(\Delta t_i + t_i)$. The sum of these two consumptions is less than or equal to the residual (current) energy ($E(v_j)$) of node v_j. *Equation* (3.9) depicts that the difference of any two consecutive visiting times of mobile actor at data collection region (DCR) of the sensor tree (T_i) will have maximum time bound T_d^i (estimated using *Equation* (3.6)) which minimizes the packets dropping rate. In *Equation* (3.10), f_{cj} represents the total incoming data to the node v_j from its children ($\bar{n}$ is the maximum number of children), $(\Delta t_i + t_i)l\beta$ is data generated in time ($\Delta t_i + t_i$) and f_{jp} is total size of data flow from v_j to its parent or root node (v_p). Moreover, *Equation* (3.10) depicts the data flow constraint.

The above optimization problem is modeled using linear programming (*Equation* (3.8), *Equation* (3.9) and *Equation* (3.10)) and can be solved in polynomial time. Based on the optimal contact time (t_i) which is computed from optimization problem (refer *Equation* (3.7)), actor calculates its accurate velocity in the data collection region of sensor tree (T_i). The constraints (in *Equation* (3.9) and *Equation* (3.10)) ensures the actor to collect maximum number of packets which are available at nodes. Further, the actor needs to visit within the stipulated (bound) time (as given in *Equation* (3.9)). The optimal states are identified where end-to-end delay and energy consumptions will be minimized.

3.2.4 Analysis of End-to-End Delay using Queuing Theory

In this section, buffer of a node is modeled using $M/M/1/b$ queuing system [107] to estimate the average end-to-end path delay of a packet in a sensor tree.

In a sensor tree, nodes generate data (by sensing operation) and forward to its parent node. Assume that the arrival rate of data flow to a node follows poisson distribution. In Fig. 3.4, $\lambda_1, \lambda_2, ...\&\lambda_{\bar{n}}$ (where, $\bar{n}$ is maximum number of children of node v_i) are poisson flows generated by the children of node v_i. Using superposition property of poisson processes [108], the combined stream arrives at a parent node v_i is a poisson process with rate λ_i where, $\lambda_i = \lambda_1 + \lambda_2 .... + \lambda_{\bar{n}} + \lambda_s$ and λ_s is sensing data rate of node v_i. $M/M/1/b$ queue is the link queue for node v_i, with arrival parameter λ_i and service rate μ_i, where μ_i=$\frac{\text{Bandwidth of link}}{\text{Packet size}}$. According to $M/M/1/b$ queuing model [107], the expected waiting time in a node v_i is given by $Ed(v_i)$

$$Ed(v_i) = \frac{\hat{\rho}_i[1 - (b+1)\hat{\rho}_i{}^b + b\hat{\rho}_i{}^{b+1}]}{\lambda_i(1 - P_b)(1 - \hat{\rho}_i)(1 - \hat{\rho}_i{}^{b+1})} \tag{3.11}$$

and $\hat{\rho}_i$ is the expected number of packets at node v_i ($\hat{\rho}_i = \frac{\lambda_i}{\mu_i}$), P_n is the steady state probability that a node v_i has n packets and is as follows

$$P_n = \frac{\hat{\rho}_i{}^n(1 - \hat{\rho}_i)}{1 - \hat{\rho}_i{}^{b+1}}$$

Consider a path where a node at level k_l generates a packet and sends to its parent. After queuing delay at the parent node, the parent forwards the packet to its parent. Finally, the packet is relayed to level-1 node i.e root node. End-to-end path delay defined as the time required to reach the root. End-to-end path delay of a packet is formed by sum of queuing delay of k_l levels nodes and propagation delay of k_l hops. End-to-end path delay of a packet (in sensor tree with level-k_l) is given by

$$E(T) = \sum_{i=1}^{k_l} Ed(v_i) + \sum_{i=k_l}^{1} (PD_i^{i-1}) \tag{3.12}$$

where PD_i^{i-1} is the propagation delay of a packet from level-i to level-$(i-1)$.

3.2.5 Estimation of Energy Consumption and End-to-End Delay for Retransmission of Dropped Packets

In a sensor tree, additional delay and energy consumption will be incurred for the dropped packets. If a mobile actor visits a data collection region late by δt time, then the queuing system is discussed here by considering two factors, (a) Nodes with infinite buffer (memory) (b) Nodes with finite buffer (memory). In the first case, packet end-to-end delay is directly proportional to the delayed visit by the actor. Retransmissions of packets are not needed because there is no packet dropping in the first case. However, in the second case, nodes have finite buffer (b is capacity of buffer), packets generated in δt time are stored in buffer if the memory is not filled. If the buffer is full, then the packets will be dropped by the nodes. In this case, source needs to retransmit (the packets) which leads to more delay in data delivery and more energy consumption.

Consider a tree where a node v_i at level-k_l drops the packets due to unavailability of actor in the data collection region or due to empty buffer space. Assume, the volume of dropping packets by node v_i is dp_{k_l}. Node v_i receives the packets from nodes belong to lower levels in the path. Therefore, packet size can be written as, dp_{k_l}= $dp_{k_l+1} + dp_{k_l+2} + ...dp_h$ where h is height of sensor tree and dp_{k_l+1} is size of the packets generated at level-($k_l + 1$). Energy consumption for retransmission of dropped packets at node v_i is given by Ex_c

$$Ex_c = \sum_{j=1}^{h-k_l} j(\alpha_1 + \alpha_2 \hat{d}^{\hat{n}})dp_{k_l+j} \tag{3.13}$$

Consider, a packet dp_{k_l} is dropped by level-k_l node and it is generated at level-j $(j > k_l)$, then delay for this packet is given by Ex_d

$$Ex_d = \sum_{i=0}^{j-k_l-1} (Ed(v_{j-i}) + PD_{j-i}^{j-i-1}) + Ed(v_{k_l}) \tag{3.14}$$

Energy consumption and delay reduce by dropping the packets that are generated by the nearest level nodes (i.e while dropping the packets at level-k, the packets which are generated at nearest level will be dropped first).

3.3 Markov Decision Process based buffer management mechanism

In this section, MDP based buffer management mechanism is proposed to reduce the average end-to-end delay and to improve energy efficiency in tree-based *Wireless Sensor and Actor Networks*.

In the proposed network model (refer Fig. 3.1), when the actor enters into the data collection region of a sensor tree, every node transmits the buffered data to the parent node. Further, the actor collects the data from the root node during its travel through data collection region of the sensor tree. The packets are dropped if sufficient buffer space is not available to store the data. Then, the source node has to retransmit (the dropped packets) which leads to increase in average end-to-end delay and energy consumption (for retransmissions of dropped packets). Hence, we need to determine the buffer state at which packets need to be dropped so that average delivery delay and energy consumption for retransmissions of dropped packets are minimized. To address the above problems a *Markov Decision Process* based buffer management is presented in the next section.

3.3.1 MDP Based Buffer Management Algorithm

In this section, MDP based buffer management mechanism (*Algorithm* 3.1) is presented. After construction of sensor trees (line 1-3, *Algorithm* 3.1), every node in the network finds the optimal policy (line 4-6, *Algorithm* 3.1) by running the MDP_BUFFER () algorithm (*Algorithm* 3.2). In *Algorithm* 3.2 (refer line 1-4), every node calculates the reward for all policies (which are defined in the Table 3.4) using the *Equation* (3.17) and chooses the optimal reward policy. Further, a node determines the buffer state (B) and the number of packets to be dropped (D) from chosen policy (refer line 5, *Algorithm* 3.2). When node's buffer state is reached to buffer state of selected policy (line 8, *Algorithm* 3.1) then the node drops the lower priority packets (those packets that are generated by nearest level nodes to the root in a sensor tree). The detailed explanation of the proposed mechanism is given in section 3.3.2.

Algorithm 3.1 MDP_Buffer_Management_Mechanism (G)

Input: G is a graph with m disconnected components

```
1:  for i = 1 to m do
2:      For each disconnected component i, sensor tree (T_i) is constructed using breadth-first search algorithm
3:  end for
4:  for i = 1 to m do
5:      for j = i to n do
6:          Node (v_ij) calls the MDP_BUFFER () algorithm (Algorithm 3.2) to choose the optimal policy and nodes finds the buffer state (B) and the number of packets (D%) to be drop.
7:          Node calculates the buffer state based on its available buffer
8:          if buffer_state(v_ij) == B then
9:              Node (v_ij) drops the D% of lower priority packets
10:         end if
11:     end for
12: end for
13: go to 4
```

3.3.2 MDP Based Buffer Management Policies

Markov process helps to model optimal policy by comparing the original condition of wireless sensor network with improved condition. In *Markov Theory*, the future state of the sensor node depends upon the current state, rather than the past state [76].

Let, the set $\{F_0,\ F_1,\ F_2,...\}$ represents the number of available buffer units a sensor node has at $\{0^{th},\ 1^{st},\ 2^{nd},...\}$ time intervals. The total available buffer (free buffer) of the node is categorized into 4 levels and each level is considered as one state (refer Fig. 3.5). State B_1 will have maximum available buffer and state B_4 will have minimum available buffer. Initially, a node starts with maximum available buffer. $P\{F_{t+1} = j|F_t = i\}$ is the probability that a node has i units of available buffer at time t and it has j units of available buffer after time $(t + 1)$. Further, $\{b_0,\ b_1,\ b_2...\}$ are the number of buffer units occupied during $\{0^{th},\ 1^{st},\ 2^{nd},...\}$ time intervals and so on. Therefore, the random variable b_t is the units of buffer occupied in time interval t. Assume, b_t follows a Poisson distribution with mean $(\hat{\lambda})$ one unit of available buffer. Thus, the over all process can be treated as an *Markov Decision Process*. The probability of $\hat{k}$ number of buffer units occupied during the

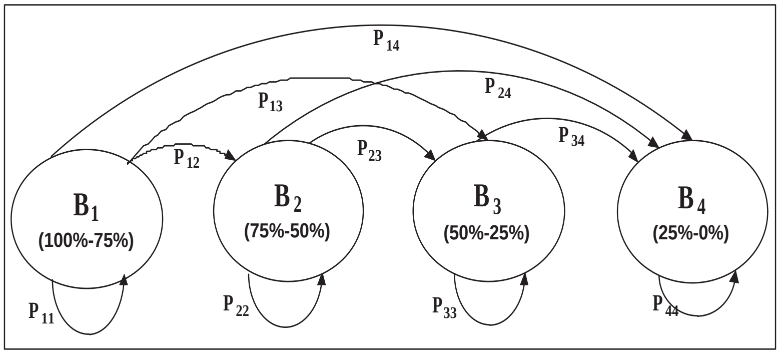


Figure 3.5: State transition diagram with different buffer states

interval $(t+1)$ is given by

$$f(b_{t+1} = \hat{k}) = (\hat{\lambda}^{\hat{k}} e^{-\hat{\lambda}})/\hat{k}! \tag{3.15}$$

Thus, F_t (for $t = 0, 1, 2, ...$) is a stochastic process, the possible states of the process are the integers $4, 3, 2, 1$ representing the possible buffer (available) units at the end of a time interval. The value of b_{t+1} is calculated for the transition from F_t to F_{t+1} using the following:

$$F_{t+1} = \begin{cases} max\{4 - b_{t+1}, 1\}, & \text{if } F_t = 4, \\ max\{F_t - b_{t+1}, 1\}, & \text{if } F_t \leq 3, \end{cases}$$

Using the value of b_{t+1}, the probability P_{ij} for transition from F_t to F_{t+1} is calculated by *Equation* (3.15). Fig. 3.5 shows the state transition diagram where B_1, B_2, B_3 and B_4 are the states of the node and Table 3.2(a) shows the initial transition matrix for state transition diagram where P_{ij} is the probability of changing the state from B_i to B_j. In state B_1, a sensor node operates with maximum available buffer (4 units) where the probability of buffer overflow is minimum. In states B_2 and B_3, a node operates with 3 units and 2 unites of available buffer, respectively. In state B_4, a node operates with minimum available buffer (1 unit) where the probability of buffer overflow is maximum. If available buffer (memory) is more (in terms of units) then, the possibility of dropping of packets reduce. Hence, delivery delay of packets will be less. Therefore, delivery delay of a packet is inversely

Table 3.2: Initial transition matrix and actions with relevant states

(a) Initial Transition Matrix

P_{ij}	B_1	B_2	B_3	B_4
B_1	0.368	0.368	0.184	0.08
B_2	0	0.368	0.368	0.264
B_3	0	0	0.368	0.632
B_4	0	0	0	1

(b) Decisions and Actions

Decision	Action	Relevant States
1	Do not drop the packets	B_1,B_2
2	Drop the packets	B_3,B_4

proportional to the available buffer space of the node as given in following equation,

$$\text{Delivery delay} = \frac{\tau}{\text{Available buffer unites}} \tag{3.16}$$

where, τ is a positive constant. The expected delivery delay is considered as reward of the MDP policy and is estimated in different buffer states by using the *Equation* (3.16) (as shown in Table 3.3).

Algorithm 3.2 MDP_BUFFER ()

1: **for** All MDP buffer management mechanism policies **do**
2: Find steady state probabilities Π_1, Π_2, Π_3,...,Π_{n_s} where n_s is the number of states (probability values will be calculated by transition probability matrix of each policy)
3: Reward = $\Pi_1 R_1$ + $\Pi_2 R_2$ + $\Pi_3 R_3$ +,...,+ $\Pi_{n_s} R_{n_s}$ (refer *Equation* (3.17))
4: **end for**
5: Select the optimal policy based on the rewards and returns the buffer state (B) and number of packets (D) to be dropped

In this work, two decisions are considered for MDP policy (refer Table 3.2(b)). *Decision 1* is *Do not drop packets* , where a node will not drop the packets from the buffer. *Decision 2* is *Drop the packets*, where a node drops the packets (low priority packets) from the buffer. Table 3.4 shows the policies and decisions for each state. For simplicity, two policies (P_2 and P_6) have been considered in simulation (refer section 3.4.1). In policy P_2, if the node in state B_3, B_4 then, *drop the 10% packets* (lower priority packets) from the

Table 3.3: Buffer availability and expected delay

State	Available buffer (in units)	Expected delay due to dropping of packets
B_1	4	0.25 τ
B_2	3	0.33 τ
B_3	2	0.5 τ
B_4	1	τ

Table 3.4: Dropping Policies

Policy	Verbal Description	D1	D2	D3	D4
P_1	Drop 5% packets at state B_3, B_4	1	1	2	2
P_2	Drop 10% packets at state B_3, B_4	1	1	2	2
P_3	Drop 15% packets at state B_3, B_4	1	1	2	2
P_4	Drop 5% packets at state B_4	1	1	1	2
P_5	Drop 10% packets at state B_4	1	1	1	2
P_6	Drop 15% packets at state B_4	1	1	1	2

Table 3.5: The Transition matrix for policy P_2 and P_6

(a) The Transition matrix for policy P_2

P_{ij}	B_1	B_2	B_3	B_4
B_1	0.368	0.368	0.184	0.08
B_2	0	0.368	0.368	0.264
B_3	0	0.4	0.6	0
B_4	0	0	0.4	0.6

(b) The Transition matrix for policy P_6

P_{ij}	B_1	B_2	B_3	B_4
B_1	0.368	0.368	0.184	0.08
B_2	0	0.368	0.368	0.264
B_3	0	0	0.368	0.632
B_4	0	0	0.6	0.4

Table 3.6: Steady state probabilities and rewards of different policies

Policy	Π_1	Π_2	Π_3	Π_4	Reward
P_1	0	0.18248	0.57664	0.24087	0.34175
P_2	0	0.30864	0.48765	0.20370	0.54914
P_3	0	0.40107	0.42246	0.17647	0.519
P_4	0	0	0.24038	0.75962	0.31077
P_5	0	0	0.3870	0.6124	0.80615
P_6	0	0	0.48701	0.51299	0.7555

buffer and the decision for states B_1, B_2 is *Do not Drop* (refer policy P_2 in Table 3.4). The transition matrix for policy P_2 (refer Table 3.5(a)) is calculated as follows

In state B_3, the available buffer space for a node is 50%-25% (refer Fig. 3.5). In state B_3, a node drops 10% packets (policy P_2) to increase the available buffer space. A node may stay in the same state (B_3) or it may change its state to B_2. In the first case, the node stays in the same state (B_3) if it has 25%-40% available buffer space. Therefore, the probability of staying in the same state is $15/25 = 0.6$ (refer Table 3.5(a)). In the second case, the node will switch from state B_3 to state B_2 if it has 40%-50% available buffer space. Therefore, the probability of transition from state B_3 to state B_2 is $10/25 = 0.4$ (refer Table 3.5(a)). Similarly, transition probability values for state B_4 are calculated.

In policy P_6, if the node in state B_4, then *drop the 15% packets* (lower priority packets) from the buffer and the decision for states B_1, B_2, B_3 is *Do not Drop* (refer policy P_6 in Table 3.4). Table 3.5(b) shows the transition matrix for policy P_6 (which is calculated similarly as Table 3.5(a)). Expected delivery delay (reward) is calculated by using the equation as given in below:

$$\sum_{k=1}^{n_s} [\Pi_k R_k] \tag{3.17}$$

where, Π_k is the steady state probability and it is calculated using transition probability matrix (Table 3.5(a) for policy P_2 and Table 3.5(b) for policy P_6). R_k is the expected delivery delay (refer Table 3.3) of a packet at state B_k and n_s is number of states.

Steady state probabilities $(\Pi_1, \Pi_2, \Pi_3$ and $\Pi_4)$ for policy P_2 is calculated by solving the steady state equations (*Equation* (3.18)). Steady state equations are constructed using the transition probability matrix (Table 3.5(a)) as shown below

$$\left.\begin{aligned} \Pi_1 &= 0.368\Pi_1 \\ \Pi_2 &= 0.368\Pi_1 + 0.368\Pi_2 + 0.4\Pi_3 \\ \Pi_3 &= 0.184\Pi_1 + 0.368\Pi_2 + 0.6\Pi_3 + 0.4\Pi_4 \\ \Pi_4 &= 0.08\Pi_1 + 0.264\Pi_2 + 0.6\Pi_4 \\ \Pi_1 + \Pi_2 &+ \Pi_3 + \Pi_4 = 1 \end{aligned}\right\} \tag{3.18}$$

Reward for policy P_2 is estimated as follows

Reward $(P_2) = \Pi_1 R_1 + \Pi_2 R_2 + \Pi_3 R_3 + \Pi_4 R_4$

$= 0 * 0.25 + 0.30864 * 0.33 + 0.48765 * 0.5 + 0.20370 * 1 = 0.54914$

Similarly, steady state probabilities and rewards (expected delivery delays) are calculated for policies P_1, P_3, P_4, P_5 and P_6 and presented in Table 3.6. It is observed from Table 3.6 that policy P_4 is giving optimal reward.

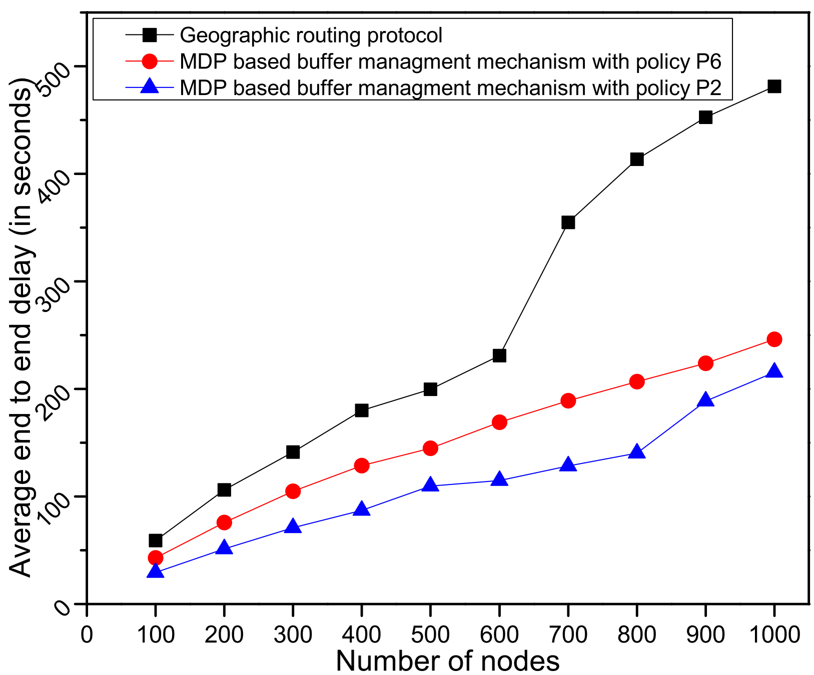


Figure 3.6: Average end-to-end delay of a packet in a sensor tree ($\lambda = 10, \mu = 15$, buffer size = 100 packets)

3.4 Performance Evaluation

The proposed adaptive buffer management mechanism using MDP has been simulated using *Network Simulator-3* [9]. A CSMA/CA based medium access control protocol (IEEE 802.11 standard) is used in the simulation [109]. The performance metrics such as average end-to-end delay, energy consumptions and lifetime are taken to compare proposed MDP based adaptive buffer management mechanism policies and geographic routing protocol [10]. In the geographic routing protocol, a source node sends the data to the destination using the location information. It does not require flooding and hence, reduces the control overhead. In our work, a tree-based sensor network is considered where each node sends the data to its parent node and it does not require flooding. Further, dedicated routing tables and flooding are not required in the geographic routing protocol and the proposed approach.

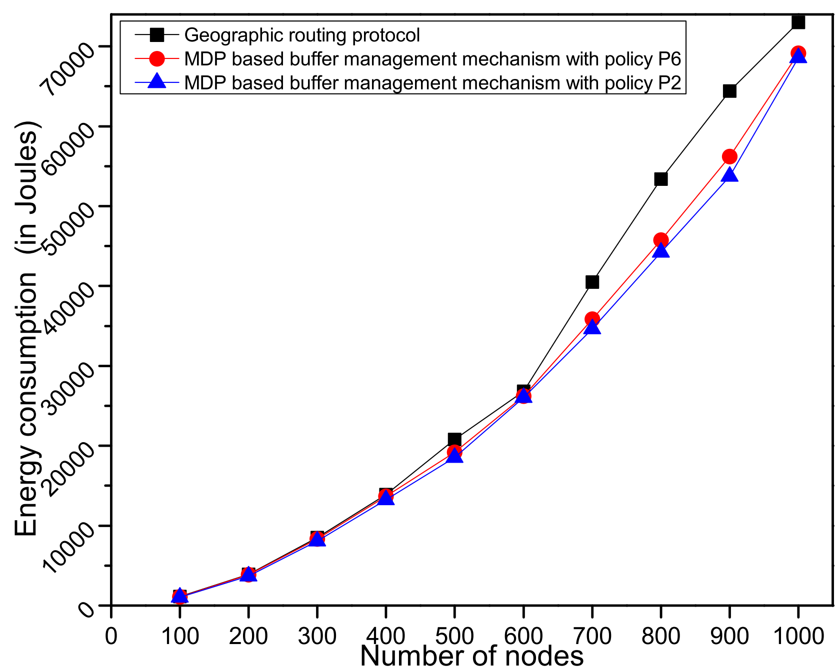


Figure 3.7: Energy consumption of a sensor tree in the network ($\lambda = 10, \mu = 15$, buffer size = 100 packets)

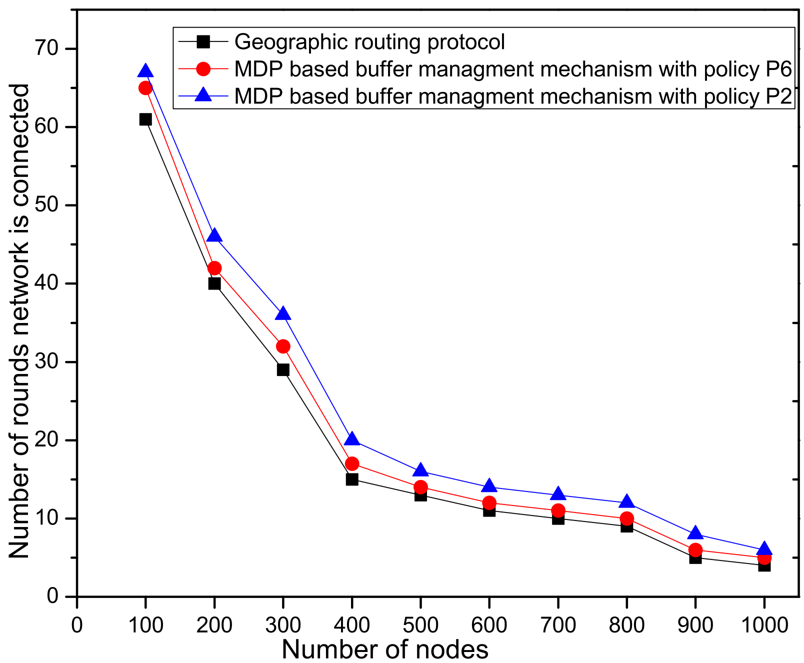


Figure 3.8: Number of data collection rounds network is connected ($\lambda = 10, \mu = 15$, buffer size = 100 packets)

3.4.1 Simulation Results and Discussions

Sensor nodes are randomly deployed in a deployment area and trees are created with disconnected component using breadth-first-search algorithm (see Fig. 3.1). Actor determines the contact times of data collection region (DCR) (refer *Equation* (3.7)) and calculates its velocity in the sensor tree. Actor collects the data via root node of a sensor tree. Actor node is not an energy constraint node in comparison to a sensor node [2]. In each data collection round, root node collects data from all sensor nodes in a tree. Lifetime of the network is defined as the number of data collection rounds in which all the nodes are connected [19]. In simulation, retransmissions are performed till the root node get all the packets from the sensor tree.

The following simulation parameters are considered in our work. The number of nodes in a sensor tree is 1000, transmission range is 30 m, the minimum distance between any two sensor nodes in a tree is 5 m, initial energy of a node is 2500 $Joules$, $\alpha_{11} = 0.937 \times 10^{-6}$

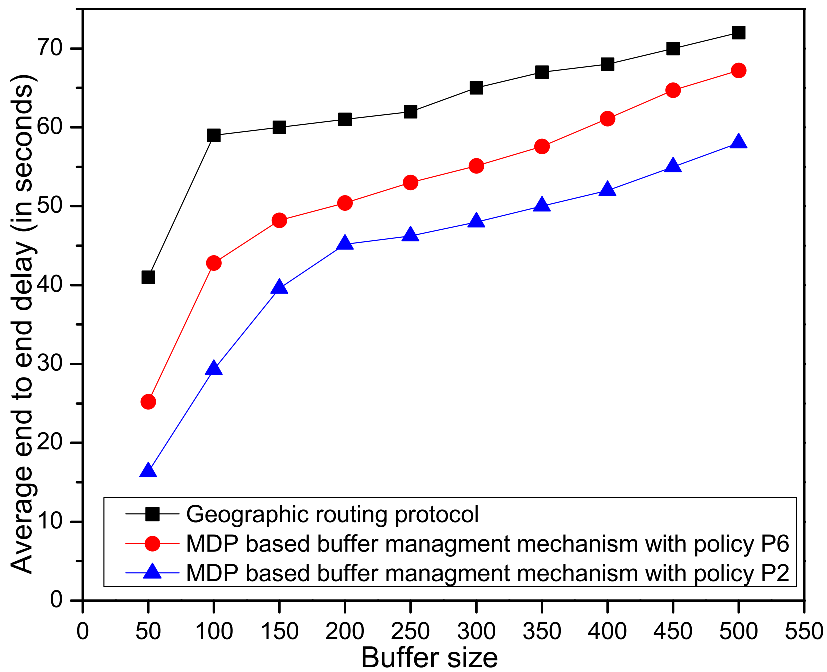


Figure 3.9: Average end-to-end delay of a packet in a sensor tree with varying buffer size ($\lambda = 10, \mu = 15$, number of nodes = 100)

$Joules/bit$, $\alpha_{12} = 0.787 \times 10^{-6}$ $Joules/bit$ [8], $\alpha_2 = 10 \times 10^{-12}$ $Joules/bit$, $\alpha_3 = 50 \times 10^{-9}$ $Joules/bit$, $\hat{d}$ = 85 m and path loss component $(\hat{n})$ is 2 [7]. Amount of data generated by each node per event i.e. l = 960 $bits$ [8] and all the nodes are having different values of β. λ is a arrival data flow (follows poisson distribution) into a node and its value is 10 in simulation. Departure parameter of link is μ and its value is considered as 15. Buffer size of a node is 100 packets and each packet size is 960 bits. Two policies, namely, policy P_2 (*Drop 10% packets from buffer at state* B_3, B_4) and policy P_6 (*Drop 15% packets from buffer at state* B_4) have been experimented in simulation for simplicity.

In this work, the proposed MDP based buffer management mechanism policies P_2 and P_6 have been compared with geographic routing protocol [10] in terms of average end-to-end delay of the packet, energy consumption of the sensor tree and the number of rounds in which all the nodes in a sensor tree are connected.

Fig. 3.6 shows the comparison of average end-to-end delay between the MDP based buffer management mechanism policies and geographic routing protocol [10] by consider-

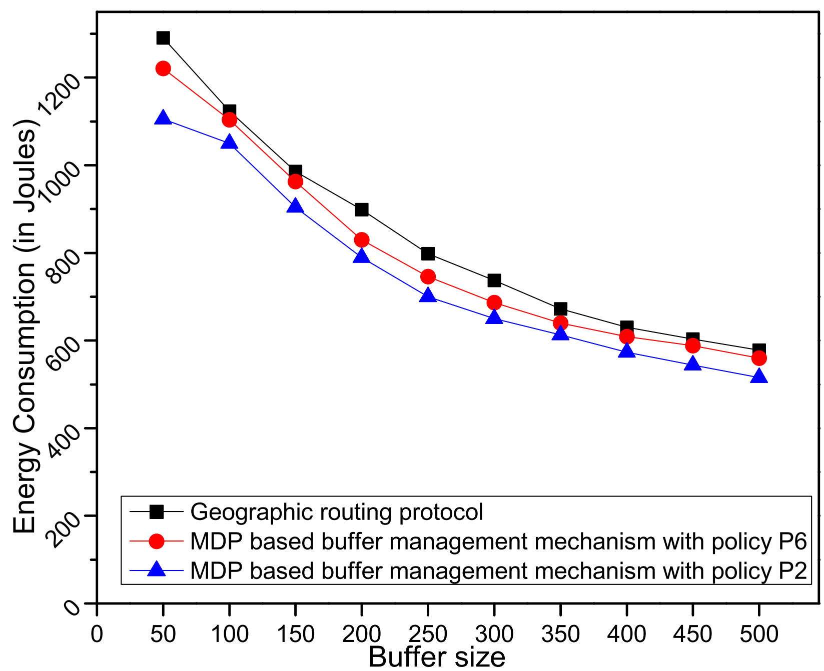


Figure 3.10: Energy consumption of a sensor tree with varying buffer size ($\lambda = 10, \mu = 15$, number of nodes=100)

ing the buffer size as 100 packets, arrival data flow (λ) as 10 and service rate (μ) (departure parameter) as 15. It can be observed that from Fig. 3.6, policy P_2 provides less delay as compared to the policy P_6 and geographic routing protocol (refer Table 3.6). As the number of nodes increases, the height of the tree also increases. The nodes in nearest levels to the root node of a tree drops more number of packets due to more data traffic. Every node determines its optimal buffer state and number of packets to be dropped using the proposed MDP algorithm. The total number of dropping of packets is reduced with the proposed MDP algorithm. Therefore, the number of retransmissions is reduced which leads to the reduction in average end-to-end delay of a packet as shown in Fig. 3.6. In Fig. 3.6, after 600 nodes, delay has increased rapidly using geographic routing protocol in comparison to the P_2 and P_6 policies.

The energy consumption of a sensor tree (refer Fig. 3.7) is compared between the MDP based buffer management mechanism policies and geographic routing protocol [10] by considering the buffer size as 100 packets, arrival data flow (λ) as 10 and service rate

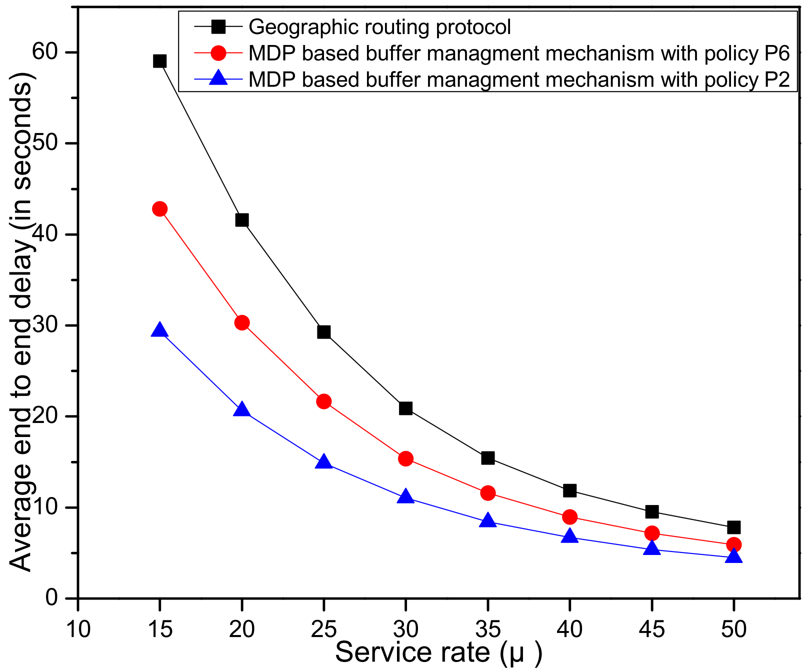


Figure 3.11: Average end-to-end delay of a packet in a sensor tree with varying departure time ($\lambda = 10$, buffer size=100 packets, number of nodes=100)

(μ) (departure parameter) as 15. In the proposed method, low priority packets (which are generated by the nearest level nodes) are dropped to reduce the delay and as well as energy consumption. The proposed MDP algorithm reduces the number of dropped packets (low priority) which leads to the reduction in the number of retransmissions. Therefore, energy consumption of a tree is reduced with the proposed MDP algorithm. It can be seen from Fig. 3.7 that energy efficiency of the sensor tree is improved (i.e. reduction of energy consumption upto 19.83%) using the proposed MDP based buffer management mechanism.

Fig. 3.8 shows performance of the proposed mechanism in terms of network lifetime (defined as number of rounds in which network is connected). It can be seen from Fig. 3.8 that network remains connected more number of rounds using the MDP based buffer management mechanism in comparison to geographic routing protocol. The proposed mechanism identifies the optimal state to drop the low priority packets and hence, it reduces the number of retransmissions when compared with the geographic routing protocol. There-

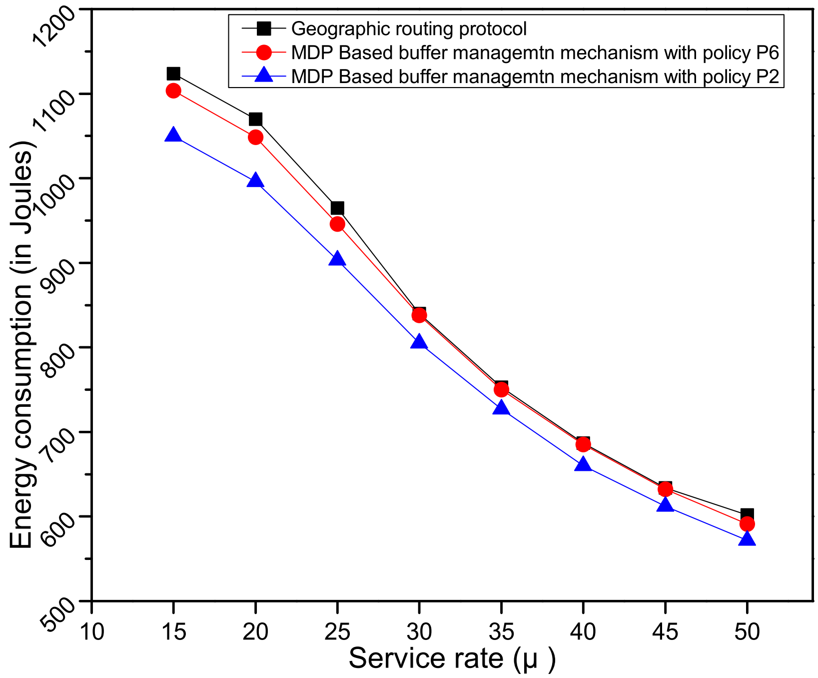


Figure 3.12: Energy consumption of a sensor tree with varying departure time ($\lambda = 10$, buffer size=100 packets, number of nodes=100)

fore, energy consumptions of the sensor tree is reduced. Further, the number of rounds, in which network is connected, increases using the proposed mechanism. As the number of nodes increases, buffer of the nodes in the initial levels of the tree will be overloaded with heavy traffic. Further, the node's energy deplete quickly due to more number of packet droppings. Hence, connectivity of the network (in terms of rounds) is reduced as the number of nodes increases (refer Fig. 3.8).

Fig. 3.9 and Fig. 3.10 show the comparison of delay and energy consumption of a tree between proposed buffer management mechanism policies and geographic routing protocol [10] by varying the buffer size of a node from 50 packets to 500 packets, respectively. Simulation results are presented for 100 nodes in the both cases. A node can accommodate more number of packets as the buffer size increases. Therefore, the queuing delay of a packet increases where, queuing delay is the amount of time packets will stay in buffer. It can be observed from Fig. 3.9 that as the buffer size (memory capacity) increases, the average end-to-end delay is increased. As buffer size increases, more number of packets can be accommodated in the buffer of a node and this approach reduces the number of dropping of packets. Therefore, the number of retransmissions for dropped packets also reduces. Hence, reductions in number of retransmissions occurs which leads to the reduction in the energy consumption (refer Fig. 3.10). It is observed from Fig. 3.9 that proposed MDP based buffer management mechanism with policy (P_2) gives significant reduction in delay when compared with the geographic routing protocol [10]. Energy consumption is reduced up to 16.75% (refer Fig. 3.10) using the MDP based buffer management mechanism.

Fig. 3.11 and Fig. 3.12 show the comparison of delay and energy consumption of a tree between proposed buffer management mechanism policies and geographic routing protocol [10] by varying the service rate (departure parameter) from 10 to 50, respectively. Simulation results are presented for 100 nodes and each node buffer size is 100 packets. It can be observed from Fig. 3.11 that increase in service rate will reduce the delay because the queuing delay of a packet is reduced. This leads to reduction in packets dropping (i.e nodes accommodate the new packets). Hence, energy consumption is reduced in a tree (ref Fig. 3.12). It is observed from Fig. 3.11 that proposed MDP based buffer management mechanism with policy (P_2) provides significant reduction in delay. Energy consumption is reduced up to 7.04% (see Fig. 3.12) using the MDP based buffer management mechanism.

3.5 Summary

In this chapter, an adaptive buffer management mechanism using Markov Decision Process (MDP) has been presented for a tree-based *Wireless Sensor and Actor Network*. Analysis on energy consumptions is performed for a typical data acquisition model and a retransmission (of dropped packets) model in a tree based sensor actor network. The optimal buffer state of a node is evaluated to drop the minimal number of packets to reduce the average end-to-end delay and energy consumptions. The bounds on visiting times and contact times are estimated for a mobile actor to mitigate buffer overflow problem (dropping of packets). Simulation results show that the average end-to-end delay reduces significantly using the proposed Markov Decision Process based approach in comparison to a geographic routing protocol. Further, it is observed that there is a reduction of energy consumption upto 19.83% using the proposed MDP based buffer management mechanism in comparison to a geographic routing protocol. In this chapter, reliable data transmission is achieved by reducing the packet drops at buffer level. In the next chapter, reduction of packet drops at link level has been investigated for reliable data gathering in a flat WSAN.

Chapter 4

Reliable Data Gathering Mechanism using Opportunistic Encoding in WSANs

In a WSAN with faulty regions, achieving high degree of data transmission reliability with appropriate end-to-end delay is an inescapable challenge. Data transmission reliability depends on the quality of the wireless channel. It will change dynamically with environmental conditions, such as fog, rainfall, high temperature [11] and movable obstacles. Therefore, the quality of link varies dynamically in the presence of environmental interferences.

Packet loss may happen due to fluctuation of wireless links (quality) in the presence of adverse environmental conditions. Reliable data collection can be achieved using acknowledgment based retransmission (automatic retransmission) approaches [12, 13]. These approaches require more number of data transmissions (including retransmissions) to achieve reliable data collection. As the number of data transmissions increase, energy consumption and packet delivery delay also increase. Hence, there is a requirement for an efficient data delivery approach for the delay sensitive applications in WSAN. Network coding mechanism (is a mechanism by which raw packets are encoded and transmitted by sensor nodes [99]) can be applied to improve reliability in data gathering and to reduce packet delivery delay (in case a link is unreliable [12, 13]). But, the level of packet redundancy (to be used in encoding process) influences the number of data transmissions and energy consumptions. In order to improve the data collection reliability along with energy efficiency, appropriate level of packet redundancy must be chosen adaptively based on the quality of the wireless links.

In this chapter, *a reliable data transmission mechanism using opportunistic encoding* has been proposed for a wireless sensor and actor network to achieve reliable data collection along with energy efficiency in the presence of faulty regions. A *markov decision process* (MDP) algorithm has been designed by considering variable (wireless) link quality. The novelty of this work lies in the MDP algorithm which chooses: (a) whether to apply coding for relaying packets, and (b) the level of packet redundancy. Further, a *Learning Automata* based mechanism has been proposed to route the traffic through alternate links, in case, the present link quality is bad. The major contributions of this chapter are as follows:

- Design of a network coding based reliable data gathering mechanism for a wireless sensor and actor network in the presence of faulty regions.

- Design of a learning automata based mechanism to route the traffic through alternate links, when the present link quality becomes bad.

- Simulation results are presented to show the efficacy of the proposed mechanism in terms of packet latency, number of data transmissions, lifetime and energy consumption.

The rest of the chapter is organized as follows. In Section 4.1, motivation and problem formulation have been discussed. In Section 4.2.1, network model is introduced. In Section 4.2.2, A *Network Coding based Reliable Data Transmission Mechanism* has been proposed to provide reliable data collection in the wireless sensor and actor network. Network coding using deterministic coding vectors has been discussed in Section 4.2.3. In Section 4.2.4, a case study is presented. In Section 4.2.5, a link quality has been analyzed using *Markov Decision Process* to determine the required amount of packet redundancy. Further, performance evaluation has been done by presenting the simulation results in Section 4.3. Finally, the work is concluded in Section 4.4.

4.1 Motivation and Problem Formulation

Achieving reliable data collection (packet level reliability) with stringent end-to-end delay requirement is a challenging issue in some of the applications (in WSAN with faulty regions), such as issuing tsunami alerts, chemical attack detection, forests fire detection and intrusion detection in military surveillance [2, 3, 110]. A link may be unreliable in the presence of a faulty region that results in loss of packets. In an automatic retransmission (AR) approach, a source packet will be retransmitted till reception of an acknowledgment [12, 13]. This leads to increase in the number of data transmissions, delivery delay and energy consumption [111]. Therefore, the automatic retransmission approach is inefficient for delay-sensitive and energy-constrained applications in WSAN. Network coding mechanism encodes raw packets appropriately to reduce the number of transmissions [12, 99]. Network coding can be applied based on link loss rates and appropriate level of redundancy. Network coding offers a high degree of reliable data with low delay in comparison to the non-coded data transmission when a wireless link is unreliable. Additional computation overhead helps in reducing the number retransmissions. Retransmission overhead with network coding approach is less than the overhead introduced by the retransmission-based approach.

Wireless sensor network operates in dynamic (random) environments and most of the system parameters (such as link and energy) deviate from the required values. In such cases, system parameters should be adaptively changed to meet the desired goals of a network. In such a stochastic environment, a different set of optimized decisions or actions can be implemented in sensor nodes to achieve design goals. A mathematical framework called *Markov Decision Processes* (MDPs) can be adopted to model the system dynamics to optimize the desired objectives of the network.

To address the above mentioned problems, a network coding based reliable data transmission mechanism has been proposed in our work for delay-sensitive applications. Further, a *markov decision process* is designed for opportunistic network coding decisions. Our algorithm determines the level of packet redundancy (adaptively) in the network coding process for reduction of data transmissions. A learning automata based mechanism is proposed to route the traffic through another node if the quality of link to the next node is less than a threshold (the value of the threshold may vary from application to application).

In [12], source node determines the redundancy level by considering the link loss rate of the worst link to the sink node. However, in our work, a MDP algorithm has been designed to analyze the quality of link states which may change with the adverse environmental conditions. The novelty of the proposed mechanism lies on applying MDP to choose the network coding decisions and to determine the appropriate level of packet redundancy in the encoding process. The proposed MDP based mechanism determines the optimal policy which ensures the reliable data delivery with optimal packet redundancy level. Moreover, the optimal policy reduces the number of data transmissions and thereby, improves the energy efficiency.

4.2 A Network Coding based Data Transmission Mechanism

In this section, firstly, the network model is presented. Secondly, a *Network Coding based Data Transmission Mechanism* is proposed to provide reliable data gathering. Further, an encoding process is presented using deterministic coding vectors. Finally, a markov decision process based algorithm has been proposed to analyze the quality of wireless communication link.

4.2.1 Network Model

A wireless sensor and actor network consists of n sensor nodes that are deployed randomly in an area A as shown in Fig. 4.1. Let, $G(V, E)$ be a graph representing the monitoring area A that contain the sensor nodes and an actor, where $V = \{v_1, v_2, ..., v_n\}$ denotes a set of sensor nodes and $E = \{e_1, e_2, ..., e_n\}$ denotes the set of communication links. A faulty region is defined as a region which is affected by adverse environmental effects for temporary period of time [11, 112, 113, 114]. Nodes in a faulty region are called as faulty nodes. After the temporary period, functionality of the faulty nodes will be normal. But,

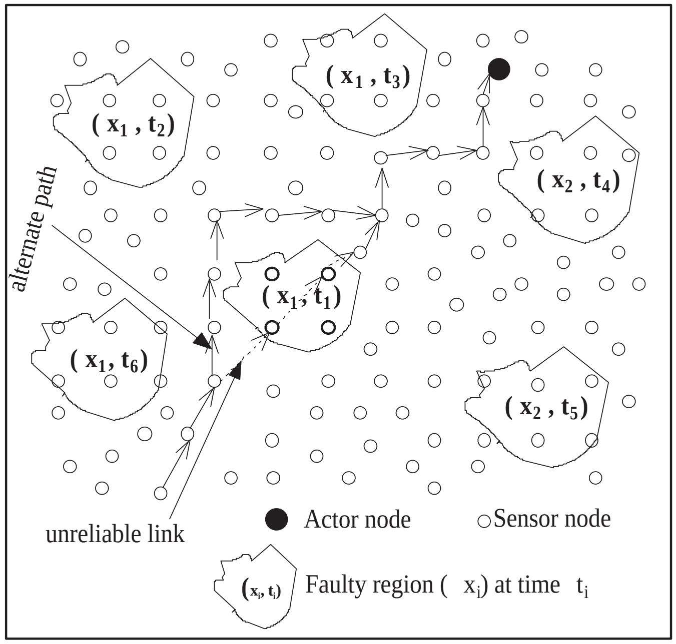


Figure 4.1: The illustration of faulty regions in the wireless sensor and actor network

a node will be failed permanently due to energy depletion. In Fig. 4.1, the regions x_1 and x_2 are affected by the environmental conditions at different times. In a *data collection round*, an actor node collects the data from every sensor node in the network. The actor node process the collected data and performs the appropriate actions [2, 115]. *Lifetime* of a network is defined as the number of data collection rounds upto which all the nodes are connected [19]. p_{th} and $\hat{p}_{th}$ are the lower and upper threshold values of a wireless link quality, respectively. These values may be chosen based on the application requirements. Table 4.1 shows the notations used in this chapter.

4.2.2 A Network Coding based Data Transmission Mechanism

In this section, a *Network Coding based Data Transmission Mechanism* is proposed. The proposed mechanism (*Algorithm* 4.1) has three major functions, such as *LINK_QUALITY()*, *DIVERT_TRAFFIC()* and *MDP_Redundancy()*. The *LINK_QUALITY()* function (refer *Algorithm* 4.2) finds the quality of links (i.e. the probability of successful transmission) of a node to all its neighbors. The *DIVERT_TRAFFIC()* (refer *Algorithm* 4.3) function diverts the traffic through one of the neighbor nodes. The *MDP_Redundancy()* (refer *Algorithm*

Table 4.1: Summary of Notations

Symbol	Description
$G = (V, E)$	V is set of sensor nodes and E is set of edges
A	Monitoring area
n	Number of sensor nodes in the network
d	Degree of a node
P_{ij}	Link quality (Probability of successful transmission of a packet from node-i to node-j)
p_{th}	Lower threshold value of a link quality
$\hat{p}_{th}$	Upper threshold value of a link quality
r_l	Level of a encoded packet redundancy
E^i	Energy of node-i
D^i	Degree (number of neighbors a node have) of node-i
B^i	Buffer of node-i
Bd^i	Bandwidth of node-i
S_i	Score of a node-i
$\hat{P}_{ij}$	Selection probability of node-j from node-i
$\hat{P}_{ij}^{old}$	Previous selection probability of node-j from node-i
α_{11}	Energy consumption per bit by the transmitter electronics
α_2	Energy consumption per bit in the transmit op-amp
α_{12}	Energy consumption per bit by the receiver electronics
α_3	Energy consumption for sensing a bit
α_1	$\alpha_{11} + \alpha_{12}$
$\hat{n}$	Path loss exponent
H_s	Number of broadcasted $Hello$ packets
H_r	Number of ACK received for respective $Hello$ packets

4.6) function decides whether to encode the packets or not and it finds the level of packet redundancy which will be applicable in the encoding process (if the decision is *Do Network Coding*). Moreover, *MDP_Redundancy()* function returns the optimal policy in terms of, number of data transmissions per data collection round (including retransmissions), data gathering (i.e. collection) reliability, packet latency and energy efficiency.

The *Wireless Sensor and Actor Network* ($G(V, E)$), as discussed in Section 4.2.1, is provided as input to the *Algorithm* 4.1. The link quality of every node in the network is computed using *Algorithm* 4.2 (as mentioned in lines 1-7, *Algorithm* 4.1). A node broadcasts *Hello* packets (H_s number of packets) in a duration of time T (line 2, *Algorithm* 4.1). The node will receive the acknowledgments (H_r number of ACK) for respective broadcasted packets from its neighbors (where d is the maximum number of neighbors) (lines 3-4, *Algorithm* 4.1). In *Algorithm* 4.2, P_{ij} is the link quality (i.e. probability of successful transmission of a packet) of a node (v_i) with its neighbor node (v_j) and is estimated as the ratio of H_r to H_s. Further, if P_{ij} is less than a threshold value (p_{th}), then the node will choose another next hop node from its neighbors as a new next hop node to route the data traffic using *Algorithm* 4.3 (as mentioned in lines 10-11, *Algorithm* 4.1). If the P_{ij} is greater than or equal to $\hat{p}_{th}$, then the packets are forwarded without using encoding (as per lines 12-13, *Algorithm* 4.1). However, in other cases of link qualities, the decision (whether to do encoding or not) will be taken by *MDP_Redundancy()* (lines 15-16, *Algorithm* 4.1) which is explained in section 4.2.5.

In *Algorithm* 4.3, a node selects a new next node which has maximum selection probability as compared with its neighbors using Learning Automata (LA). Every node calculates the score value of its neighbors using *Algorithm* 4.4 (as mentioned in lines 2-3, *Algorithm* 4.3). Score (S_j) of a node-j defines the level of resources of the node and it depends on the resources available with node-j, such as energy (E^j), degree (D^j), buffer (B^j) and bandwidth (Bd^j) [116]. In *Algorithm* 4.4, Score of a node is calculated by

$$S = W_{energy} * E_{nor} + W_{degree} * D_{nor} + W_{buffer} * B_{nor} + W_{bandwidth} * Bd_{nor}$$

where W_{energy}, W_{degree}, W_{buffer} and $W_{bandwidth}$ are the weights of energy, degree, buffer and bandwidth, respectively.

Normalized values of energy (E_{nor}), degree (D_{nor}), buffer (B_{nor}) and bandwidth (Bd_{nor})

are estimated using *Algorithm* 4.5 (as given in lines 1-3, *Algorithm* 4.4). Further, probability of selection of node-j $(\hat{P}_{ij})$ is calculated using score (S_j), link quality (L_{ij}) and previous probability of selection value ($\hat{P}_{ij}^{old}$) (line 4, *Algorithm* 4.3) and is given by

$$\hat{P}_{ij} = W_{score} * S_j + W_{link} * L_{ij} + W_{old} * \hat{P}_{ij}^{old}$$

where W_{score}, W_{link} and W_{old} are weights of score, link quality and previous probability of selection of node-j, respectively and $W_{score} + W_{link} + W_{old} = 1$.

A neighbor node (which has maximum selection probability) will be selected as a new next node (by following lines 5-10, *Algorithm* 4.3). The entire process (as mentioned in *Algorithm* 4.1 and discussed in this section) has been represented in a flowchart as shown in Fig. 4.2. The encoding process using deterministic coding vectors is presented in the next section.

4.2.3 Network Coding using Deterministic Coding Vectors

Network coding provides a high degree of reliable data collection with low delay by introducing redundancy of packets when a link is unreliable [12, 111]. However, the amount of redundancy (to be maintained) depends on the quality of the link. The amount of redundancy affects the energy efficiency. Moreover, proposed mechanism adaptively maintains the redundancy levels using network coding (depending on quality of links) to achieve energy efficient reliable data collection.

In the proposed mechanism, every node forwards the coded packets. Deterministic coding vectors are applied in the encoding process. A node encodes the pair-wise packets using deterministic coding vectors [12]. The number of coded packets (redundancy levels) will be determined by markov decision process. The redundancy level may be varied from 1 to k_l where, k_l is the maximum redundancy level of an encoded packet. However, this depends on the link loss rates and application requirements. In this work, a maximum of ten deterministic coding vectors (linearly independent) ($k_l = 10$) are implemented in the encoding process. The matrix C (*Equation* 4.1) shows the linear independent deterministic

Algorithm 4.1 Network Coding based Data Transmission Algorithm

Input: $G(V, E)$

Output: Optimal policy and level of packet redundancy

1: **for** $i = 1$ to n **do**

2: Node $v_i \in V$ broadcast the *Hello* packets in the duration of T.

3: **for** $j = 1$ to d **do** // (d is the degree of node v_i)

4: H_r and H_s are the number of *ack* received and number of broadcasted *Hello* packets respectively.

5: P_{ij} =**LINK-QUALITY(**H_r**,** H_s**)** (*Algorithm* 4.2) //(every node calculates the quality of link and quality of link state of all its neighbors)

6: **end for**

7: **end for**

8: Node j is the next node of node i. P_{ij} is the link quality of node i to node j

9: **for** $i = 1$ to n **do**

10: **if** $P_{ij} < p_{th}$ **then**

11: next_node= **DIVERT_TRAFFIC(**i**)** //(Divert the traffic through next_node (using *Algorithm* 4.3))

12: **else if** $P_{ij} \geq \hat{p}_{th}$ **then**

13: Do not use Network Coding

14: **else**

15: Node v_i calls the **MDP_Redundancy()** (*Algorithm* 4.6). It will select the optimal policy (in terms of number of transmissions) which is suitable to its quality of link state and redundancy level (r_l) based on the P_{ij}.

16: Node v_i transmit the data using network coding with redundancy r_l.

17: **end if**

18: **end for**

19: go to *step 1*

Algorithm 4.2 LINK_QUALITY(H_r,H_s)

Input: H_r,H_s.

Output: Quality of a link or probability of successful transmission (P_l).

if $H_r \geq H_s$ **then**

$P_l = 1$

else

$$P_l = \frac{H_r}{H_s}$$

end if

return P_l

Algorithm 4.3 DIVERT_TRAFFIC(i)

$max = -1, nn = -1$
for $j = 1$ to d **do** //d is the degree of node i.
    $S_j = SCORE(E^j, D^j, Bd^j, B^j)$ //(*Algorithm* 4.4)
    $\hat{P}_{ij} = W_{score} * S_j + W_{link} * L_{ij} + (1 - W_{score} - W_{link}) * \hat{P}_{ij}^{old}$
    if $\hat{P}_{ij} > max$ **then**
        $max = \hat{P}_{ij}$
        $nn = j$
    end if
end for
return nn

Algorithm 4.4 SCORE(E,D,Bd,B)

Input: E,D,Bd,B
Output: S

Set E_{nor}=NORMALIZE(E,$maxenergy$,$minenergy$)
Set D_{nor}=NORMALIZE(D,$maxdegree$,$mindegree$)
Set B_{nor}=NORMALIZE(B,$maxbuffer$,$minbuffer$)
Set Bd_{nor}=NORMALIZE(Bd,$maxbdwidth$,$minbdwidth$)
S= W_{energy}*E_{nor} +W_{degree}*D_{nor}+ W_{buffer}*B_{nor}+ $W_{bandwidth}$*Bd_{nor}
return S

Algorithm 4.5 NORMALIZE($value$,max,min)

Input: $value$, max, min
Output: $value_{nor}$

Set $value_{nor} = \dfrac{value - min}{max - min}$
return $value_{nor}$

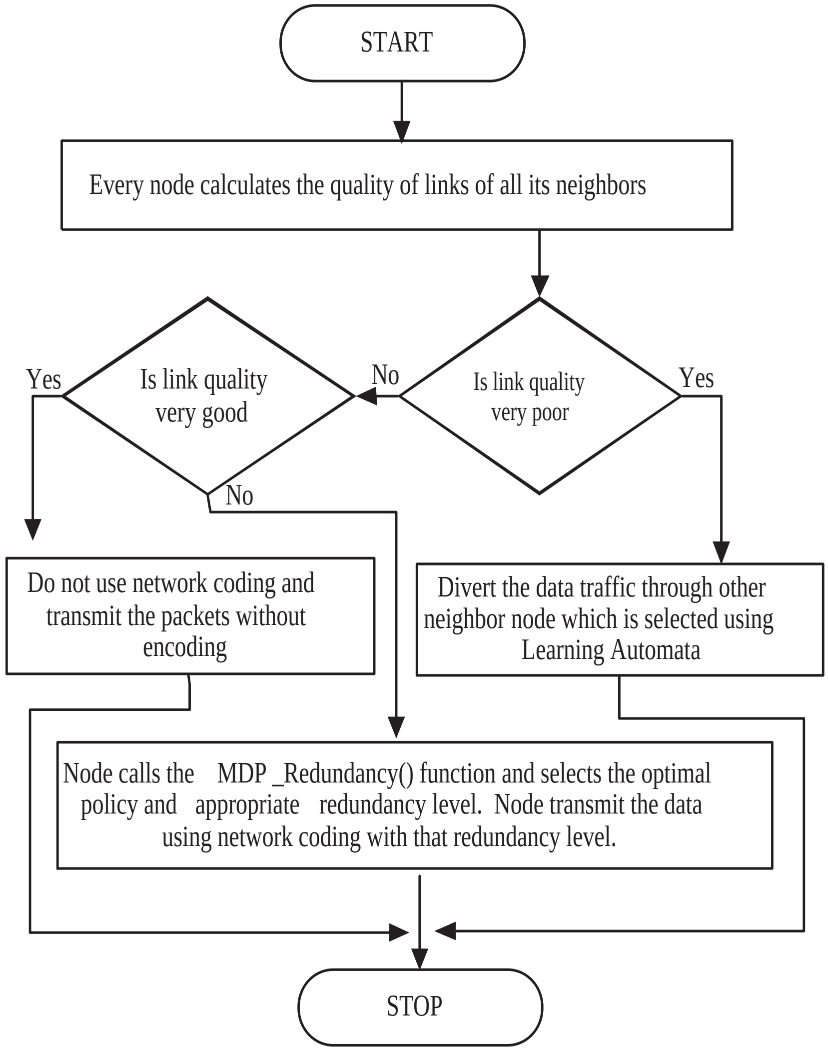


Figure 4.2: Flowchart of the proposed mechanism

coding vectors $(c_1, c_2, ..., c_{10})$ which are chosen from *GF(*2^3*)*.

$$C = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{10} \end{pmatrix} = \begin{pmatrix} 5 & 1 \\ 3 & 2 \\ 6 & 7 \\ 2 & 5 \\ 1 & 2 \\ 5 & 6 \\ 4 & 3 \\ 1 & 3 \\ 4 & 5 \\ 2 & 3 \end{pmatrix} \tag{4.1}$$

The encoded packets are obtained as given in *Equation* 4.2, where p_1, p_2 are the raw packets.

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{10} \end{pmatrix} = \begin{pmatrix} 5 & 1 \\ 3 & 2 \\ 6 & 7 \\ 2 & 5 \\ 1 & 2 \\ 5 & 6 \\ 4 & 3 \\ 1 & 3 \\ 4 & 5 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} 5p_1 + p_2 \\ 3p_1 + 2p_2 \\ 6p_1 + 7p_2 \\ 2p_1 + 5p_2 \\ p_1 + 2p_2 \\ 5p_1 + 6p_2 \\ 4p_1 + 3p_2 \\ p_1 + 3p_2 \\ 4p_1 + 5p_2 \\ 2p_1 + 3p_2 \end{pmatrix} \tag{4.2}$$

Here, $y_1, y_2, ...y_{10}$ are the encoded packets.

A node forwards an appropriate number of encoded packets based on the level of link quality. In decoding process, a node retrieves the original packets from at least two encoded packets. Further, the following lemma holds for the increment of the size of an encoded packet proportional to a constant d.

Lemma 4.2.1. *Encoded packet size increases by maximum of* $(d + 1)$ *bits in the encoding process. Encoded packet size depends on the field size of* $GF(2^d)$ *(i.e. on* d*), but not on packet size.*

Proof. The maximum deterministic coding vector with *GF(*2^3*)* is (7, 7). One of the encoded packets is ($7p_1 + 7p_2$). The number of bits required to store $7p_1$ is $(l + 3)$ bits (because, $2^3 * 2^l = 2^{l+3}$, where l is the packet size). Similarly, $7p_2$ packet requires $(l + 3)$ bits. The total number of bits required to store ($7p_1 + 7p_2$) packet is $(l + 4)$ bits (i.e. one bit for carry in the addition operation ($2^{l+3} + 2^{l+3}$)). Therefore, encoded packet size increases by a maximum of four bits. Hence, the difference between the energy consumptions for transmitting a packet with size l bits and an encoded packet with size $(l + 4)$ bits is constant (i.e. proportional to $(d + 1)$). □

4.2.4 Case Study

An example is presented by computing the number of data transmissions for an automatic retransmission (AR) approach [12] and the proposed mechanism. Let, *node*-s want to forward the packets (p_1 and p_2) to *node*-d with link loss rate (i.e. from *node*-s to *node*-d) as 0.5. In the automatic retransmission mechanism, a node retransmits the packet if it fails to detect the acknowledgment. This happens when either the packet or the ACK (acknowledgment) is not reached. Therefore, the number of data transmissions (including retransmissions and ACK) carried out for successful transmission of packets $(p_1$ and $p_2)$ is 6 (if loss probability of ACK is same as data packet). In the proposed mechanism, *node*-s will choose four encoded packets from matrix Y (refer *Equation* 4.2) and encoded packets will be transmitted. *Node*-d can extract the original packets if it receives at least two encoded packets. Therefore, proposed mechanism needs four data transmissions for successful delivery of packets (p_1 and p_2). Moreover, a retransmission increases the packet delay. However, the proposed mechanism, ensures the reliable data delivery with a network coding mechanism, which reduces the delivery delay. Further, without network coding mechanism, a node must send two p_1 and two p_2 packets (with link loss rate of 0.5) so that next receiver node can receive the two packets. But, this mechanism may not give reliable data delivery because the possible received combinations of that two packets are (p_1, p_1) or (p_2, p_2) or (p_1, p_2). Both the packets are received in the latter combination only, i.e. (p_1, p_2). Therefore, the proposed network coding based mechanism gives the high degree of reliable data delivery with a reduced number of data transmissions.

4.2.5 Analysis of Link Quality using Markov Decision Process

Quality of a wireless communication link fluctuates with dynamic adverse environmental effects and with movable obstacles. Wireless communication link may be stable in the favorable environment [11]. A non-faulty node may have different quality of links in comparison to the faulty nodes. The data may not reach to faulty nodes due to adverse environmental effects. Further, more number of data transmissions (including retransmissions) are required to achieve high data reliable data collection by using hop-by-hop automatic

Table 4.2: Expected number of packet redundancy (per pair-wise packets) for encoding process to achieve reliable data delivery

State	Quality of link (in units)	Quality of link (P_l) (Probability of successful transmission)	Redundancy ($r_l = \frac{2}{P_l}$)
L_1	1	0-0.2	10
L_2	2	0.2-0.4	5
L_3	3	0.4-0.6	3.32
L_4	4	0.6-0.8	2.5
L_5	5	0.8-1	2

retransmission approach in the presence of faulty regions [12]. This leads to increase in the energy consumption and packet latency. Therefore, to reduce the number of data transmissions and to increase the reliable data delivery, network coding mechanism is applied. However, data delivery reliability depends on the level of packet redundancy in the encoding process. Selection of appropriate level of packet redundancy improves the energy efficiency and reduces the packet latency. The quality of a link helps in deciding the appropriate level of redundancy. To take effective decisions, quality of link is modeled as *Markov Chain*. According to *Markov Theory*, the future state of a link quality depends on the current state rather than the past state [76, 79, 117].

In this work, five different states L_1, L_2, L_3, L_4 and L_5 are defined on the different levels of link qualities (refer Table 4.2). State L_5 has maximum link quality and state L_1 has minimum link quality. Let, the set $\{L_0, L_1, L_2, ...\}$ represents the number of link quality units (levels) that a link has at $\{0^{th}, 1^{st}, 2^{nd}, ...\}$ time slots. $P\{L_{t+1} = j|L_t = i\}$ is the probability that a link has i units of link quality at time t and it has j units of link quality at $(t+1)$. $\{LR_0, LR_1, LR_2, ...\}$ are the number of units reduction in quality of link during the $\{0^{th}, 1^{st}, 2^{nd}, ...\}$ time slots. It is assumed that the reduction in the quality of link (LR_t) follows the Poisson distribution with mean (λ) as one unit (occurrence of faulty region follows the Poisson distribution) [76, 117]. The probability of reduction of number of link quality units ($\hat{n}$) during the interval $(t+1)$ is given by *Equation* 4.3.

$$P\{LR_{t+1} = \hat{n}\} = \frac{\lambda^{\hat{n}} e^{-\lambda}}{\hat{n}!} \tag{4.3}$$

The possible states for quality of link at a given time slot are L_5 with a link quality

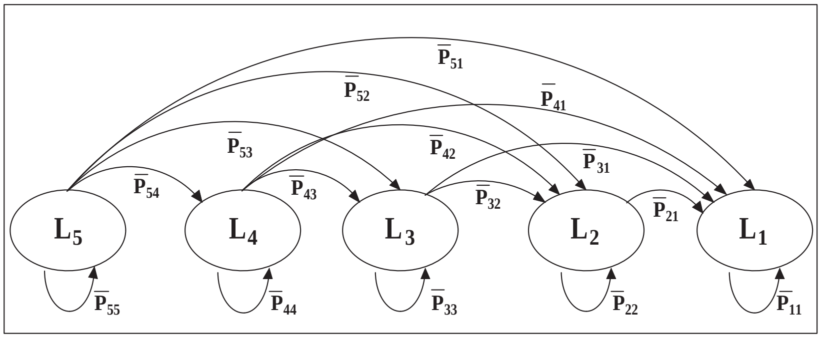


Figure 4.3: The State Transition Diagram

Table 4.3: Initial Transition Matrix

$\overline{P}_{ij}$	L_5	L_4	L_3	L_2	L_1
L_5	0.368	0.368	0.184	0.061	0.019
L_4	0	0.368	0.368	0.184	0.08
L_3	0	0	0.368	0.368	0.264
L_2	0	0	0	0.368	0.632
L_1	0	0	0	0	1

value of 5 units, L_4 with a link quality value of 4 units, L_3 with a link quality value of 3 units, L_2 with a link quality value of 2 units and L_1 with a link quality value of 1 unit (refer Table 4.2). In level L_5, quality of link is better than level L_1 (link loss rate is more at state L_1). A node may be isolated from the network in state L_1 due to its bad link quality. The value of LR_{t+1} is calculated for transition from L_t to L_{t+1} by the following:

$$L_{t+1} = \begin{cases} max\{5 - LR_{t+1}, 1\}, & \text{if } L_t = 5, \\ max\{L_t - LR_{t+1}, 1\}, & \text{if } L_t \leq 4, \end{cases}$$

The probability value $\overline{P}_{ij}$ for transition from L_t to L_{t+1} is calculated by placing the appropriate value of LR_{t+1} in *Equation* 4.3. Table 4.3 shows the initial transition matrix and Fig. 4.3 shows the state transition diagram, where L_1, L_2, L_3, L_4 and L_5 are the states of the link quality and $\overline{P}_{ij}$ is the probability of transition from state L_i to L_j. Link quality deteriorates only in the period of adverse environmental conditions. After this period, wireless link quality may improve.

In this work, the maximum number of data transmissions required for the successful transmission of a packet is considered as a *reward*. If the quality of a link is maximum (very good), then packet will not be lost and the number of data transmissions will be very much less. If the link quality is not good, then more number of retransmissions need to be done for successful delivery of the packet using hop-by-hop mechanism. In this case, energy consumption and packet delay will be increased. The number of data transmissions (including ACK and retransmissions) generated per data packet can be calculated by *Equation* 4.4 (which is adopted from [12]).

$$N_d = (1 + \sum_{x=1}^{n_r} F_x) + (1 + \sum_{x=1}^{n_r} F_x) * P_{ij} \tag{4.4}$$

where, $F_x = (1 - P_{ij}^2)^x$.

In *Equation* 4.4, N_d is the total number of transmissions (including ACK and data transmissions), n_r is the maximum number of retransmissions (n_r=5) , P_{ij} is the probability of successful transmission from node i to node j, F_x is the probability that a node fails to detect the corresponding ACK of the x^{th} data transmission. The first part of the *Equation* 4.4 denotes the number of data transmissions and the second part denotes the number of ACK transmissions per data packet for an automatic retransmission (AR) approach [12]. In the automatic retransmission approach (without encoding the data), the number of data transmissions (refer Table 4.2 for different levels of link quality) required for the successful transmission of a packet is calculated by using *Equation* 4.4. Table 4.2 shows the number of data transmissions (number of redundancy) needed to achieve the reliable data delivery in the network coding based approach. Thus, the maximum number of transmissions (expected costs) are calculated in both the approaches (hop-by-hop automatic retransmission approach and network coding based approach) for different levels of link quality (refer Table 4.4). This will be used in finding the reward (refer *Equation* 4.5) in the proposed MDP approach.

In this work, three decisions are considered for MDP as shown in Table 4.5. The first decision is *Do Network Coding*, where a node send the data after encoding and the level of redundancy depends on its link quality (refer Table 4.2). In *decision 1*, the number of data transmissions will be reduced. Further, packet delay and energy consumption will also be

Table 4.4: Expected cost (in terms of number of transmissions per pair-wise packets i.e. any two encoded combinations (refer *Equation* 4.2))

Decision	Relevant States	Expected cost due to encoding(refer Table 4.2)	Expected cost due to AR approach (refer *Equation* 4.4)	Total expected cost
Do Network Coding	L_1	10	0	10
	L_2	5	0	5
	L_3	3.32	0	3.32
	L_4	2.5	0	2.5
Don't Do Network Coding	L_2	0	11.34	11.34
	L_3	0	8.26	8.26
	L_4	0	5.6	5.6
	L_5	0	4	4

Table 4.5: Decisions and actions

Decision	Action	Relevant States
1	Do Network Coding	L_1, L_2, L_3, L_4
2	Don't Do Network Coding	L_4, L_5
3	Divert (route) the Traffic	L_1

Table 4.6: Policies

Policy	Verbal description	D_5	D_4	D_3	D_2	D_1
P_1	Do Network Coding if state is L_1	2	2	2	2	1
P_{12}	Do Network Coding if state is L_1, L_2	2	2	2	1	1
P_{13}	Do Network Coding if state is L_1, L_3	2	2	1	2	1
P_{14}	Do Network Coding if state is L_1, L_4	2	1	2	2	1
P_{123}	Do Network Coding if state is L_1, L_2, L_3	2	2	1	1	1
P_{1234}	Do Network Coding if state is L_1, L_2, L_3, L_4	2	1	1	1	1
P_d	Divert the traffic if the state is L_1	2	2	2	2	3

Algorithm 4.6 MDP_Redundancy ()

1: **for** all policies **do**
2: Find steady state probabilities Π_1, Π_2, Π_3,...,Π_{n_s} where n_s is the number of states (probability values will be calculated by transition probability matrix of each policy)
3: Reward = $\Pi_1 R_1$ + $\Pi_2 R_2$ + $\Pi_3 R_3$ +,...,+ $\Pi_{n_s} R_{n_s}$
4: **end for**
5: Select the optimal policy based on the rewards. If the policy is the network coding based policy, then it returns the level of redundancy (r_l) which will be used in encoding process to get reliable data collection.

Table 4.7: Transition Matrix for Policy P_1

$\overline{P}_{ij}$	L_5	L_4	L_3	L_2	L_1
L_5	0.368	0.368	0.184	0.061	0.019
L_4	0	0.368	0.368	0.184	0.08
L_3	0	0	0.368	0.368	0.264
L_2	0	0	0	0.368	0.632
L_1	1	0	0	0	0

Table 4.8: Transition Matrix for Policy P_{12}

$\overline{P}_{ij}$	L_5	L_4	L_3	L_2	L_1
L_5	0.368	0.368	0.184	0.061	0.019
L_4	0	0.368	0.368	0.184	0.08
L_3	0	0	0.368	0.368	0.264
L_2	1	0	0	0	0
L_1	1	0	0	0	0

Table 4.9: Policies and rewards

Policy	Π_1	Π_2	Π_3	Π_4	Π_5	Reward
P_1	0.1817	0.1820	0.1812	0.1674	0.2875	7.46
P_{12}	0.08154	0.14064	0.2215	0.2047	0.33515	5.9
P_{13}	0.09423	0.1072	0.1605	0.2347	0.4031	5.618
P_{14}	0.1205	0.12152	0.13298	0.16809	0.45679	5.928
P_{123}	0.0296	0.0759	0.1798	0.2629	0.4516	4.552
P_{1234}	0.0116	0.03737	0.11274	0.22549	0.61274	3.692

reduced. The second decision is *Don't Do Network Coding*, where the data will be sent without encoding. The *decision 2* will be considered if the quality of link is good. Finally, the third decision is *Divert the Traffic*, where a node will not get a link (i.e. the quality of link is less than a threshold value (p_{th})) to the next node. Moreover, the traffic will be routed by selecting a new next node from its neighbors (refer *Algorithm* 4.3).

Table 4.6 shows the decision to be taken in each state for each policy. The first policy (P_1) is *Do Network Coding if state is* $L1$, if a node's link is in state L_1 then the node encodes the packets before sending. The level of packet redundancy depends on the link quality (refer Table 4.2). Therefore, reliable data delivery is achieved with reduced number of packet redundancy (i.e. reduced number of data transmissions) in policy P_1. Table 4.7 shows the transition matrix for Policy P_1. Expected number of data transmissions (reward) is calculated by using the following equation:

$$Reward = \sum_{k=1}^{n_s} [\Pi_k R_k] \tag{4.5}$$

where, Π_k is the steady state probability and it is calculated using transition probability matrix (Table 4.7 for policy P_1 and Table 4.8 for policy P_{12}). R_k is the expected cost (number of data transmissions) (refer Table 4.4) at state L_k and n_s is the number of states. *Algorithm* 4.6 shows the calculation of reward for all policies. Similarly, reward is calculated for all policies and presented in Table 4.9. The policy P_{1234} provides the minimum number of data transmissions and it is selected as optimal policy (*Algorithm* 4.6, line 5). Therefore, if a node is in state L_1, L_2, L_3 and L_4, then the node encodes the data using appropriate number of packet redundancy (depends on the state of a link, refer Table 4.2) to achieve maximum data delivery reliability, reduction in the number of data transmissions and thus, reduction in packet latency and energy consumption.

4.3 Performance Evaluation

In this section, the proposed mechanism is compared with a redundancy-based mechanism (proposed by Wu *et al.* [12]), the end-to-end erasure coding (EEEC) [14] and automatic retransmission (AR) in terms of average packet delivery delay, number of data

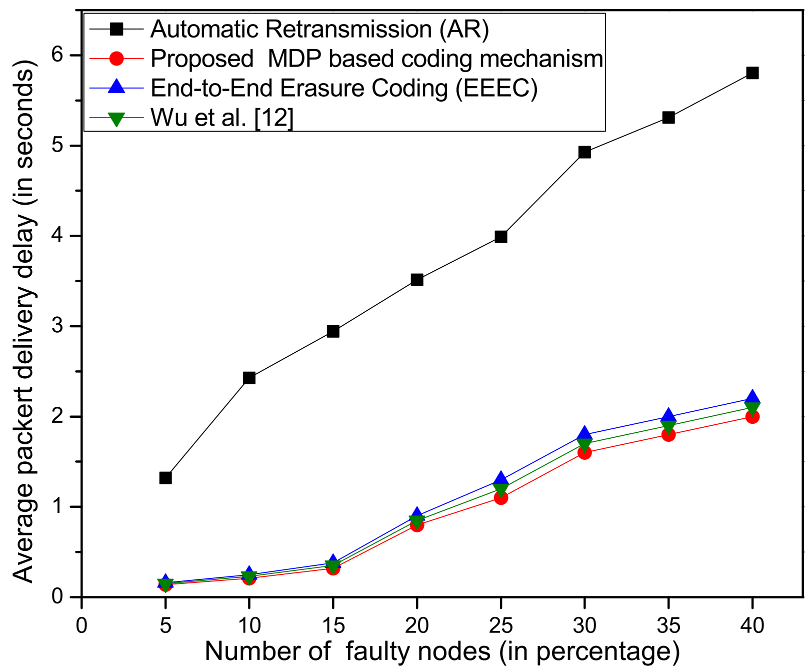


Figure 4.4: Average end-to-end delay of a packet by varying the number of faulty nodes

transmissions, energy consumptions and lifetime. The proposed mechanism is simulated using *Network Simulator-3* (NS3) [9]. The proposed mechanism has been compared with a retransmission-based approach to show the efficacy of the network coding approaches. The proposed mechanism and end-to-end erasure coding (EEEC) approach are redundancy-based mechanisms. In the EEEC approach, a node calculates the number of redundant packets according to the probability of successful delivery of a path. The proposed mechanism is hop-by-hop redundancy-based mechanism and it uses less number of transmissions in comparison to the end-to-end coding.

4.3.1 Experimental Setup

The sensor nodes are deployed randomly in an area A. Every node generates packets by sensing operation and relay towards an actor. A node determines the quality of link (refer *Algorithm 4.2*) and it determines the level of packet redundancy using *Algorithm 4.6* (if the decision is *Do Network Coding*). A *data collection round* is defined as the time interval

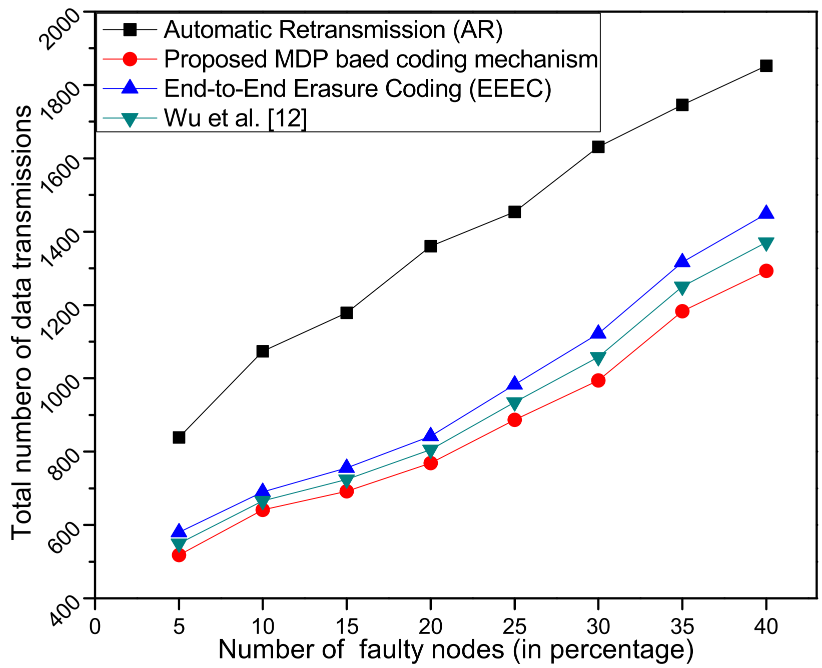


Figure 4.5: Total number of data transmissions (including retransmissions) in a round by varying the number of faulty nodes

when every node generates two packets and forwards to the actor. *Network lifetime* is defined as the number of data collection rounds in which all the nodes are connected [19, 118]. In the proposed mechanism, two packets are encoded in every data transmission. Actor is not an energy constrained node. In the automatic retransmission (AR) approach, a node will retransmit the packet till it gets corresponding acknowledgment. In the end-to-end erasure coding (EEEC) approach, a node calculates the number of redundant packets according to the probability of successful delivery of a path. In the redundancy based mechanism [12], a source node determines the redundancy level using the link loss rate of the worst link to the actor node. In the proposed mechanism, markov decision process analyzes the quality of link states and determines the applicability of network coding and redundancy level. Further, a node encodes the packets with appropriate number of redundancy level.

The simulation parameters are : number of sensor nodes in the network: 1000, transmission range of a node: $30m$ and initial energy of a node: $25KJ$. Further, the energy

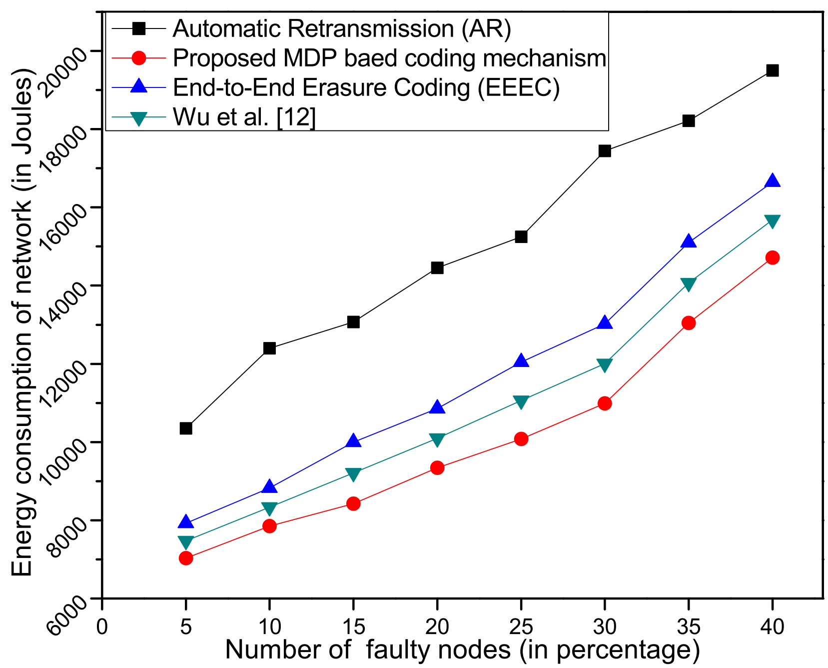


Figure 4.6: Total energy consumption of the network by varying the number of faulty nodes

consumption parameters are : E_{sense}, E_{rx} and E_{tx} which quantify energy consumptions by a sensor node in sensing, receiving and transmitting data over a distance $\hat{d}$, respectively [7]. $E_{sense} = \alpha_3$, $E_{rx} = \alpha_{12}$, $E_{tx} = \alpha_{11} + \alpha_2 \hat{d}^{\hat{n}}$, where, α_{11} represents energy consumption per bit by the transmitter electronics, α_2 denotes the energy dissipated in the transmit op-amp, α_{12} represents the energy consumption per bit by the receiver electronics, α_3 denotes the energy cost for sensing a bit and $\hat{n}$ represents the path loss component [7]. The energy parameters are adopted from [8] and the values are $\alpha_{11} = 0.937 \times 10^{-6}$ $Joules/bit$, $\alpha_{12} = 0.787 \times 10^{-6}$ $Joules/bit$, $\alpha_2 = 10 \times 10^{-12}$ $Joules/bit$, $\alpha_3 = 50 \times 10^{-9}$ $Joules/bit$, $\hat{d}$ = 85m and path loss exponent $(\hat{n}) = 2$.

4.3.2 Simulation Results and Discussion

In this section, the proposed mechanism is compared with a redundancy-based mechanism [12], end-to-end erasure coding (EEEC) [14] and automatic retransmission (AR) in terms

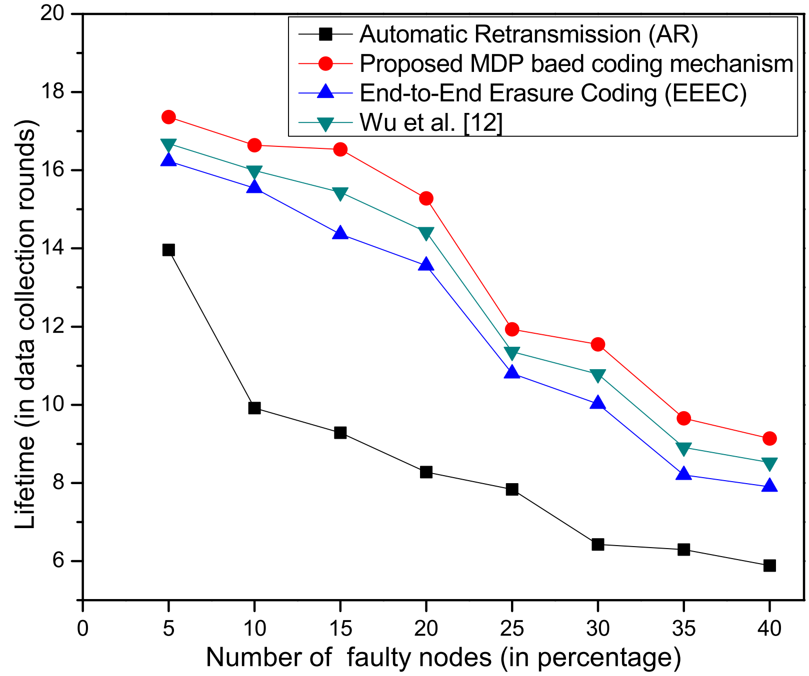


Figure 4.7: Lifetime of the network by varying the number of faulty nodes

of average packet delivery delay, number of data transmissions, energy consumptions and lifetime. Two scenarios are considered in the simulation to show the performance of the proposed MDP based opportunistic network coding mechanism. In *scenario 1*, number of nodes is 100 and mechanisms are compared in terms of performance metrics (as stated above) by varying the number of faulty nodes (from 5% to 40%) in the network. In *scenario 2*, number of faulty nodes is considered as 5%. The mechanisms are compared by varying the number of nodes (100 nodes to 1000 nodes) in the network. In Fig. Figures 4.4 to 4.7, the results are depicted by considering the simulation inputs as given in *scenario 1*. Further, the values (mentioned in *scenario 2*) are considered for the results as shown in Fig. Figures 4.8 to 4.11. In both scenarios, maximum number of retransmissions per packet is assumed as 5.

Fig. 4.4 shows the average delivery delay of a packet for one data collection round. In automatic retransmission (AR) approach, packet delay is more than other approaches. The delay increases due to increase in the number of retransmissions required for an undelivered packet. It can be observed from Fig. 4.4 that as the number of faulty nodes increases

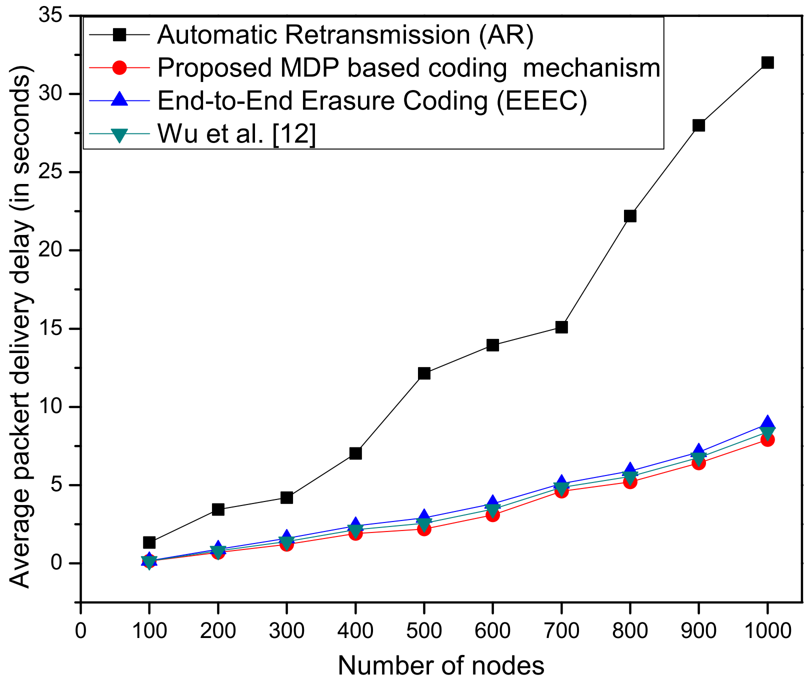


Figure 4.8: Average end-to-end delay of a packet (5% of nodes faulty)

in the network, the average delivery delay of a packet increases. Moreover, more number of data and acknowledgment packets are lost and every lost packet triggers a retransmission. The proposed MDP based mechanism provides reduction in delay as compared to other mentioned methods. In the proposed network coding based mechanism, packet drops reduce (when the faulty nodes increases in the network) due to opportunistic network coding decisions and adaptive maintenance of packet redundancy. Similarly, Fig. 4.8 shows the average delivery delay of a packet by varying number of nodes from 100 to 1000. In retransmission approach, as number of nodes increases, the delay also increases rapidly. This is due to the increase in number of hops to the actor from the source nodes.

Fig. 4.5 depicts the total number of data transmissions for one data collection round. In the automatic retransmission (AR), number of data transmissions increases with increase in the number of faulty nodes. In the end-to-end erasure coding (EEEC) [14], source estimates the number of redundant packets using the probability of successful delivery of a path. In EEEC, the number of packets transmitted in the first hop is much larger than in the last

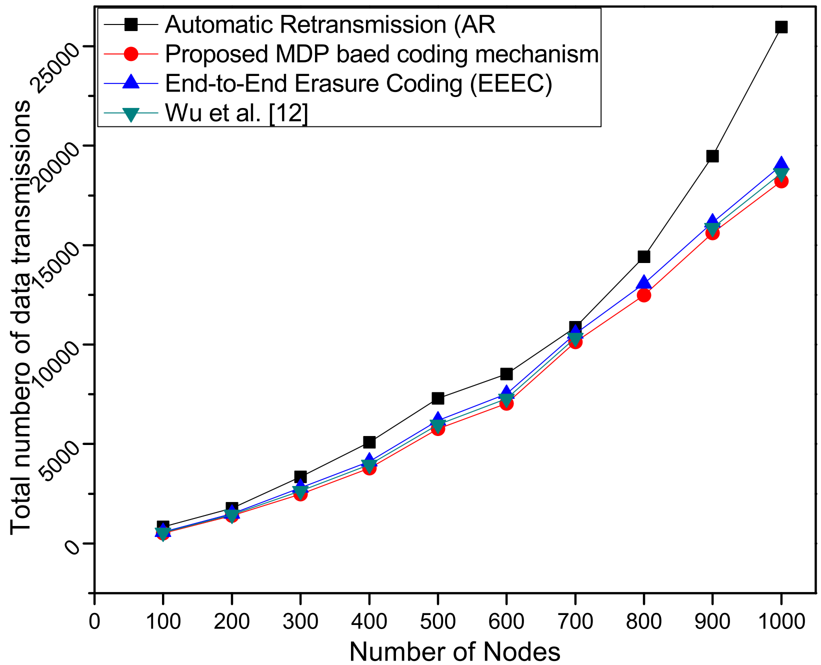


Figure 4.9: Total number of data transmissions (including retransmissions) in a round (5% of nodes faulty)

hop. This leads to increase in the number of data transmissions. In the redundancy-based mechanism (proposed by Wu et al. [12]), source node calculates the redundancy level based on the link loss rate of the worst link to the actor node. It results in increase of number of data transmissions at source node. The proposed mechanism reduces the number of data transmissions. This is due to determining the optimal packet redundancy level by analyzing the quality of link state using markov decision process. It is observed from Fig. 4.5 that the total number of data transmissions increases with the increase in the number of faulty nodes. However, the proposed mechanism determines the optimal policy which ensures the reliable data delivery (gathering) with less number of data transmissions as compared with other methods. Similarly, Fig. 4.9 shows the total number of data transmissions occurred in a data collection round by varying the number of nodes. As number of nodes increases, the number of hops to the actor also increases. It leads to more number of data transmissions.

Fig. 4.6 shows the performance of the proposed mechanism in terms of total energy

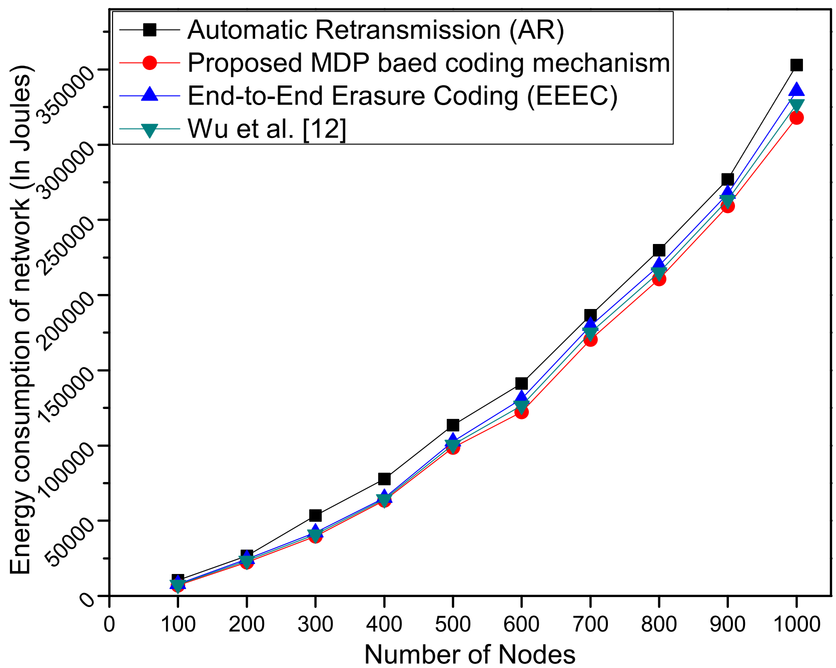


Figure 4.10: Total energy consumption of the network (5% of nodes faulty)

consumption in the network for one data collection round. The energy consumption of a network mainly depends on the total number of data transmissions. In the automatic retransmission approach, a node sends ACK packet after successful reception of every packet. This leads to increase in overhead in terms of number of data transmissions. In the EEEC, more number of data transmissions are transmitted than the proposed mechanism because EEEC is an end-to-end coding mechanism. It is observed from Fig. 4.6 that the proposed mechanism consumes less energy as compared with other methods because MDP mechanism reduces the number of data transmissions. Fig. 4.10 shows the comparison of the total energy consumption in the network by varying the number of nodes in the network. It can be seen from Fig. 4.6 that the energy efficiency of the network is improved with the proposed mechanism.

Fig. 4.7 and Fig. 4.11 show the network lifetime (in terms of the data collection rounds) for first scenario and second scenario, respectively. The proposed algorithm determines the link state at which network coding will be done and also the optimal level of packet

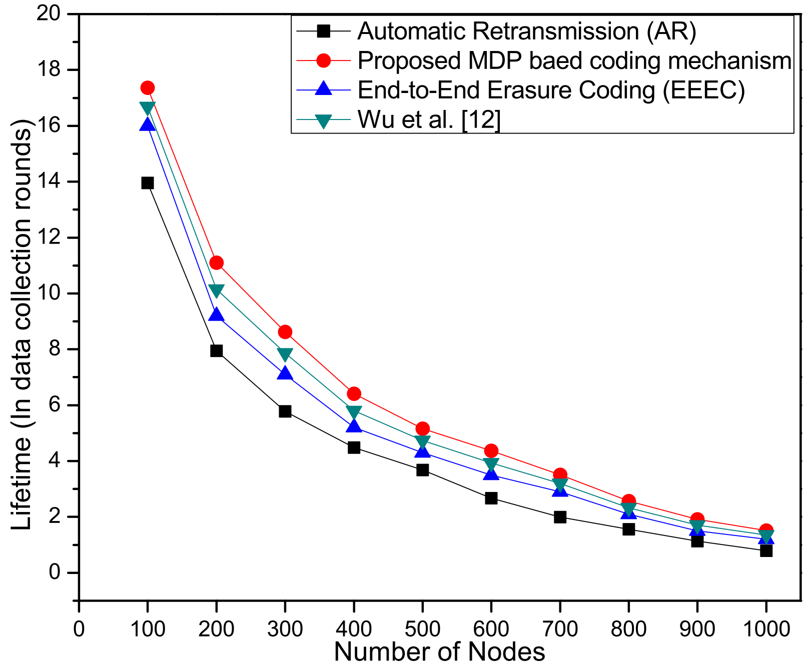


Figure 4.11: Lifetime of the network by varying the number of nodes in the network (5% of nodes faulty)

redundancy. This optimal policy reduces the number of data transmissions as compared to the other mentioned approaches. Therefore, energy consumption is reduced with the proposed mechanism. It can be observed from Fig. 4.7 and Fig. 4.11 that network remains connected more number rounds (i.e. improvement in lifetime) using the proposed MDP based network coding mechanism in comparison to the other existing approaches.

4.4 Summary

In this chapter, *a reliable data transmission mechanism using opportunistic network coding* has been proposed for wireless sensor and actor network with faulty nodes. The proposed mechanism improves the packet level reliability with stringent delivery delay requirement in the presence of faulty nodes. Further, network coding decision and level of packet redundancy are taken on the basis of quality of wireless links to achieve the energy efficient reliable data collection. Network performance parameters, such as average delivery delay, total number of data transmissions, energy consumptions and lifetime are taken for comparison of the proposed network coding mechanism with a redundancy based mechanism, end-to-end erasure coding and automatic retransmission approach. Further, network performance is observed with different number of faulty nodes in the network. From the simulation results, it has been observed that the average delivery delay of a packet is reduced using the proposed encoding approach. Further, it has been observed that the number of data transmissions in the network are reduced significantly using the proposed mechanism. Further, network lifetime is improved upto 21.5% and 56.22% with the proposed MDP based encoding mechanism in comparison to the end-to-end erasure coding and automatic retransmission mechanisms, respectively. In the next chapter, a routing metric is computed to reduce the packet loss by considering buffer occupancy, link quality and energy levels of a sensor node. Further, an intelligent delay and energy aware routing protocol has been proposed for selection of an efficient next hop node (in a routing path) to reduce the packet dropping rate.

Chapter 5

Adaptive Fuzzy Based Energy and Delay Aware Routing Protocol for Efficient Data Gathering in WSANs

Energy efficiency and reliability in data gathering are major issues in WSANs. In WSANs, a data collection protocol may satisfy the major application requirements, such as high data reliability and less delay along with energy efficiency. These requirements can be achieved by designing an energy efficient intelligent routing protocol. The sensor nodes may be deployed randomly in a dynamic environment [11]. This results in dropping of packets due to dynamic changes in quality of wireless links. The major reasons for dropping of packets may be bad wireless link quality, unavailability of free buffer at intermediate nodes and residual energy at nodes. To ensure data reliability, the dropped packets need to be retransmitted and this leads to more delay and energy consumption. Therefore, an energy efficient routing protocol may solve the above issues by taking efficient routing decisions while considering energy of a node, link quality and available buffer.

The key observation that motivates the work (presented in this chapter) is the improper selection of a next hop node in a routing path. This leads to dropping of packets. That means, selection of a node which has high available buffer, good wireless link quality, more residual energy and close distance (proximity) as next hop node reduces the packet dropping rate. Thus, reduction in packet dropping rate increases the data reliability, energy efficiency and reduces the delivery delay [15, 13]. In this chapter, a fuzzy based intelligent delay and energy aware routing mechanism has been proposed to improve the lifetime of the network. The fuzzy logic system (FLS) used for the implementation of multicriteria control strategies and it is capable of making decisions even with incomplete information. Fuzzy logic also used for blending different parameters and fuzzy rule set that may produce an optimal output [16, 17, 91]. In a routing path, selection of a node which has high available buffer, good wireless link quality, more residual energy and close distance as next hop node reduces the packet dropping rate. To handle this uncertainty, fuzzy logic system is used for the routing metric evaluation. In this chapter, fuzzy logic system is used for computing a routing metric value (called *chance of becoming next node*) by combining the parameters, such as residual energy, available buffer, quality of link and distance. Further, Dijkstra's algorithm has been adopted to find the routes where the determined routing metric value is considered as cost. The major contributions of this chapter are as follows:

- Design of an intelligent delay and energy aware routing protocol using fuzzy logic for a heterogeneous sensor actor network.
- A routing metric (i.e. a routing parameter) has been computed by considering the residual energy, link quality, available buffer and distance (proximity).
- Simulation has been performed to show the performance of the proposed protocol in terms of delay, number of retransmissions, energy consumption, lifetime, half nodes die, last node dies and network stability of the network.

The rest of the chapter is organized as follows. In section 5.1, motivation and problem formulation have been discussed. Network model has been introduced in section 5.2.1. Energy consumption model is presented in section 5.2.2. Fuzzy based energy aware routing mechanism has been proposed in section 5.2.3. In section 5.2.4, the procedure of fuzzy logic system is presented. Further, simulation results have been discussed in section 5.3. Finally, section 5.4 concludes the work.

5.1 Motivation and Problem Formulation

Design of an energy efficient routing protocol (for a heterogeneous sensor network where the environment changes dynamically) poses several challenges, such as data reliability, stringent delay and energy efficiency [15, 13]. Packet loss is one of the major factors that results in the above mentioned network. Retransmission of the lost packet may achieve data reliability, but it increases the delay and energy consumption [13]. In a heterogeneous sensor actor network, packet loss (at a node) may occur due to bad link quality, overflow of the buffer or energy levels of the node. Packet loss may be reduced by designing an efficient routing metric (or a routing parameter) which is computed based on the link quality, available buffer and energy level. In the PRD (*predicted remaining deliveries*) routing mechanism [15], a routing metric is computed using the residual energy, link quality, delay and distance. However, in a large scale sensor network, more number of packets are dropped by the nodes which are near to the actor node due to the buffer overflow. In the PRD routing metric [15], the number of expected transmissions and link bandwidth are considered to obtain the delay (the time spent in transmitting the packet) which is a constant for the network where bandwidth is assumed as same for all the links. End-to-end delay increases with the number of dropped packets. More number of packets dropped due to the poor link quality and buffer overflow. Therefore, available buffer is an important parameter that needs to be considered in the computation of a routing metric. PRD routing metric does not address the available buffer as a parameter. The proposed mechanism has been compared with the PRD routing mechanism to show that the available buffer is an important parameter in the computation of routing metric. In this work, a *Fuzzy based Energy-Aware Routing Mechanism* (FEARM) has been proposed to improve the performance of a heterogeneous sensor network by considering the residual energy, link quality, available buffer and distance (proximity). Further, Dijkstra's algorithm is adopted to find the efficient routes using a computed routing metric (or a parameter).

Table 5.1: Summary of Notations

Symbol	Description
n	Number of sensor nodes in the network
A	Local actor node
E_i	Residual energy of a node i
B_i	Available (free) buffer of a node i
L_{ji}	Quality of a link from node j to node i
D_{ji}	Distance from node j to node i
$d[u]$	cost of the path to node u
$f[v]$	Predecessor of node v
p_j	Fuzzy output value (routing metric value)
$\mu(X)$	membership value of input variable X

5.2 Fuzzy Based Energy-Aware Routing Mechanism for a Heterogeneous Sensor Actor Network

In this section, the network model is introduced. An energy consumption model has been presented. Further, a routing metric has been computed using the proposed fuzzy based routing mechanism and Dijkstra's algorithm has been adopted to find the routes in the proposed mechanism.

5.2.1 Network Model

A heterogeneous sensor actor network is represented as a graph $G(V \cup A, E)$ as shown in Fig. 5.1, where $V = \{v_1, v_2, v_3...v_n\}$ is a set of n number of sensor nodes, E is a set of links and A is a local actor node (for large scale network). Heterogeneity characteristics, such as battery capacity, quality of link, buffer capacity are considered in this work. A node (v_i) has different levels of residual energy (E_i), buffer capacity (B_i), associated with link quality (from node j to node i is L_{ji}) and distance (from node j to node i is D_{ji}). Sensor nodes generate data (by sensing operation) and transfer the data to a local actor node (A) through an efficient routing path. A node (v_i) finds the efficient routing path by determining the fuzzy output value (p_j) (where, v_j is one of the possible next hop nodes) for next hop node. The fuzzy output value (indicates the value of routing parameter or

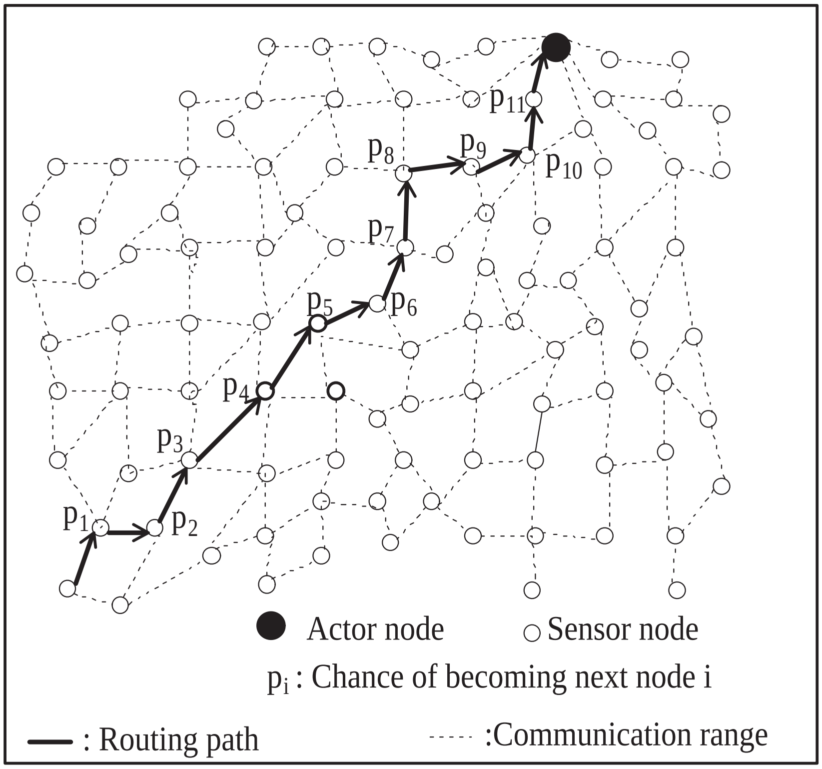


Figure 5.1: The illustration of energy-aware routing path (i.e. data gathering path) in the wireless sensor and actor network

chance of becoming next node) is a function of residual energy (E_j), available buffer (B_j), quality of link (L_{ij}) and distance (D_{ij}). Every node forward its residual energy, link quality and available buffer to its neighbor nodes to compute the routing metric. Some amount of control packet overhead will be introduced due to the above metric. The notations used in this chapter are described in Table 5.1.

5.2.2 Energy Consumption Model

This section presents the same energy consumption model as in [7, 8]. Energy consumption parameters for sensing, receiving and transmitting data (over a distance d) are E_s, E_r and E_t, respectively. According to the path loss model, signal strength is reduced by $1/d^{\hat{n}}$, where $\hat{n}$ is the path loss exponent. The energy consumption are given as follows:

$$E_s = \alpha_3$$

$$E_r = \alpha_{12}$$

$$E_t = \alpha_{11} + \alpha_2 d^{\hat{n}}$$

where, α_3 $(joules/bit)$ is the energy consumption for sensing operation, α_{12} $(joules/bit)$ is the energy consumed by the receiver electronics, α_{11} $(joules/bit)$ is the energy/bit consumed by the transmitter electronics and α_2 $(joules/bit/meter^2$ for $\hat{n} = 2)$ is energy dissipated in the transmit op-amp [7]. In this work, energy consumptions are computed for an event centric applications [8, 119]. Let, l is the size of data generated by a sensor per event and β is the average rate of events occur per unit time [32]. Therefore, the energy consumption for sensing operation till time t will be $lt\beta\alpha_3$. A node's energy consumption for sensing operation and transmitting sensed data in time t is $lt\beta(E_s + E_t)$.

5.2.3 Fuzzy Based Energy-Aware Routing Mechanism: FEARM

In this section, the proposed fuzzy based delay and energy aware routing mechanism (FEARM) is presented. Every node finds the fuzzy output values for neighbor nodes. Further, Dijkstra's algorithm [120, 121] has been adopted to find the efficient routes (i.e. data gathering routes) in the proposed mechanism.

In *Algorithm 5.1*, every node finds the fuzzy output values to its neighbor nodes based on residual energy, link quality, available buffer and distance ($lines$ 1 to 5). *Algorithm 5.4* determines the fuzzy output value (chance of becoming next node) and it is explained in the next section. In *Algorithm 5.1*, $lines$ 6 to 15 computes the efficient paths based on the fuzzy output value. In *Algorithm 5.2*, cost of the path from the starting node to a node u is $d[u]$ and it is initialized to *zero* (for starting node, cost is *one*). In *Algorithm 5.3*, if node u precedes node v, then the cost to node v is $d[v] \leftarrow d[u] * p_v$, where $d[u]$ is cost of the path to u and p_v is the fuzzy output value (chance of becoming next node) of node v. The detailed explanation of *Algorithm 5.4* is given in section 5.2.4.

Algorithm 5.1 Fuzzy Based Energy-Aware Routing Mechanism

Input: $G(V, E)$, S, n, Residual Energy (E), Available Buffer (B), Distance(D), Link Quality (L)

Output: Energy-aware routing paths

for $i = 1$ to n **do**
  for each vertex $j \in Adj[i]$ **do**
    $P[i][j]$ =Fuzzy_Logic_System(E_j, B_j, D_{ij}, L_{ij})
  end for
end for
Initialize(G, S) //refer *Algorithm* 5.2
$S \leftarrow \emptyset$
$Q \leftarrow V[G]$
while $Q \neq \emptyset$ **do**
  $u \leftarrow MAX(Q)$
  $S \leftarrow S \cup \{u\}$
  for each vertex $v \in Adj[u]$ **do**
    Relax(u, v, $P[u][v]$) //refer *Algorithm* 5.3
  end for
end while

Algorithm 5.2 Initialize(G, S)

for each vertex $u \in V[G]$ **do**
  $d[u] = 0$
  $f[u] = NULL$
end for
$d[s] = 1$

Algorithm 5.3 Relax(u, v, p_v)

if $d[v] < d[u] * p_v$ **then**
  $d[v] \leftarrow d[u] * p_v$
  $f[v] \leftarrow u$
end if

Table 5.2: Fuzzy input/output variables and their linguistic variables

Input/Output variables	**Linguistic variables**
Residual energy	Low, Medium, High
Available buffer	Low, Medium, High
Link quality	Poor, Average, Good
Distance	Far, Adequate, Close
Chance of becoming next node	Low, Weak, Medium, Strong, Very Strong

Algorithm 5.4 Fuzzy_Logic_System(E_j, B_j, D_{ij}, L_{ij})

Output: Chance of becoming node-j as next node (p_j)

```
1:  Empty the list l < value, membershiplevel >
2:  Find membership values (μ(E), μ(B), μ(D) and μ(L)) and linguistic levels using Tri-
    angular membership function.
3:  DR ={A rule set with all possible combinations of determined linguistic levels (from
    line − 2}
4:  for each rule in DR do
5:      if μ(E), μ(B), μ(D), μ(L) then fit the membership levels of this rule
6:          Add an entry to the list l with
7:          value = maximum(μ(E), μ(B), μ(D), μ(L))
8:          membershiplevel = output membership level of this rule
9:      end if
10: end for
11: p_j =Defuzzify(l)
12: return p_j
```

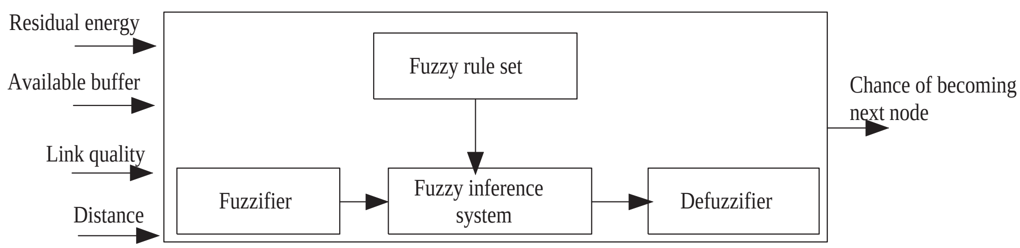


Figure 5.2: Fuzzy Logic System

5.2.4 Fuzzy Logic System for Determining Routing Metric

The fuzzy logic system (refer Fig. 5.2) has four modules, namely, fuzzifier, fuzzy rules set, fuzzy inference system and defuzzifier. In the fuzzifier module, membership values ($\mu(E), \mu(B), \mu(D), \mu(L)$) and membership levels (linguistic levels) for a given crisp input parameters (E, B, D, L) are determined using the membership functions. Fuzzy rules set is a set of *if-then* rules which are generated using *Mamdani fuzzy inference* model [122] (refer Table 5.3). A fuzzy rule can be represented as follows: *If* x is A_1 *AND* y is A_2 *Then* z is B_1, where A_1, A_2 and B_1 are the linguistic variables of fuzzy sets x, y and z, respectively. Fuzzy inference system infers the fuzzy output by applying the fuzzy rule set to the membership functions. Finally, *defuzzifier* transforms the fuzzy output value to a crisp value. In this work, *Center of Area* (COA) method has been used in defuzzification

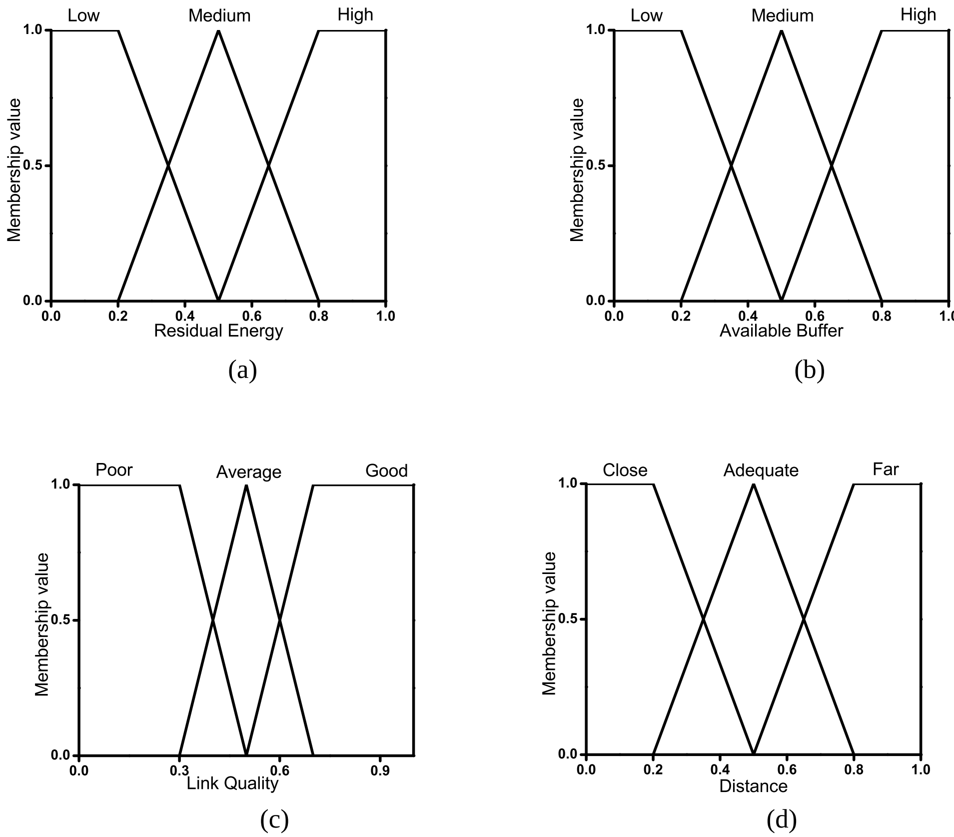


Figure 5.3: Membership functions for (a) Residual energy (b) Available (free) buffer (c) Quality of link (d) Distance

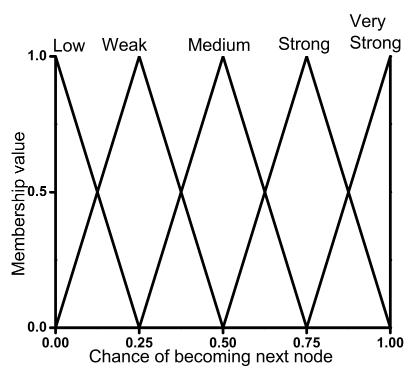


Figure 5.4: Membership function for output variable *chance of becoming next node*

process [17, 80, 81].

In this work, the input variables for the fuzzy logic system are residual energy (E), available buffer (B), quality of link (L) and distance (D). The fuzzy sets that describe the residual energy and available buffer are shown in Fig. 5.3(a) and Fig. 5.3(b), respectively.

The linguistic variables for these input variables are *low*, *medium* and *high*. The fuzzy set that describes the quality of link input variable is shown in Fig. 5.3(c). The linguistic variables for this input variable are *poor, average* and *good*. The fuzzy set that describes the distance input variable is shown in Fig. 5.3(d). The linguistic variables for this input variable are *close, adequate* and *far*. The only fuzzy output variable is the *chance of becoming next node* and is depicted in Fig. 5.4. Five linguistic variables *low, week, medium, strong* and *very strong* are considered for fuzzy output variable. The fuzzy set input and output variables and their corresponding linguistic variables are shown in Table 5.2. In this work, triangular membership functions are used for all linguistic variables of input and output variables. Triangular membership functions are used to reduce computational costs (conversion from crisp value to the fuzzy value) and to simplify implementation [80].

Fuzzy rules can be generated using experimental data or heuristic data. In this work, fuzzy rules are generated using heuristic fuzzy rule generation method which follows the following principle: a node with high residual energy, high available buffer, good quality of link and close distance has the highest *chance of becoming the next node* in a routing path. Morevoer, based on the four input variables and three membership levels, 81 fuzzy *if-then* mapping rules are defined in Table 5.3. All fuzzy rules contribute to some degree to the final decision. However, some rules do not contribute significantly to the final decision and can be eliminated. In the literature, most of the works [17, 80, 81, 82, 16] have been used all the fuzzy rules to evaluate the final decision. The fuzzy output variable is derived by predefined fuzzy rules and membership functions. This derived fuzzy output variable has been transformed to a single crisp value (which indicates the *Chance of becoming the next node* in the routing path) using center of area (COA) method in the defuzzification process.

The working process of fuzzy logic system has been presented in *Algorithm 5.4*. In *Algorithm 5.4*, a list $l < value, membershiplevel >$ is initialized to zero (*line-1*). Associate membership values and linguistic levels are determined for every given fuzzy input variable using membership functions (refer Fig. 5.3) (*line-2*). Further, *Determined Rule* set (DR) is constructed. (DR) is a set of all possible combinations of determined linguistic levels (*line-3*). For each rule in DR, the associate output linguistic variable will be found using predefined fuzzy rule set (refer 5.3). An entry will be added to the list l which has maximum membership value (*line-4 to line-10*). Finally, defuzzifier transforms the list l to a single output value (*line-11*).

Table 5.3: Fuzzy decision making rules

Inputs				**Output**
Energy	**Available buffer**	**Distance**	**Link Quality**	**Chance of becoming next node**
Low	Low	Far	Poor	Low
Low	Low	Far	Average	Low
Low	Low	Far	Good	Weak
Low	Low	Adequate	Poor	Low
Low	Low	Adequate	Average	Weak
Low	Low	Adequate	Good	Medium
Low	Low	Close	Poor	Weak
Low	Low	Close	Average	Weak
Low	Low	Close	Good	Medium
Low	Medium	Far	Poor	Low
Low	Medium	Far	Average	Low
Low	Medium	Far	Good	Low
Low	Medium	Adequate	Poor	Low
Low	Medium	Adequate	Average	Weak
Low	Medium	Adequate	Good	Medium
Low	Medium	Close	Poor	Low
Low	Medium	Close	Average	Weak
Low	Medium	Close	Good	Medium
Low	High	Far	Poor	Low
Low	High	Far	Average	Low
Low	High	Far	Good	Low
Low	High	Adequate	Poor	Low

Continued on next page

Table 5.3 – continued from previous page

Energy	Available buffer	Distance	Link Quality	Chance of becoming next node
Low	High	Adequate	Average	Weak
Low	High	Adequate	Good	Medium
Low	High	Close	Poor	Weak
Low	High	Close	Average	Medium
Low	High	Close	Good	Strong
Medium	Low	Far	Poor	Low
Medium	Low	Far	Average	Low
Medium	Low	Far	Good	Weak
Medium	Low	Adequate	Poor	Low
Medium	Low	Adequate	Average	Weak
Medium	Low	Adequate	Good	Medium
Medium	Low	Close	Poor	Low
Medium	Low	Close	Average	Weak
Medium	Low	Close	Good	Medium
Medium	Medium	Far	Poor	Weak
Medium	Medium	Far	Average	Medium
Medium	Medium	Far	Good	Strong
Medium	Medium	Adequate	Poor	Medium
Medium	Medium	Adequate	Average	Strong
Medium	Medium	Adequate	Good	Very Strong
Medium	Medium	Close	Poor	Medium
Medium	Medium	Close	Average	Strong
Medium	Medium	Close	Good	Very Strong
Medium	High	Far	Poor	Weak
Medium	High	Far	Average	Medium
Medium	High	Far	Good	Strong

Continued on next page

Table 5.3 – continued from previous page

Energy	Available buffer	Distance	Link Quality	Chance of becoming next node
Medium	High	Adequate	Poor	Medium
Medium	High	Adequate	Average	Strong
Medium	High	Adequate	Good	Very Strong
Medium	High	Close	Poor	Medium
Medium	High	Close	Average	Strong
Medium	High	Close	Good	Very Strong
High	Low	Far	Poor	Low
High	Low	Far	Average	Low
High	Low	Far	Good	Weak
High	Low	Adequate	Poor	Low
High	Low	Adequate	Average	Weak
High	Low	Adequate	Good	Medium
High	Low	Close	Poor	Low
High	Low	Close	Average	Weak
High	Low	Close	Good	Medium
High	Medium	Far	Poor	Weak
High	Medium	Far	Average	Medium
High	Medium	Far	Good	Strong
High	Medium	Adequate	Poor	Medium
High	Medium	Adequate	Average	Strong
High	Medium	Adequate	Good	Very Strong
High	Medium	Close	Poor	Medium
High	Medium	Close	Average	Strong
High	Medium	Close	Good	Very Strong
High	High	Far	Poor	Weak
High	High	Far	Average	Medium

Continued on next page

Table 5.3 – continued from previous page

Energy	Available buffer	Distance	Link Quality	Chance of becoming next node
High	High	Far	Good	Strong
High	High	Adequate	Poor	Medium
High	High	Adequate	Average	Strong
High	High	Adequate	Good	Very Strong
High	High	Close	Poor	Medium
High	High	Close	Average	Strong
High	High	Close	Good	Very Strong

5.3 Performance Evaluation

In this section, the proposed Fuzzy based Energy Aware Routing Mechanism (FEARM) has been compared with a PRD (*predicted remaining deliveries*) routing mechanism [15] and a retransmission based approach ([12]) in terms of average end-to-end delay, total number of retransmissions, lifetime, total energy consumption, data collection rounds at which half of the nodes die, data collection rounds at which all nodes die and network stability.

5.3.1 Simulation Environment

Simulation results are presented by considering the different combinations of the fuzzy input variables, (i.e. FEARM with four parameters (residual energy, available buffer, link quality and distance), FEARM with three parameters (residual energy, available buffer and link quality), FEARM with residual energy and link quality and FEARM with residual energy and available buffer). The simulation is performed using the *Network Simulator 3* [9]. The simulation parameters are shown in Table 5.4 [7, 8]. In the simulation, heterogeneity is considered in the network as follows: all nodes have different energy capacity $(20KJ$

Table 5.4: Simulation Parameters

Parameter	**Value**
Number of sensor nodes	100 to 1000 nodes
Transmission range	30 $meters$
Initial energy of a node	$20KJ$ to $25KJ$
$E_s = \alpha_3$	$\alpha_3 = 50 \times 10^{-9}\ Joules/bit$
$E_r = \alpha_{12}$	$\alpha_{12} = 0.787 \times 10^{-6} Joules/bit$
$E_t = \alpha_{11} + \alpha_2 d^{\hat{n}}$	$\alpha_{11} = 0.937 \times 10^{-6} Joules/bit,$ $\alpha_2 = 10 \times 10^{-12} Joules/bit/meter^2$ $d = 85\ meters$
Path loss exponent ($\hat{n}$)	2
Data generated per event by a node (l)	$960 bits$
Average rate of events occur per unit time (β)	100

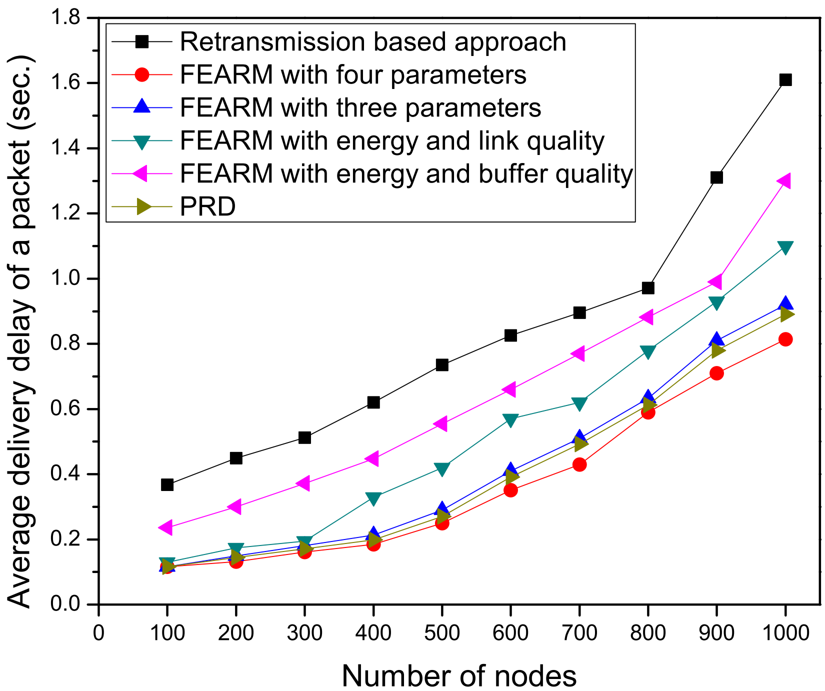


Figure 5.5: Average end-to-end delay (in *seconds*)

to $25KJ$), buffer capacity ($2K$ to $2.5K$ bytes) and all links are assigned with link qualities between 0 to 1 (follows uniform distribution). The size of generated data per event

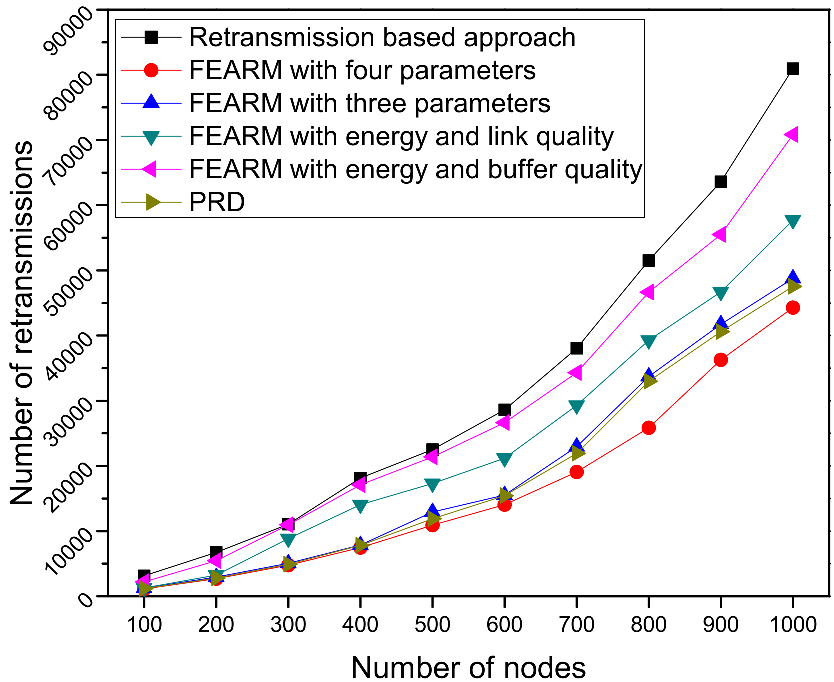


Figure 5.6: Total number of retransmissions

is considered as 960 $bits$ (refer Table 5.4). To achieve high data reliability, hop-by-hop (link layer) retransmissions are performed till a packet reaches to the next hop node. In a data collection round, every node generates one packet and transmit to the local actor. In this simulation, lifetime of the network is defined as the number of data collections rounds completed till the first node die [118, 19].

5.3.2 Simulation Results and Discussions

Fig. 5.5 and Fig. 5.6 show the comparison of average end-to-end delay of a packet and number of retransmissions among fuzzy based delay and energy aware routing mechanisms (FEARM with four parameters, FEARM with three parameters, FEARM with *residual energy and link quality* and FEARM with *residual energy and available buffer*), PRD routing mechanism [15] and a retransmission based approach [12]. In the proposed mechanism, routing decision has been taken using the proposed fuzzy logic system to reduce the packet drops. Reduction of packet dropping rate incurs reduction in the packet retransmission rate

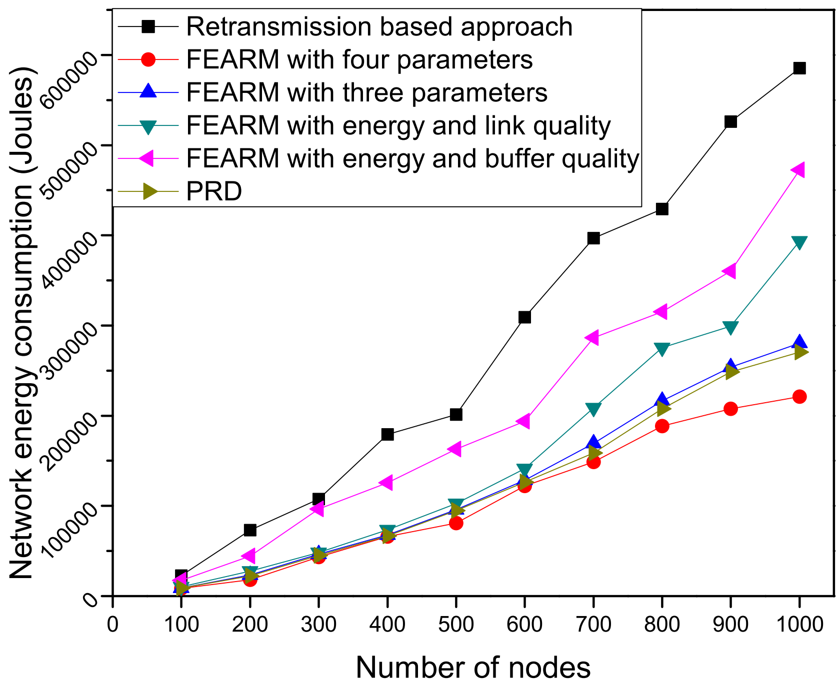


Figure 5.7: Total energy consumption (in *Joules*)

in a data gathering route. Reduction of packet retransmission rate further reduces not only the energy consumption, but also delivery delay of a packet. It has been observed from the Fig. 5.5 that the fuzzy based energy-aware routing mechanism (FEARM), FEARM with three parameters (residual energy, available buffer and link quality), FEARM with *residual energy and link quality* and FEARM with *residual energy and available buffer*, reduces the average delay upto 58.78%, 53.54%, 41.96% and 23.58%, respectively, in comparison to the retransmission based approach. PRD routing mechanism [15] does not consider the available buffer as a parameter in the computation of the routing metric. In a large scale sensor network, the nodes near to the actor node will be overburdened and drop the packets due the overflow of the buffer. These dropped packets need to be retransmitted and this leads to the increase in delay and number of retransmissions. It can be seen from Fig. 5.5 that the proposed fuzzy based routing mechanism (with four parameters) reduces the average delay upto 7.3% in comparison to the PRD routing mechanism [15]. It can be seen from Fig. 5.6 that the fuzzy based energy-aware routing mechanism, FEARM with three param-

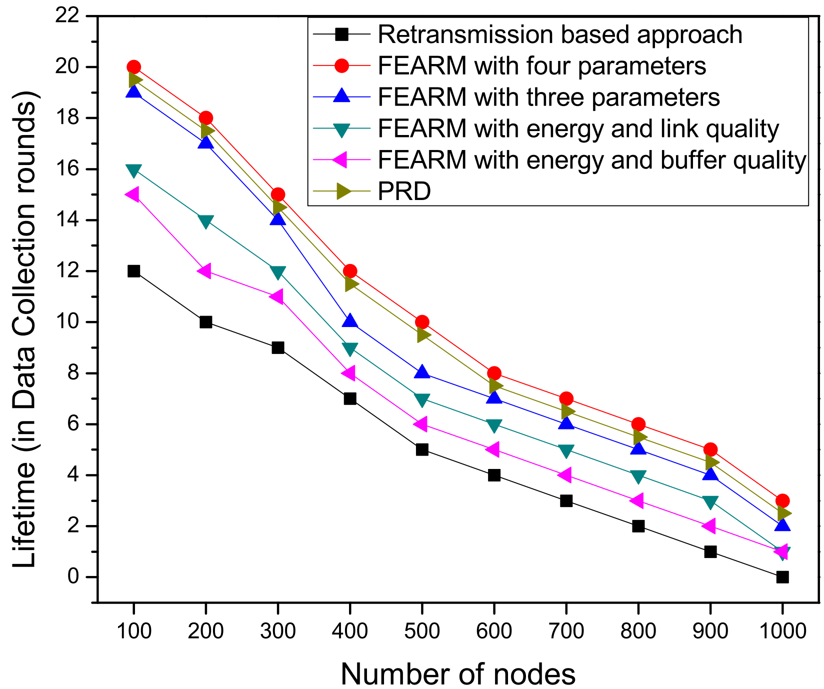


Figure 5.8: Lifetime of the network (in *data collection rounds*)

eters (residual energy, available buffer and link quality), FEARM with *residual energy and link quality* and FEARM with *residual energy and available buffer*, reduces the number of (average) retransmissions upto 52.89%, 46.6%, 30.45% and 11.19%, respectively, in comparison to the retransmission based approach [12]. The proposed mechanism reduces the number of (average) retransmissions upto 8.2% in comparision to the PRD metric [15].

Fig. 5.7 shows the comparison of energy consumption of the network among the proposed mechanisms (with different parameters) and retransmission based approach. Energy consumption of the network increases as number of nodes increases. In the PRD routing mechanism and retransmission based approach, the number of retransmissions are increased due to increase in the number of packet drops as compare to the proposed mechanism. The proposed mechanism provides an efficient routing path and this reduces the packet dropping rate in the network. The proposed mechanism achieves less energy consumption (i.e. an average reduction upto 61.82% and 9.6%) in comparison to the retransmission based approach and PRD mechanism, respectively.

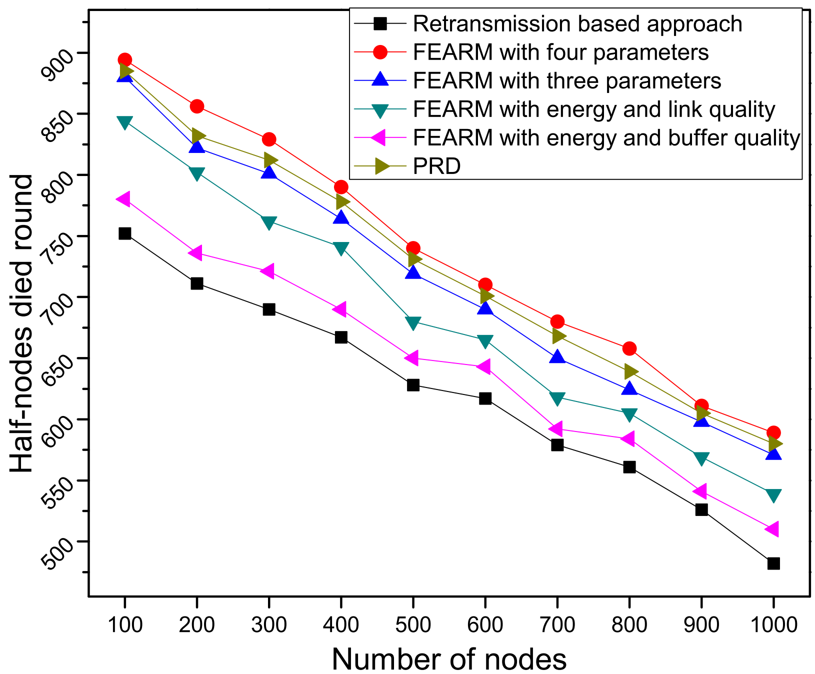


Figure 5.9: Half of the nodes die (in *data collection rounds*)

Fig. 5.8 and Fig. 5.9 illustrate the lifetime (when first node dies) and half of the nodes die round (the round in which 50% of nodes die), respectively. The lifetime of the network is improved with the proposed mechanism as compared to the retransmission based approach. *Half of the nodes die round* is a data collection round at which half of the sensor nodes are died in the network [17, 123]. It can be observed from Fig. 5.8 that as number of nodes increases, the lifetime of the network reduces. The (average) *half of the nodes die round* (in data collection rounds) is increased upto 18.38 % (refer Fig. 5.9) (i.e. lifetime is increased) with the proposed mechanism in comparison to the retransmission based approach. From the Fig. 5.8 and Fig. 5.9, it can be observed that a significant lifetime improvement has been achieved with the proposed mechanism in comparison to the PRD routing metric.

Fig. 5.10 shows the stability of the network in terms of number of data collection rounds. Network stability is defined as the difference (in rounds) between first node dies round and half of the nodes die round [82]. Fig. 5.10 shows that the proposed mecha-

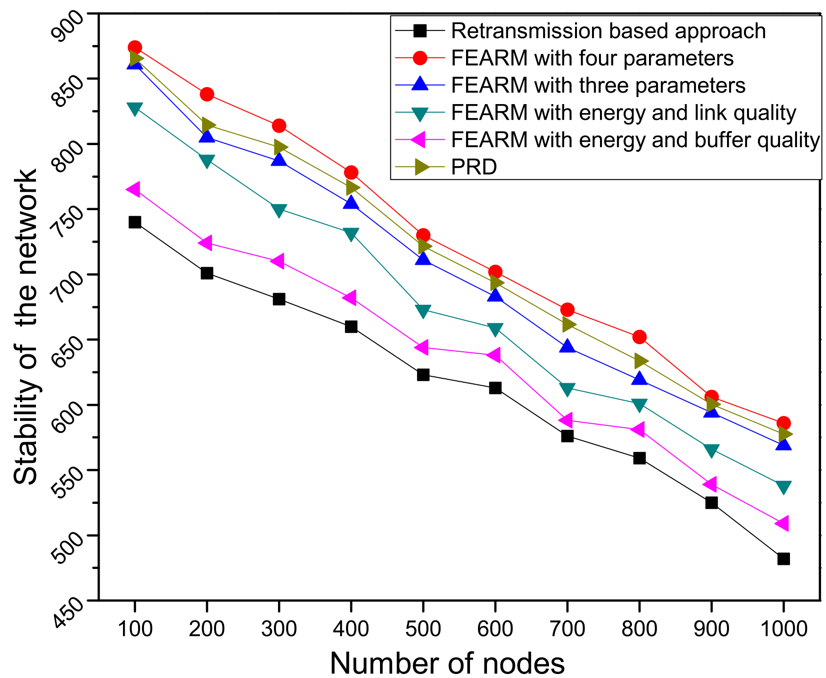


Figure 5.10: Network stability (in *data collection rounds*)

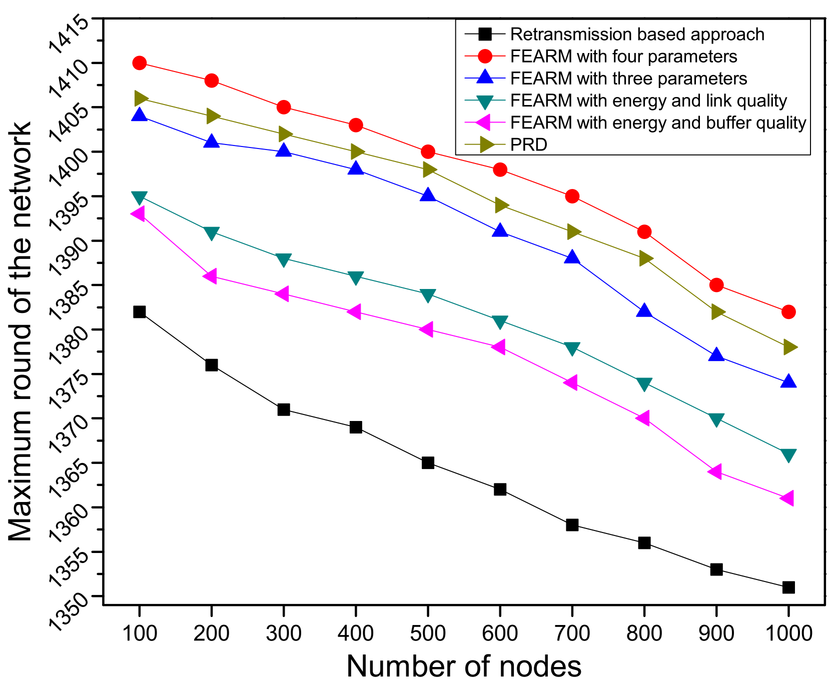


Figure 5.11: Data collection round at which last node dies

nism improves the (average) stability of the network upto 17.72 % in comparison to the existing approach [12]. The fuzzy based energy-aware routing mechanism (FEARM) with three parameters (residual energy, available buffer and link quality), FEARM with residual energy and link quality and FEARM with residual energy and available buffer improve the network stability upto 14%, 9.42%, 3.59%, respectively, in comparison to the existing approach [12]. The proposed mechanism improves the network stability in comparison to the PRD routing metric (refer Fig. 5.10). Fig. 5.11 illustrates the data collection round at which last node dies. In the PRD mechanism and retransmission based approach, last node dies much before in comparison to the proposed mechanism (refer Fig. 5.11). Therefore, residual energy, available buffer, link quality and distance (proximity) are the important parameters for choosing the next hop node in a routing path.

5.4 Summary

In this chapter, a delay and energy aware fuzzy based routing protocol is proposed to take an efficient routing decision in a heterogeneous sensor network. A node's residual energy, link quality, free buffer and distance (proximity) are considered as the fuzzy input variables. The network performance has been measured with different combination of fuzzy input variables. The proposed routing protocol reduces the packet dropping rate and this leads to reduction in delay and energy consumption. Simulation results show that there is a reduction in (average) delay upto 58.78% with the proposed fuzzy based mechanism. Network (average) energy consumption is reduced upto 61.82% with the proposed mechanism. Further, it has been shown that lifetime of the network in case of *fuzzy with four parameters* is better than *fuzzy with three* and *two parameters*. It has been observed that the network stability is improved upto 17.72% in comparison to the retransmission based data forwarding approach. In the next chapter, energy harvesting nodes are considered in WSANs for prolonging the network lifetime. Node survivability has been addressed (in the next chapter) to investigate the reliable data transmissions in energy harvesting sensor and actor networks.

Chapter 6

Fuzzy Logic Based Adaptive Duty Cycling Algorithm for Sustainability in Energy Harvesting WSANs

Wireless Sensor Actor Networks (WSANs) remain operational for limited amount of time due to limited battery capacity of a sensor node [2]. A long operational lifetime of WSAN is required for applications, such as monitoring forest fires, nuclear plants, object tracking and military surveillance [2, 33, 19, 79, 4, 124]. Energy harvesting technology provides a long lasting lifetime in WSAN. A node recharges battery from the charging vehicles or from the natural resources, such as solar light, thermal, wind and vibration [4, 35, 18, 125, 126]. A node is capable of harvesting energy from sunlight. The volume of harvested energy varies dynamically with the weather conditions over time [18]. The residual energy may reduce with usages and time. This results in temporary disconnection of the nodes from the network. The temporal-disconnected nodes may join into the network in the next energy available time slot. Hence, the node should sustain (survive) till next available period of the energy source for improving the lifetime of the network. Node sustains till the next recharge round by using the energy conservation techniques, such as duty cycling, load balancing and data aggregation [19].

The existing models (such as [4, 35]) are based on the wireless charging by a mobile charger. Wireless charging technology may not be suitable for applications (like deep forest) where a mobile charger cannot travel to replenish the energy of sensor nodes. In this type of applications, solar energy harvesting technology provides a long lasting lifetime to the network. Further, the energy resources are typically dynamic and uncontrolled [127]. Achieving network sustainability with solar based harvesting technology is one of the challenging issues which has not been addressed adequately in the literature.

Duty cycle and load balancing techniques collectively improve the network sustainability and throughput. Duty cycled wireless sensor network is categorized into two types: *random duty cycling* and *coordinated duty cycling* [54]. In the random duty cycled wireless sensor network, the sensor nodes are turned *on* and *off* randomly on their own. In a coordinated duty cycled wireless sensor network, the sensor nodes coordinate with each other (via exchanging the information) to follow *on* and *off* (i.e.*active* and *sleep*) schedule. The coordinated duty cycled WSN is potentially efficient for communication in terms of synchronization among nodes and network connectivity [54, 33].

In this chapter, a fuzzy based adaptive duty cycling algorithm has been proposed to achieve the network sustainability in a tree-based energy harvesting sensor actor network. Moreover, a prediction model has been proposed to estimate energy consumption (i.e. residual energy) for a future interval of time. A node estimates duty cycle value using fuzzy logic system for a future time period by considering residual energy, predicted harvesting energy (for a future time slot) and predicted residual energy (for a future time slot). Further, a switching decision has been taken by a node based on the predicted duty cycle value of a node for survival. Coordinated duty cycling is considered in this work. To achieve efficient communication, *active and sleep schedules* are forwarded to the children nodes. The major contributions of this chapter are as follows:

- Design of a fuzzy based adaptive duty cycling algorithm to achieve sustainability in an energy harvesting sensor actor network.
- Prediction of duty cycle using fuzzy logic by considering current residual energy, futuristic harvesting energy and residual energy.
- Simulation results are presented to show the efficacy of the proposed mechanism to provide improved network sustainability and successful data reception.

The rest of the chapter is organized as follows. Section 6.1 presents the motivation of our work and problem formulation. In section 6.2.1, a network model is introduced. In section 6.2.2, the energy consumption has been computed for future time period. In section 6.2.3, energy consumption of the network has been derived for the proposed network model. Further, a fuzzy based adaptive duty cycle algorithm has been proposed in section 6.2.4. Section 6.2.5 presents the fuzzy logic system procedure. Section 6.3 discusses the performance evaluations of the proposed mechanism and finally, section 6.4 concludes the work.

6.1 Motivation and Problem Formulation

In an energy harvesting sensor networks, a node harvests energy from ambient sources, such as solar light, thermal, wind and vibration. The amount of harvested energy may vary with weather conditions or seasonal changes. Therefore, the volume of harvested energy may be less than the energy consumption of a node in a time slot. This results in temporary disconnection of the nodes. A node may join into the network in the next (solar) energy available period. The temporary disconnected nodes affect the quality of service of the network. To address the temporary disconnection of the nodes, we have proposed a fuzzy based adaptive duty cycle mechanism for sustainability of the nodes till next recharge period. Fuzzy logic system (FLS) is used to estimate the expected duty cycle of the node for future time slot. Current residual energy, predicted harvesting energy (for a future time slot) and expected residual energy (for a future time slot) are considered as input parameters and expected duty cycle is considered as an output parameter for fuzzy logic system. FLS generates an optimal output value (expected duty cycle) by integrating the input variables and fuzzy rules set [17, 16, 91].

In a tree-based sensor network, load balancing is one of the techniques to improve the lifetime of the network. A randomized switching algorithm has been proposed in [19] to balance the load (in terms of data forwarding) in a sensor tree, so that all nodes have uniform load. In a network, load balancing is achieved by switching operations (switch the children from its parent node to other potential parent nodes) [19]. However, rechargeable

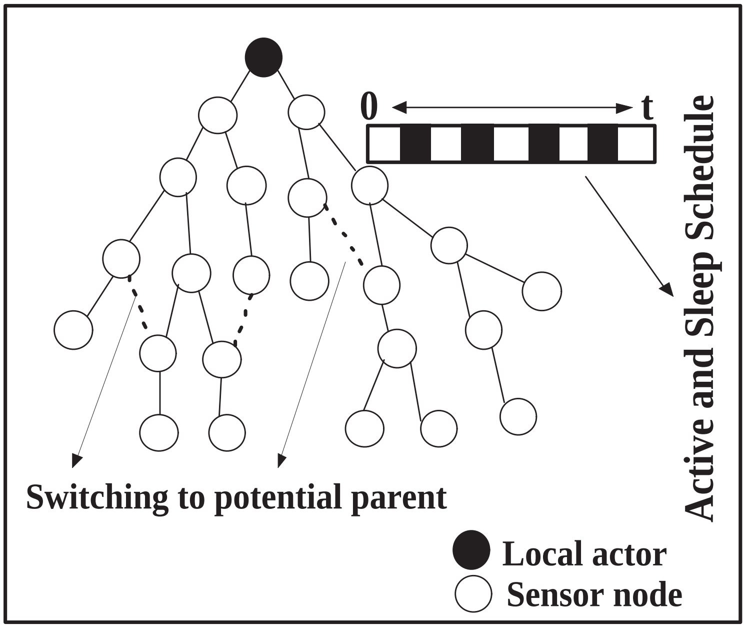


Figure 6.1: The illustration of duty cycle based switching process in a tree-based sensor and actor network

sensor nodes are not considered in [19]. In a rechargeable tree-based sensor network, every node has different energy levels due to the unequal amount of harvested energy and energy consumption. Therefore, load balancing technique may not achieve sustainability. Duty cycle is a suitable mechanism for a rechargeable sensor network to achieve the sustainability (by conserving the energy). In our work, duty cycle is estimated using a fuzzy logic system. Based on the expected duty cycle value, a node forwards the *active and sleep schedule* to the children nodes or switches its children to potential parent nodes for reliable data gathering.

6.2 Adaptive Duty Cycling Mechanism using Fuzzy Logic

In this section, the proposed network model is introduced. An energy consumption model and a harvesting model are presented to estimate the residual energy of a node for future time slots. Further, a fuzzy based adaptive duty cycling mechanism has been proposed for energy harvesting sensor networks.

6.2.1 Network Model

A tree-based wireless sensor and actor network is considered (shown in Fig. 6.1), where the root node is a local actor which collects sensing data from a sensor tree. Let, $T_i = (V, E)$ be a sensor tree, where $V = \{v_0, v_1, ..., v_n\}$ represents a set of n sensor nodes and v_0 is a local actor (i.e. Fig. 6.1 shows the sub-tree from a large scale network tree) and E represents a set of communication links between the sensors. Here, coordinated duty cycle enabled sensor nodes are considered and a node harvests energy from the sunlight [128]. Further, a node switches to a potential parent (a node which has more duty cycle). A *data collection round* is defined as the duration that an actor collects the data from a sensor tree [19]. The notations used in this chapter are shown in Table 6.1.

6.2.2 Energy Consumption and Harvested Model

In this work, energy consumptions are estimated by adopting energy parameters as proposed in [33, 79]. The parameters of energy consumptions for sensing, receiving and transmitting data (over distance $\hat{d}$) are E_s, E_r and E_t, respectively and are given by E_s=α_3, E_r=α_{12}, E_t=$\alpha_{11} + \alpha_2\hat{d}^{\hat{n}}$, where $\hat{n}$: path loss exponent, α_{11}: energy consumption per bit by the transmitter electronics, α_2: energy dissipation in the transmit op-amp, α_{12}: energy consumption per bit by the receiver electronics and α_3: energy consumption for sensing a bit [79]. Further, an event centric application has been considered in our work. Energy estimation is done by considering occurrence of events. A sensor node generates l bits of sensing data per event. β is an average rate of events that occur per unit time [79]. Therefore, the energy consumption for sensing in time slot t is given by $\alpha_3 lt\beta$. The energy consumption (Ec_i) of a node (v_i) is the sum of the consumptions for sensing, receiving data from its children (C_i is the total number of children of a node v_i) and transmitting the total data to its parent node and is given by

$$Ec_i = \alpha_3 lt\beta + \alpha_{12}C_i lt\beta + (C_i lt\beta + lt\beta)(\alpha_{11} + \alpha_2\hat{d}^{\hat{n}}) \tag{6.1}$$

In our work, we have adopted an energy prediction model called *Weather-Conditioned Moving Average Model* [128] to predict the harvested energy (using solar sensors) for a future time slot. In our work, a day is divided into r time slots $(t_0, t_1, ..., t_r)$. According to

Table 6.1: Summary of Notations

Symbol	Description
n	Number of sensor nodes in the network
T	Sensor tree
$E_i(t)$	Residual energy of a node i at time t
$Ex_i(t)$	Expected residual energy of a node i at time t
$H_i(t)$	Harvesting energy of a node i at time t
$Ec_i(t)$	Energy consumption of a node i at time t
C_i	Number of children of node v_i
B_r	Radius of bottleneck zone
k	Maximum number of hops in bottleneck zone
h	Height of sensor tree
$\bar{n}$	Maximum number of children in sensor tree

the model [128], the predicted harvesting energy is computed using the harvesting energy value at previous time slot of the same day and the mean value of the past days of same time slot and is given by

$$H(d, t+1) = \alpha.H(d, t) + GAP_k.(1-\alpha).M_D(d, t+1) \tag{6.2}$$

where $H(d, t+1)$ represents the predicted harvesting energy for a day d at time slot t, α is a weighting factor, $M_D(d, t+1)$ is the mean of the harvesting energies at $(t+1)$ time slots for the previous D days and GAP_k measures the solar conditions in the present day relative to the previous days [128].

The energy consumption (*Equation* 6.1), residual energy (E_i) and the harvesting energy (*Equation* 6.2) are used in the estimation of expected residual energy (Ex_i) of a node for the future time slot (refer *line* 4, *Algorithm* 6.1).

6.2.3 Energy Consumption Model

In this section, energy consumption of bottleneck zone in a sensor tree is estimated by considering duty cycle enabled nodes. In a sensor tree (shown in Fig. 6.2), nodes around the root node (called bottleneck zone) consume more energy than other nodes in the tree [5]. B_r is the radius of the bottleneck zone. The symbol k is the maximum number of hops in the bottleneck zone and h is the height of the sensor tree. Height of a sensor tree is the

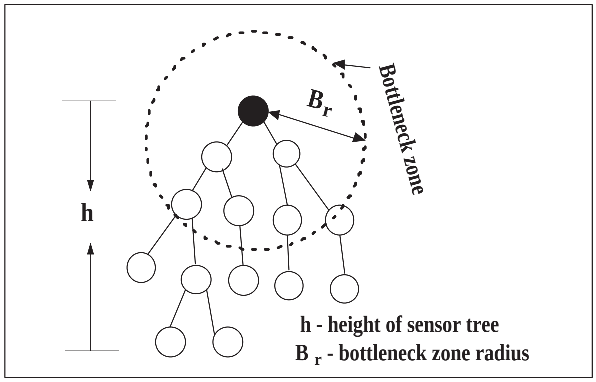


Figure 6.2: Bottleneck zone in a duty cycle enabled sensor tree

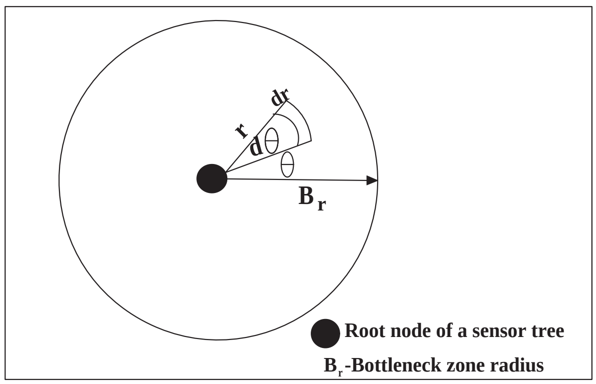


Figure 6.3: A bottleneck zone of radius B_r at root node of sensor tree

longest path (number of hops) from the root node to a leaf node in the tree. Each sensor node in the tree has maximum of $\bar{n}$ children.

The total energy consumption in the bottleneck zone is the summation of consumptions due to (i) sensing operation of the nodes in the bottleneck zone (E_{sb}), (ii) relaying its own sensed data (generated by bottleneck zone nodes) to the root node (E_{rb}), and (ii) relay the data (which is generated by the outside of the bottleneck zone nodes) to the root node (E_{rob}). Therefore, the total energy consumption of the bottleneck zone (E_{bc}) is given by

$$E_{bc} = E_{sb} + E_{rb} + E_{rob} \tag{6.3}$$

The energy consumption of bottleneck zone nodes for sensing operation (E_{sb}) is $p\bar{n}^{k+1}\alpha_3 lt\beta$, where $p\bar{n}^{k+1}$ is the average number of bottleneck zone nodes which are in the active state, probability p denotes the average proportion of time (during the time slot t) in which a node will be in the active state. The energy consumption (E_{rb}) for relying sensed data (generated by bottleneck zone nodes) is given by

$$E_{rb} = \int_0^{B_r} \int_0^{2\pi} k(x) p l t \beta \rho_t r d\theta dr \tag{6.4}$$

where $k(x)$ is the energy consumption for transmitting a bit by a bottleneck zone node which is x distance away from the root (actor) node [7, 8] and is given by

$$k(x) \geq \alpha_1 \frac{\hat{n}}{\hat{n}-1} \frac{x}{d_m} - \alpha_{12}$$

where, d_m is the length of one hop [7] and $d_m = \sqrt[\hat{n}]{\frac{\alpha_1}{\alpha_2(\hat{n}-1)}}$, $\alpha_1 = \alpha_{11} + \alpha_{12}$. Density of a sensor tree is $\rho_t = \frac{\bar{n}^{h+1}}{A}$, where, A is the area of a sensor tree (T_i). The term ($p\rho_t r d\theta dr$) is the number of active nodes in the differential area (as shown in Fig. 6.3) [129].

The energy consumption (E_{rob}) for relaying sensed data (generated at out side of the bottleneck zone) is given by

$$E_{rob} = \sum_{i=1}^{p(\bar{n}^{h+1} - \bar{n}^{k+1}) l t \beta} E_i \tag{6.5}$$

where, E_i is the energy consumption of a sensor node (which is inside the bottleneck zone) to relay i^{th} bit from outside of the bottleneck zone to the root (actor) node and is given by

$$E_i \geq \alpha_1 \frac{\hat{n}}{\hat{n}-1} \frac{B_r}{d_m} \tag{6.6}$$

Therefore, the total energy consumption in the bottleneck zone with duty cycle (i.e. p) is given by

$$E_{bc} = p\bar{n}^{k+1} \alpha_3 l t \beta + \int_0^{B_r} \int_0^{2\pi} k(x) p l t \beta \rho_t r d\theta dr + \sum_{i=1}^{p(\bar{n}^{h+1} - \bar{n}^{k+1}) l t \beta} E_i \tag{6.7}$$

Duty cycle and its impact on sustainability

Duty cycle mechanism is one of the energy conservation techniques in wireless sensor networks. Most of the existing works have not addressed the estimation of duty cycle value in the literature. In a harvesting sensor network, a node does not sustain till next recharge time if it does not have sufficient residual energy. The node conserves some amount of energy to sustain till next rechargeable time. Estimating the appropriate amount of conservation energy (when a node sleeps) to achieve the sustainability and reliable data delivery is a challenging problem in energy harvesting sensor networks. In our work, duty cycle (for a future time slot) is estimated by considering the current residual energy, harvesting energy (for a future time slot) and expected residual energy (for a future time slot) to achieve the sustainability in the harvesting sensor network. If the duty cycle value is very low, then the node does not perform its operations for a long period of time (because node spends long period of time in sleep state). This leads to unreliable data delivery. Therefore, children (of those nodes) will be switched to other potential parent nodes to achieve reliable data delivery. Further, *active and sleep schedule* (coordinated duty cycle schedule) improves the reliable data delivery. The proposed fuzzy based adaptive duty cycling mechanism has been presented in the next section.

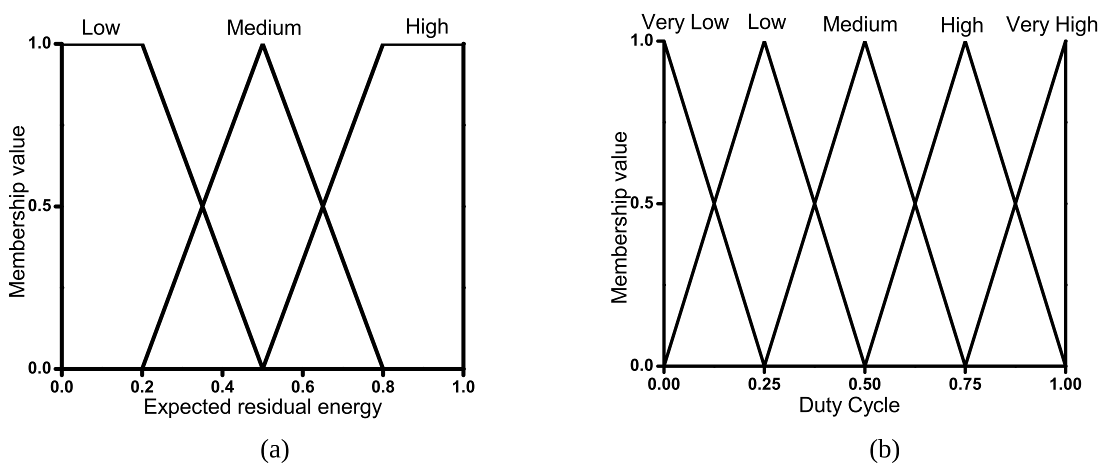


Figure 6.4: Membership functions for (a) input variables (Residual energy, Predicted harvesting energy, Expected residual energy) (b) output variable (Duty cycle)

6.2.4 Fuzzy Based Adaptive Duty Cycling Algorithm

In this section, fuzzy based adaptive duty cycle algorithm (*Algorithm* 6.1) is proposed for an energy harvesting sensor actor network to achieve sustainability. At the end of time slot t, a node estimates the energy consumption $(Ec(t+1))$ for next time slot $(t+1)$ using the *Equation* 6.1 and predicts the value of harvested energy $(H_i(t+1))$ for time slot $(t+1)$ using the *Equation* 6.2 (*lines* 2 to 3). Further, the node determines expected residual energy $(Ex_i(t+1))$ for time slot $(t+1)$ (*line* 4). A Fuzzy Logic System (FLS) determines the expected duty cycle (Ed) adaptively by considering the residual energy (E_i), predicted

Algorithm 6.1 Fuzzy Based Adaptive Duty Cycling Algorithm

Input: Tree-based sensor actor network (T_i) **Output:** Sustainable sensor tree

1: **for** $i = 1$ to n **do**
2:     $Ec_i(t+1) = \alpha_3 l(t+1)\beta + \alpha_{12} C_i l(t+1)\beta + (C_i l(t+1)\beta + l(t+1)\beta)(\alpha_{11} + \alpha_2 \hat{d}^{\hat{n}})$
3:     Predict the value of $H_i(t+1)$ using the *Equation* 6.2.
4:     $Ex_i(t+1)$=$E_i(t) + H_i(t+1) - Ec_i(t+1)$
5:     $Ed[i]$ =Fuzzy_Duty_Cycle($E_i(t), H_i(t+1), Ex_i(t+1)$)
6:     **if** $Ed[i] < T_h$ **then**
7:         Switch_Children(i)
8:     **else**
9:         Forward the *active and sleep schedule* to the children
10:     **end if**
11: **end for**

Algorithm 6.2 Fuzzy_Duty_Cycle($E_i(t), H_i(t+1), Ex_i(t+1)$)

Output: Expected duty cycle Ed_i

1: Empty the list $l < value, membershiplevel >$
2: Find membership values ($\mu(E_i(t)), \mu(H_i(t+1))$ and $\mu(Ex_i(t+1))$ and linguistic levels using *Triangular* membership function.
3: DR ={A rule set with all possible combinations of determined linguistic levels (from $line - 2$)}
4: **for** each rule in DR **do**
5:     **if** $\mu(E_i(t)), \mu(H_i(t+1)), \mu(Ex_i(t+1))$ **then** fit the membership levels of this rule
6:         Add an entry to the list l with
7:         $value$=Max($\mu(E_i(t)), \mu(H_i(t+1)), \mu(Ex_i(t+1))$)
8:         $membershiplevel$ = output membership level of this rule
9:     **end if**
10: **end for**
11: Ed_i =Defuzzify(l)
12: return Ed_i

Algorithm 6.3 Switch_Children(i)

1: Enqueue(children(i))
2: **while** IsEmptyQueue==false **do**
3:     j =Dequeue()
4:     P_p =Find_Potential_Parents(j)
5:     **if** $P_p == \emptyset$ **then**
6:         Enqueue(children(j))
7:     **end if**
8:     $\forall \bar{p} \in P_p$, Select a potential parent node $\bar{p}$ with maximum duty cycle value $(Ed[\bar{p}])$.
9:     $\bar{p}$ is the new parent of node j
10:     Update the tree
11: **end while**

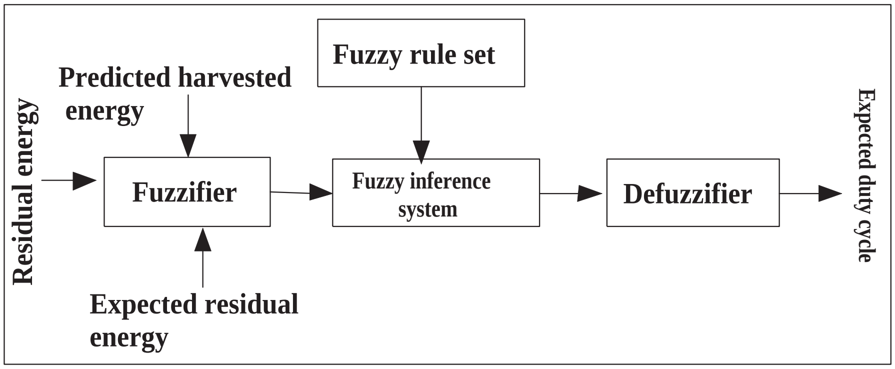


Figure 6.5: Fuzzy Logic System

harvested energy (H_i) and expected residual energy (Ex_i) (*line* 5) as input parameters. The explanation of the FLS process (*Algorithm* 6.2) is discussed in the next section. Further, a node takes the switching decision based on the determined expected duty cycle value to sustain till the next rechargeable time. If the expected duty cycle value of a node is less than a threshold value (T_h) (which varies from application to application), then children of the node will be switched to potential parent (refer *Algorithm* 6.3). Otherwise, the node forwards its *active and sleep schedule* to the children so that the children send their data at active time periods (for reliable data delivery) (*lines* 6 to 10).

In *Algorithm* 6.3, switching process is presented. Children of a node (which has the less expected duty cycle value than the threshold value) will be inserted into a queue (called *Enqueue*) (*line* 1). Further, the potential parents will be found for every removal node from the queue (called *Dequeue*) (*lines* 3 to 4). Potential parents of a node is a set of nodes which are in the communication range of that node (other than the current parent). A node with highest expected duty cycle value will be chosen from the potential parents as a new parent. If a child node does not have a potential parent, then its subsequent children will be inserted into the queue (*lines* 5 to 10).

6.2.5 Fuzzy Logic System for Determining Duty Cycle

In this section, a fuzzy logic system (FLS) is presented and the duty cycle is estimated for a future time slot. The process of fuzzy logic system has been presented in *Algorithm* 6.2. Fig. 6.5 shows the fuzzy logic system with three input parameters, such as current residual

Table 6.2: Fuzzy decision making rules

Inputs			**Output**
Residual energy	**Predicted harvesting energy**	**Expected residual energy**	**Duty cycle**
Low	Low	Low	Very Low
Low	Low	Medium	Low
Low	Low	High	Medium
Low	Medium	Low	Low
Low	Medium	Medium	Medium
Low	Medium	High	High
Low	High	Low	Low
Low	High	Medium	Medium
Low	High	High	Very High
Medium	Low	Low	Very Low
Medium	Low	Medium	Low
Medium	Low	High	Medium
Medium	Medium	Low	Low
Medium	Medium	Medium	Medium
Medium	Medium	High	High
Medium	High	Low	Low
Medium	High	Medium	Medium
Medium	High	High	High
High	Low	Low	Very Low
High	Low	Medium	Low
High	Low	High	Medium
High	Medium	Low	Low
High	Medium	Medium	Medium
High	Medium	High	Medium
High	High	Low	Low
High	High	Medium	Medium
High	High	High	Very High

Table 6.3: Fuzzy input/output variables and their linguistic variables

Input/Output variable	**Linguistic values**
Residual energy	Low, Medium, High
Predicted harvesting energy	Low, Medium, High
Expected residual energy	Low, Medium, High
Duty cycle	Very Low, Low, Medium, High, Very High

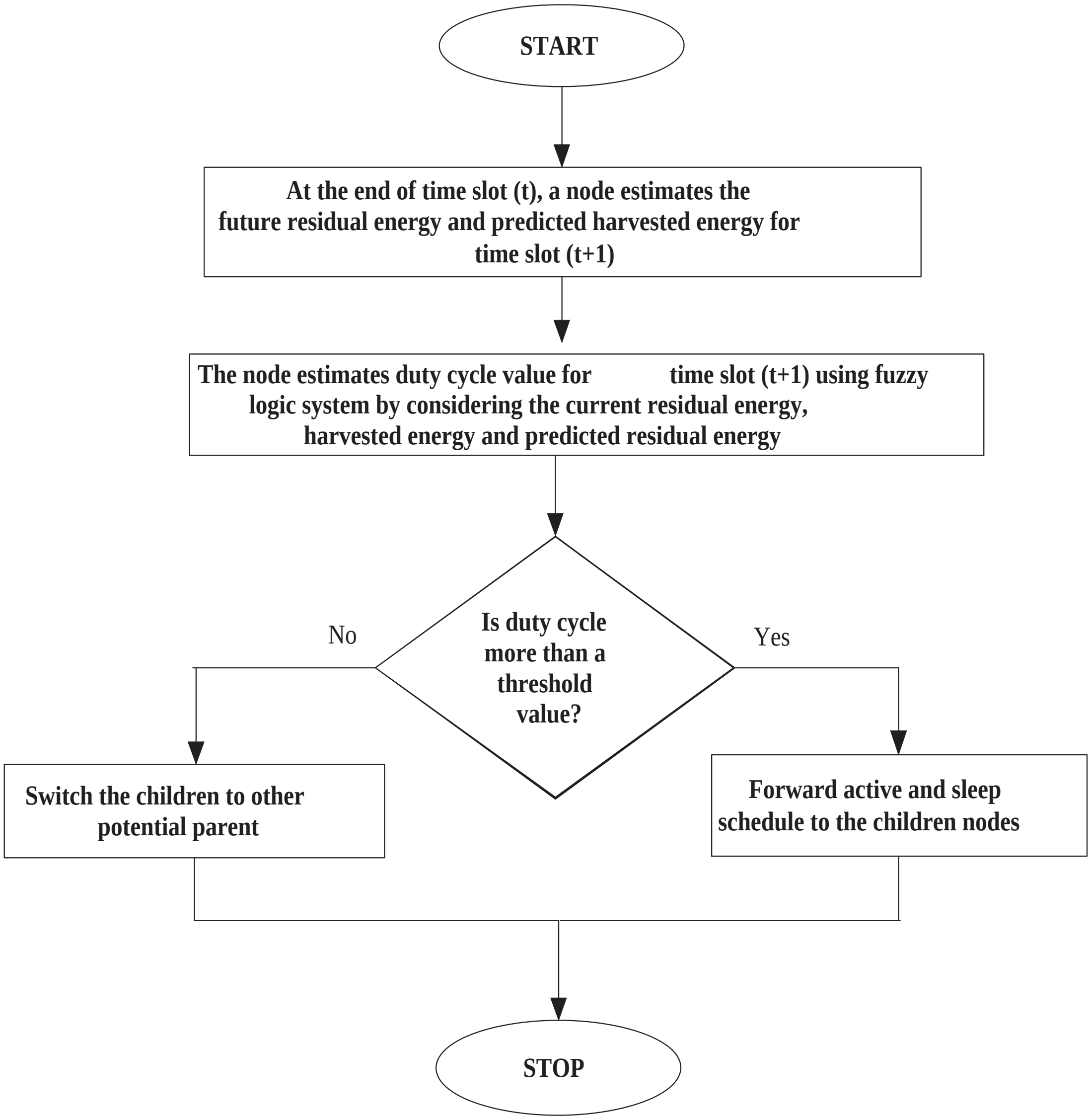


Figure 6.6: Flowchart of the proposed mechanism

energy, predicted harvesting energy (for a future time slot) and predicted residual energy (for a future time slot) and (one output parameter) expected duty cycle. FLS process has four steps: (i) Fuzzifier, (ii) Fuzzy rule set, (iii) Fuzzy inference system and (iv) Defuzzifier. The fuzzifier converts the crisp values of input parameters to the membership values and linguistic levels using the triangular membership functions (*line* 2, *Algorithm* 6.2). The membership functions of the input and output variables are shown in Fig. 6.4(a) and Fig. 6.4(b), respectively. The linguistic variables of input variables are considered as *low, medium* and *high* and the linguistic variables of output variable (expected duty cycle) are considered as *very low*, *low*, *medium*, *high* and *very high* (as shown in Table 6.3).

Fuzzy rule set is a collection of fuzzy decision making rules which are generated from three input variables as shown in Table 6.2. Fuzzy rules are generated based on the following principle. A node with high current residual energy, high predicted harvesting energy (for a future time slot), high expected residual energy (for a future time slot) acquires highest expected duty cycle value. Fuzzy inference system finds the fuzzy output by using the fuzzy rule set and membership functions. In *Algorithm* 6.2, *line* 3 to *line* 9 explains the fuzzy inference process. Further, the defuzzifier converts the fuzzy output to a crisp value (which indicates the expected duty cycle of a node) (*line* 11). In our work, *center of area* (COA) method (is also called center of gravity in the literature [83, 84]) is used in the defuzzification process [17].

The complete working operation of the proposed fuzzy based adaptive duty cycling algorithm is given in the flowchart as shown in Fig. 6.6. At the end of time slot t, a node estimates the future residual energy and predicted harvested energy for time slot $(t + 1)$. Further, the node determines the expected duty cycle (for time slot $(t + 1)$) using fuzzy logic system. The parameters current residual energy, harvesting energy (for a future time slot) and expected residual energy (for a future time slot) are considered as fuzzy inputs and duty cycle is the fuzzy output. The node switches its children to another potential parent node if duty cycle value is less than the threshold value, otherwise, the node forwards the *active and sleep schedule* to its children for reliable data collection.

6.2.6 Analysis of Fuzzy Based Adaptive Duty Cycling Algorithm

In this section, the proposed mechanism has been analyzed in terms of time complexity and overhead. In *Algorithm 6.3*, line 1 takes $O(\hat{C}_{max})$ time where, $\hat{C}_{max}$ is the maximum number of children of a node in the tree. The loop on line 2 runs for $O(\hat{C}_{max})$ time. Line 3 takes $O(1)$ time. Find_Potential_Parents() function on line 4 takes $O(\hat{P}_{p\_max})$ time where, $\hat{P}_{p\_max}$ is maximum number of neighbors of a node in the tree. Line 5 runs in $O(1)$ time. Line 6 takes $O(\hat{C}_{max})$ time. Lines 8 and 9 run in $O(1)$ time. Line 10 takes $O(n)$ time to update the tree (n is the total number of nodes in the tree). Since $\hat{C}_{max} \leq \hat{P}_{p\_max} < n$, the complexity of Switch_Children() function (*Algorithm 6.3*) is obtained as $O(n\hat{C}_{max})$. In *Algorithm 6.1*, the loop on line 1 runs for all nodes in the network. Line 2 takes $O(\hat{C}_{max})$ time. Lines 3 and 4 runs in $O(1)$ time. In line 5, Fuzzy_Duty_Cycle() function (*Algorithm 6.2*) runs in $O(1)$ time. The Switch function (*Algorithm 6.3*) on line 7 takes $O(n\hat{C}_{max})$ time. Therefore, the total time complexity of the proposed mechanism is obtained as $O(n^2\hat{C}_{max})$.

Every node in the tree forwards *active and sleep schedule* only to its children in each time slot (every round) (line 9 in *Algorithm 6.1*). A data packet traverses multiple hops to reach the actor node whereas duty cycle scheduling packets are forwarded only to the children nodes (i.e. with one hop). Therefore, in comparison to energy consumption for data packet transmission, the energy consumptions for control packet transmissions are minimal for large-scale sensor networks. Assume that, in a network, maximum 1000 number of control packets are generated for 100 number of nodes in 100 data collection rounds. The size of the control packet is considered as 10 bytes [130] which is significantly smaller than the size of a data packet (refer Table 6.4). The energy consumption for transmitting the control packets is $1000 * 10 * 8 * E_t = 0.080\ Jules$ (refer Table 6.4). Moreover, the maximum number of n control packets (duty cycle scheduling packets) are transmitted in the tree for each time slot where n is the number of nodes in the tree.

6.3 Performance Evaluation

In this section, the performance of the proposed mechanism is compared with a randomized switching algorithm [19] by considering network sustainability metrics, such as number of rounds network is connected, first node disconnected round, number of packets received at actor node and maximum number of dead nodes.

6.3.1 Simulation Environment

The proposed fuzzy based adaptive duty cycle algorithm is simulated using the *Network Simulator-3* [9]. A sensor tree is constructed using the breadth-first-search algorithm. In a data collection round, a node relays the data to a parent node. Assume that, every node generates 1 unit of data packet (i.e. 960 bits each) in a round (each time slot) [19]. The ini-

Table 6.4: Simulation Parameters

Parameter	**Value**
Number of sensor nodes	100 to 500 nodes
Transmission range	30 $meters$
Initial energy of a node	$2500KJ$
$E_s = \alpha_3$	$\alpha_3 = 50 \times 10^{-9}\ Joules/bit$
$E_r = \alpha_{12}$	$\alpha_{12} = 0.787 \times 10^{-6} Joules/bit$
$E_t = \alpha_{11} + \alpha_2 \hat{d}^{\hat{n}}$	$\alpha_{11} = 0.937 \times 10^{-6} Joules/bit,$ $\alpha_2 = 10 \times 10^{-12} Joules/bit/meter^2$ $\hat{d} = 85\ meters$
Path lose exponent ($\hat{n}$)	2
Data generated per event by a node (l)	$960bits$
Average rate of events occur per unit time (β)	100

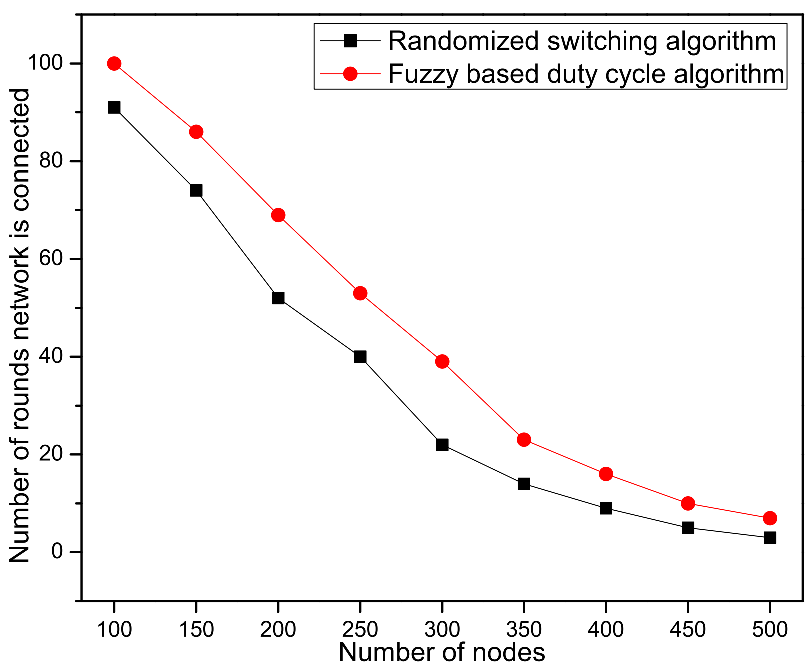


Figure 6.7: Total number of rounds for which network is connected

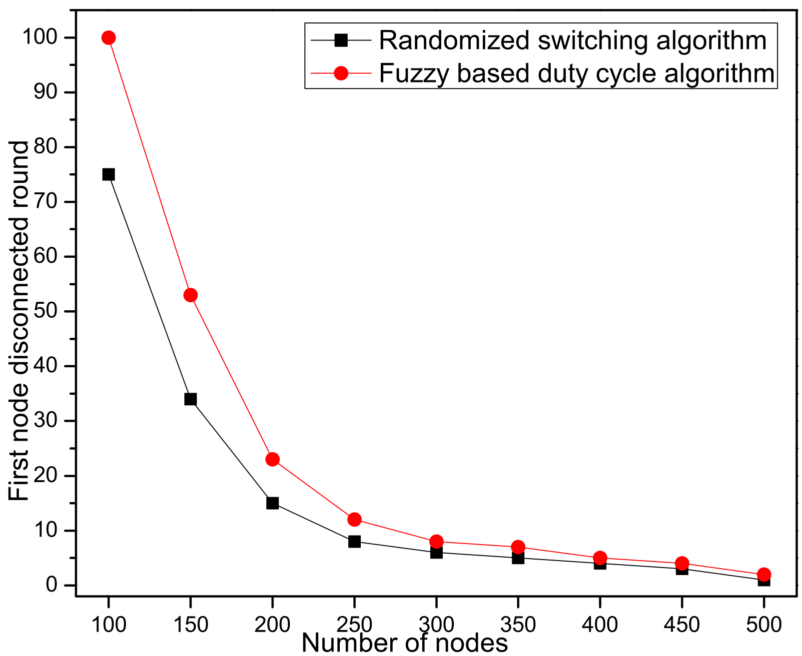


Figure 6.8: First node disconnected round

tial energy of a node is $2500 Joules$ and a node harvests maximum of $200 Joules$ per round. The following simulation parameters are considered: $\alpha_3 = 50 \times 10^{-9} Joules/bit$, $\alpha_{12} = 0.787 \times 10^{-6} Joules/bit$, $\alpha_{11} = 0.937 \times 10^{-6} Joules/bit$, $\alpha_2 = 10 \times 10^{-12} Joules/bit/meter^2$, $d = 85\ meters$ and $\hat{n}$=2 [79, 8]. The number of sensor nodes are varied from 100 to 500 and the simulation results are presented for 100 rounds. Table 6.4 shows the parameters which are considered in the simulation.

6.3.2 Simulation Results and Analysis

Fig. 6.7 shows the comparison of the number of rounds (a parameter for time duration) in which all nodes are connected between the proposed fuzzy based adaptive duty cycle mechanism and the randomized switching algorithm [19]. It can be observed from Fig. 6.7 that, using the proposed approach the network remains connected for (on an average) 60.44% of time duration (in rounds) in comparison to the randomized switching algorithm.

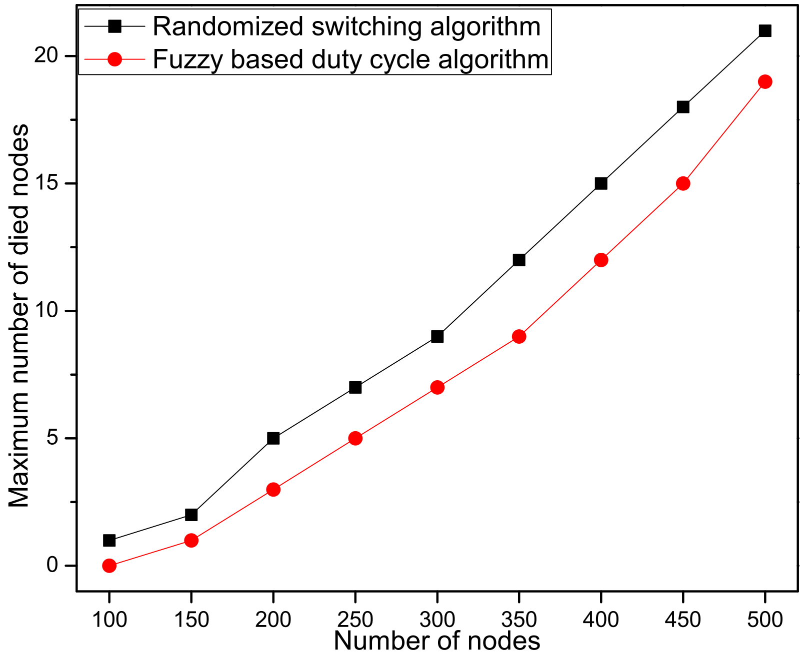


Figure 6.9: Maximum number of dead nodes

Table 6.5: Total (average) number of control packets generated per data collection round

Number of nodes	Number of control packets
100	99
150	148
200	195
250	239
300	278
350	311
400	339
450	359
500	374

The proposed mechanism estimates the expected duty cycle value of a node for future time slots. According to the duty cycle value, the node will perform the switching operation or forwards the *active and sleep schedule* to the children for sustainability (i.e. till next rechargeable time slot). On the other hand, the randomized switching algorithm performs

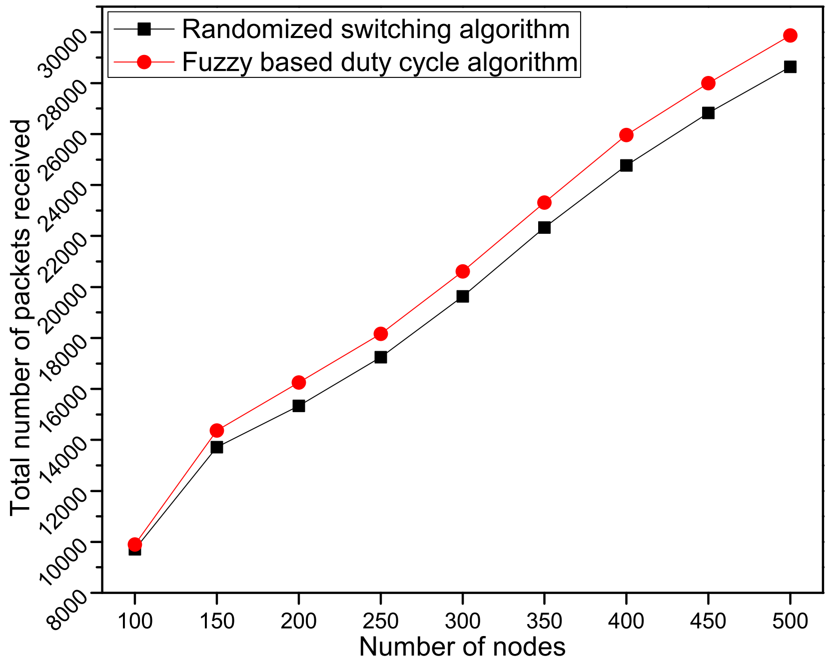


Figure 6.10: Total number of packets received for 100 data collection rounds

the switching operation to balance the current load factor. The existing approach does not consider the sustainability condition. It can be seen from Fig. 6.7 that, as number of nodes increases, the network connectivity rounds decreases because the nodes near to the actor are overloaded with more incoming data traffic. Therefore, the nodes near to the root node deplete their energy quickly.

Fig. 6.8 shows that the first node die round (first round in which first node dies) and it occurs first with the randomized algorithm [19] in comparison to the proposed mechanism. The proposed mechanism takes switching decision based on the expected residual energy. This gives the survivability to a node till next rechargeable time. It can be seen from Fig. 6.8 that, a node survives on an average of 47.13% rounds with the proposed mechanism. Further, Fig. 6.9 shows the maximum number of dead nodes for 100 rounds. The proposed mechanism reduces the maximum number of dead nodes significantly as compared with the randomized switching algorithm [19].

Fig. 6.10 and Fig. 6.11 show the average number of packets received at the root (local

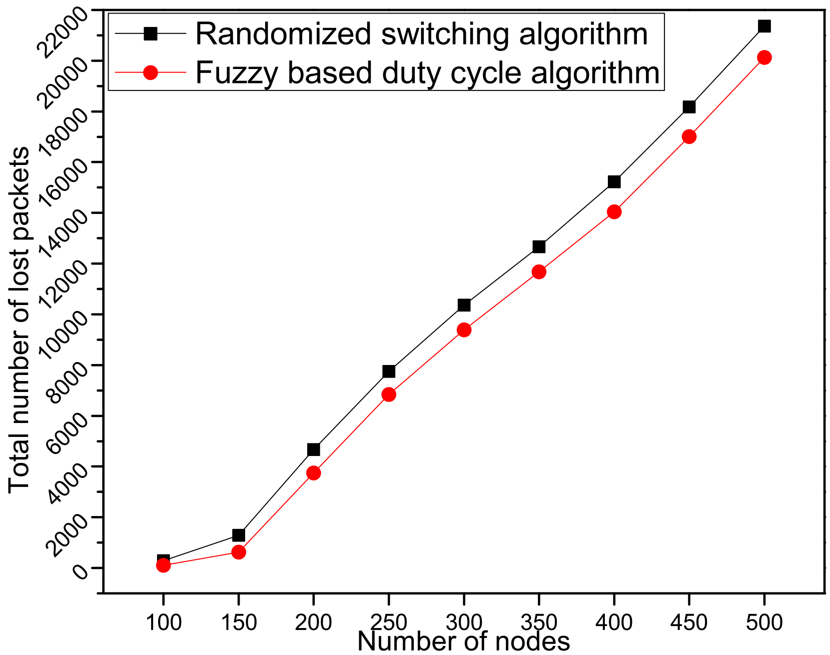


Figure 6.11: Total number of lost packets for 100 data collection rounds

actor) node and number of lost packets in a sensor tree, respectively. In a duty cycle based network, packets may be dropped if the receiver node is in inactive state. In the proposed mechanism, a node forwards the *active and sleep schedule* to the children so that children relay the data at active time periods. This results in reduction of dropping packets (refer Fig. 6.11) with the proposed mechanism as compared to the randomized algorithm [19]. Table. 6.5 shows that the total (average) number of control packets (*active and sleep schedule* packets) generated in a sensor tree per data collection round. It can be seen from Table. 6.5 that as number of nodes increases the traffic generated by the control packets increases. It has been observed from Fig. 6.7, Fig. 6.8 and Fig. 6.9 that the sustainability of the network is improved and this provides better network stability. Further, successful reception of packets is also improved as shown in Fig. 6.10.

6.4 Summary

In this chapter, a fuzzy based adaptive duty cycle algorithm has been proposed to achieve sustainability for an energy harvesting sensor network. Duty cycle of a sensor node is adaptively chosen by the fuzzy logic controller. Fuzzy logic system determines the expected duty cycle value by considering the residual energy, predicted harvesting energy and expected residual energy. Further, the proposed mechanism takes a switching decision for survivability of the network. Coordinated duty cycle schedule has been considered to improve the reliable data gathering. From the simulation results, it has been observed that there is an improvement in network connected rounds on an average of 60.44% with the proposed mechanism. Survivability of a node is increased on an average of 47.13% with the proposed mechanism. Further, there is a reduction in the maximum number of dead nodes and a significant improvement in the total number of received packets at actor node with the proposed mechanism in comparison to a randomized switching algorithm.

Chapter 7

Conclusion and Future Scope

This thesis investigates the design and development of data gathering protocols which enhance the lifetime and reliable data transmission while providing stringent delivery delay in wireless sensor and actor networks. Different data gathering protocols which utilize less energy and delay when gathering data in the sensor actor network are presented. The proposed mechanisms achieve better performance in terms of the end-to-end delay, energy consumptions, number of data transmissions and lifetime. The proposed data collection protocols are implemented and verified using the *network simulator-3*. A comparative study of the proposed protocols has been presented and discussed through a number of experiments in order to demonstrate their merits and capabilities.

In this thesis, the main challenges of WSANs, such as energy efficiency, reliable data delivery and delivery delay have been addressed. For various applications of WSANs, it is important to improve reliable data collection along with energy efficiency. Reliable data transmission depends on packet loss. Retransmission of dropped packet does not satisfy the strict delivery delay requirement and also it increases energy consumption. Therefore, reduction of packet dropping rate improves energy efficiency and reduces delay for reliable data collection. In this thesis, the main parameters, such as buffer overflow, unreliable link quality and low energy levels (temporary disconnection of nodes due to low energy levels) are identified that cause the dropping of packets. This thesis has made contributions by considering the above factors in making data gathering algorithms in WSANs energy efficient and reliable. Contributions of this thesis are to investigate the reduction of packet loss (by considering different aspects like buffer occupancy, link loss rate and energy levels of nodes) in the data gathering process.

7.1 The Major Contributions of the Thesis

An adaptive buffer management mechanism using markov decision process has been developed for tree-based sensor and actor networks. This is presented in Chapter 3. The proposed approach reduces the number of retransmissions, delivery delay and improves the energy efficiency. The state of a node is represented as buffer levels which may change with dynamic traffic. The buffer states (in which packets will be dropped) are determined to minimize the average delivery delay and energy consumption. Also, the bounds on visiting times and contact times are estimated for a mobile actor to mitigate buffer overflow problem. The energy consumptions have been estimated with the proposed network model.

In Chapter 4, a reliable data transmission mechanism using opportunistic encoding has been proposed for a WSAN with faulty nodes. A link may be unreliable in the presence of a faulty region that results in loss of packets. A network coding approach has been designed by considering link loss rates and appropriate level of redundancy to achieve reliable data delivery. Further, a markov decision process is developed for opportunistic network coding decisions. The proposed mechanism determines the level of packet redundancy adaptively in the encoding process to improve reliable data gathering and to reduce the number of data transmissions.

A fuzzy based delay and energy-aware intelligent routing mechanism has been proposed in Chapter 5 to select efficient routes. In a heterogeneous sensor actor network, packet loss may occur due to bad link quality, overflow of buffer and low energy levels. Retransmission of the lost packets leads to increase in energy consumption and delay. In the proposed mechanism, routing decisions are taken using a fuzzy logic system by considering network resources, such as residual energy, quality of link, available buffer size and distance (proximity).

In Chapter 6, a fuzzy based adaptive duty cycling algorithm has been designed to achieve the network sustainability in harvesting sensor actor networks. In the proposed approach, current residual energy, predicted harvesting energy (for a futuristic time slot) and predicted residual energy parameters are considered as fuzzy input variables to estimate duty cycle for a sensor node. A node takes a switching decision based on predicted duty cycle value for survivability. Also, coordinated duty cycling is considered in Chapter 6 to improve reliable data collection. The energy consumptions in bottleneck zone have been estimated by considering duty cycle in WSANs.

7.2 Future Scope

Although the proposed data gathering algorithms show promising performance improvements as compared to existing relevant mechanisms available in the literature, there are other aspects and scenarios which could be considered. Some of the potential extensions of our research work presented in this thesis are listed as follows:

- In Chapter 3, we have formulated a model for an actor to estimate bound on buffer occupancy and energy consumption with one mobile actor. However, in a *multi-actor* scenario, the actor-actor coordination during path scheduling and packet gathering can be explored as future scope of research. Further, research may be extended to investigate the performance of the proposed MDP approach in a duty-cycled flat network.

- In Chapter 4, a network coding based reliable data gathering mechanism has been presented. This approach can be further explored in the presence of node mobility and rechargeable sensor nodes.

- As a future research scope, the problem addressed in Chapter 5 can be further investigated with duty cycle based sensor network. Further research may be extended by exploring the proposed delay and energy aware routing protocol with various degree of node mobility and their movement patterns.

- As future research challenge, energy harvesting sensor actor networks can be further investigated for different node mobility and network topologies. .

Bibliography

[1] Saeed Sedighian Kashi and Mohsen Sharifi. Connectivity weakness impacts on coordination in wireless sensor and actor networks. *IEEE Communications Surveys and Tutorials*, 15(1):145–166, 2013.

[2] I.F.Akyildiz and Ismail H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351–367, 2004.

[3] I.F.Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(2):393–422, 2002.

[4] Songtao Guo, Cong Wang, and Yuanyuan Yang. Joint Mobile Data Gathering and Energy Provisioning in Wireless Rechargeable Sensor Networks. *IEEE Transactions on Mobile Computing*, 13(12):2836–2852, December 2014.

[5] Qinghua Wang and Tingting Zhang. Bottleneck Zone Analysis in Energy-Constrained Wireless Sensor Networks. *IEEE Communications Letters*, 13(6):423–425, 2009.

[6] Xuan Rao, Hongyu Huang, Jiqiang Tang, and Huan Zhao. Residual energy aware mobile data gathering in wireless sensor networks. *Telecommunication Systems*, 62(1):31–41, March 2016.

[7] Manish Bhardwaj, Timothy Garnett, and Anantha P. Chandrakasan. Upper Bounds on the Lifetime of Wireless Sensor Networks. *IEEE International Conference on Communications (ICC)*, pages 785–790, 2001.

[8] Rashmi Ranjan Rout and Soumya K. Ghosh. Adaptive data aggregation and energy efficiency using network coding in a clustered wireless sensor network: An analytical approach. *Computer Communications*, 40:65–75, March 2014.

[9] Network Simulator 3.

[10] Quan Jun Chen, Salil S. Kanhere, Mahbub Hassan, and Kun-Chan Lan. Adaptive Position Update in Geographic Routing. *IEEE International Conference on Communications (ICC)*, pages 4046–4051, June 2006.

[11] Sudip Misra, Pushpendu Kar, Arjit Roy, and Mohammad S. Obaidat. Existence of dumb nodes in stationary wireless sensor networks. *The Journal of Systems and Software*, 91:135–146, May 2014.

[12] Celimuge Wu, Yusheng Ji, Juan Xu, Satoshi Ohzahata, and Toshihiko Kato. Coded packets over lossy links: A redundancy-based mechanism for reliable and fast data collection in sensor networks. *Computer Networks*, 70:179–191, September 2014.

[13] Muhammad Adeel Mahmood, Winston K.G. Seah, and Ian Welch. Reliability in wireless sensor networks: A survey and challenges ahead. *Computer Networks*, 79:166–187, March 2015.

[14] M. Sammer Srouji, Zhonglei Wang, and Jorg Henkel. RDTS: A Reliable Erasure-Coding Based Data Transfer Scheme for Wireless Sensor Networks. *2011 IEEE 17th International Conference on Parallel and Distributed Systems*, January 2011.

[15] Xiaohan Lai, Xiaoyu Ji, Xinyan Zhou, and Longdao Chen. Energy Efficient Link-delay Aware Routing in Wireless Sensor Networks. *IEEE Sensors Journal*, 18(2):837–848, November 2017.

[16] Mario Collotta, Giovanni Pau, and Vincenzo Maniscalco. A Fuzzy Logic Approach by Using Particle Swarm Optimization for Effective Energy Management in IWSNs. *IEEE Transactions on Industrial Electronics*, 64(12):9496–9506, December 2017.

[17] Jin-Shyan Lee and Wei-Liang Cheng. Fuzzy-Logic-Based Clustering Approach for Wireless Sensor Networks Using Energy Predication. *IEEE Sensors Journal*, 12(9):2891–2897, September 2012.

[18] Liansheng Tan and Shengda Tang. Energy Harvesting Wireless Sensor Node With Temporal Death: Novel Models and Analyses. *IEEE/ACM Transactions on Networking*, 25(2):896–909, April 2017.

[19] Sk Kajal Arefin Imon, Adnan Khan, Mario Di Francesco, and Sajal K. Das. Energy-efficient randomized switching for maximizing lifetime in tree-based wireless sensor networks. *IEEE/ACM Transactions on Networking*, 23(5):1401–1415, October 2015.

[20] D. Culler J. Hill. Mica: a wireless platform for deeply embedded networks. *IEEE MICRO*, 22(6):12–24, 2002.

[21] Nikolaos A. Pantazis, Stefanos A. Nikolidakis, and Dimitrios D. Vergados. Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys and Tutorials*, 15(2):551–591, July 2013.

[22] Naseer Sabri, S. A. Aljunid, R. B. Ahmad, M.F.Malik, Abid Yahya, R. Kamaruddin, and M.S.Salim. Towards Smart Wireless Sensor Actor Networks: Design Factors and Applications. *IEEE Symposium on Industrial Electronics and Applications (ISIEA2011)*, pages 704–708, September 2011.

[23] Vehbi Cagri Gungor, Ozgur B. Akan, and Ian F. Akyildiz. A Real-Time and Reliable Transport Protocol for Wireless Sensor and Actor Networks. *IEEE/ACM Transactions on Networking*, 16(2):359–370, April 2008.

[24] Tommaso Melodia, Dario Pompili, Vehbi C. Gungor, and Ian F. Akyildiz. Communication and Coordination in Wireless Sensor and Actor Networks. *IEEE Transactions on Mobile Computing*, 6(10):1116–1129, October 2007.

[25] Tommaso Melodia, Dario Pompili, and Ian F. Akyildiz. Handling Mobility in Wireless Sensor and Actor Networks. *IEEE Transactions on Mobile Computing*, 9(2):160–173, February 2010.

[26] Yung-Liang Lai and Jehn-Ruey Jiang. Optimal Multipath Planning for Intrusion Detection in Smart Homes Using Wireless Sensor and Actor Networks. *39th International Conference on Parallel Processing Workshops*, October 2010.

[27] Jan Sliwa and Emmanuel Benoist. Wireless sensor and actor networks: e-Health, e-Science, e-Decisions. *International Conference on Selected Topics in Mobile and Wireless Networking (iCOST-2011)*, October 2011.

[28] Kemal Akkaya, Fatih Senel, and BrianMcLaughlan. Clustering of wireless sensor and actor networks based on sensor distribution and connectivity. *Journal of Parallel and Distributed Computing*, 69(6):573–587, June 2009.

[29] Akhilesh Shrestha and Liudong Xing. A Performance Comparison of Different Topologies for Wireless Sensor Networks. *IEEE Conference on Technologies for Homeland Security*, June 2007.

[30] Chen Xuhui, Yang Zhiming, and Cheng Huiyan. Unequal Clustering Mechanism of LEACH Protocol for Wireless Sensor Networks. *WRI World Congress on Computer Science and Information Engineering*, July 2009.

[31] Isabel Dietrich and Falko Dressler. On the Lifetime of Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 5(1):5:1– 5:39, February 2009.

[32] Soobin Lee and Hwang S.Lee. Analysis of Network Lifetime in Clustere-Based Sensor Networks. *IEEE Communications Lettters*, 14(10):900–902, October 2010.

[33] Rashmi Ranjan Rout and Soumya K Ghosh. Enhancement of Lifetime using Duty Cycle and Network Coding in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 12(2):656–667, February 2013.

[34] Can Tunca, Sinan Isik, M. Yunus Donmez, and Cem Ersoy. Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys and Tutorials*, 16(2):877–897, 2014.

[35] Cong Wang, Ji Li, Fan Ye, and Yuanyuan Yang. A Mobile Data Gathering Framework for Wireless Rechargeable Sensor Networks with Vehicle Movement Costs and Capacity Constraints. *IEEE Transactions on Computers*, 65(8):2411–2427, August 2016.

[36] Yuanchao Shu, Kang G. Shin, Jiming Chen, and Youxian Sun. Joint Energy Replenishment and Operation Scheduling in Wireless Rechargeable Sensor Networks. *IEEE Transactions on Industrial Informatics*, 13(1):125–134, February 2017.

[37] Miao Zhao, Ji Li, and Yuanyuan Yang. A Framework of Joint Mobile Energy Replenishment and Data Gathering in Wireless Rechargeable Sensor Networks. *IEEE Transactions on Mobile Computing*, 13(12):2689–2705, December 2014.

[38] Ricardo Silva, Jorge Sa Silva, and Fernando Boavida. Mobility in wireless sensor networks – Survey and proposal. *Computer Communications*, 52:1–20, October 2014.

[39] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003*, June 2003.

[40] Lang Tong, Qing Zhao, and S. Adireddy. Sensor networks with mobile agents. In *IEEE Military Communications Conference, 2003. MILCOM 2003.*, pages 688–693, October 2003.

[41] Sushant Jain, Rahul C. Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. *Mobile Networks and Applications*, 11(3):327–339, June 2006.

[42] Shashidhar Rao Gandham, Milind Dawande, Ravi Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. *IEEE Global Telecommunications Conference, GLOBECOM '03.*, December 2004.

[43] Z. Maria Wang, Stefano Basagni, Emanuel Melachrinoudis, and Chiara Petrioli. Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, January 2005.

[44] Yi Shi and Y. Thomas Hou. Theoretical Results on Base Station Movement Problem for Sensor Network. *IEEE INFOCOM 2008*, May 2008.

[45] Jun Luo and Jean-Pierre Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, August 2005.

[46] Ashraf Hossain, S. Chakrabarti, and P.K. Biswas. Equal energy dissipation in wireless image sensor network: A solution to energy-hole problem. *Computers and Electrical Engineering*, 39(6):1789 – 1799, 2013.

[47] Xiaojiang Ren, Weifa Liang, and Wenzheng Xu. Data Collection Maximization in Renewable Sensor Networks via Time-Slot Scheduling. *IEEE Transactions on Computers*, 64(7):1870–1883, July 2015.

[48] Yu Gu, Yusheng Ji, Jie Li, and Baohua Zhao. ESWC: Efficient Scheduling for the Mobile Sink in Wireless Sensor Networks with Delay Constaint. *IEEE Transactions on parallel and Distributed Systems*, 24(7):1310–1320, July 2013.

[49] Tariq Umer, Muhammad Amjad, Muhammad Khalil Afzal, and Muhammad Aslam. Hybrid rapid response routing approach for delay-sensitive data in hospital body area sensor network. *Proceedings of the 7th International Conference on Computing Communication and Networking Technologies*, pages 3:1–3:7, 2016.

[50] Ming Ma, Yuanyuan Yang, and Miao Zhao. Tour Planning for Mobile Data-Gathering Mechanisms in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, 62(4):1472–1483, May 2013.

[51] Miao Zhao and Yuanyuan Yang. Bounded Relay Hop Mobile Data Gathering in Wireless Sensor Networks. *IEEE Transactions on Computers*, 61(2):265–277, February 2012.

[52] Miao Zhao, Ming Ma, and Yuanyuan Yang. Efficient Data Gathering with Mobile Collectors and Space-Division Multiple Access Technique in Wireless Sensor Networks. *IEEE Transactions on Computers*, 60(3):400–417, March 2011.

[53] Ricardo C. Carrano, Diego Passos, Luiz C. S. Magalhaes, and Celio V. N. Albuquerque. Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 16(1):181–194, 2014.

[54] Chih fan Hsin and Mingyan Liu. Randomly Duty-cycled Wireless Sensor Networks: Dynamics of Coverage. *IEEE Transactions on Wireless Communications*, 5(11):3182–3192, November 2006.

[55] Chih fan Hsin and Mingyan Liu. Network coverage using low-duty cycled censors: random and coordinated sleep algorithms. *in Proc. International Symposium on Information Processing in Sensor Networks (IPSN) 2004*, 2004.

[56] Byungseok Kang, Phan Khanh Ha Nguyen, Vyacheslav Zalyubovskiy, and Hyunseung Choo. A Distributed Delay-Efficient Data Aggregation Scheduling for Duty-Cycled WSNs. *IEEE Sensors Journal*, 17(11):3422–3437, June 2017.

[57] Hongseok Yoo, Moonjoo Shim, and Dongkyun Kim. Dynamic Duty-Cycle Scheduling Schemes for Energy-Harvesting Wireless Sensor Networks. *IEEE Communications Letters*, 16(2):202–204, February 2012.

[58] Shaobo Wu, Jianwei Niu, Wusheng Chou, and Mohsen Guizani. Delay-Aware Energy Optimization for Flooding in Duty-Cycled Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 15(12):8449–8462, December 2016.

[59] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[60] M. Nagajothy and S. Radha. Network lifetime enhancement in wireless sensor network using network coding. *International Conf. on Control, Automation, Communication and Energy Conservation*, 2009.

[61] Y. Hong, J. Xu, and C. Jiang. Lifetime Maximisation in Wireless Sensor Networks with Network Coding. *International Conf. on Wireless Communications, Networking and Mobile Computing (WiCom)*, 2007.

[62] P.M. Glatz, L.B. Hormann, C. Steger, and R. Weiss. Implementing autonomous network coding for wireless sensor network applications. *18th International Conference on Telecommunications (ICT)*, 2011.

[63] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear Network Coding. *IEEE Transactions on Information Theory*, 49(2):371–381, February 2003.

[64] Osameh M. Al-Kofahi and Ahmed E. Kamal. Network coding-based protection of many-to-one wireless flows. *IEEE Journal on Selected Areas in Communications*, 27(5):797–813, June 2009.

[65] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, and Jon Crowcroft. XORs in the Air: Practical Wireless Network Coding. *IEEE Transactions on Networking*, 16(3):497–510, June 2008.

[66] R. R. Rout, S. K. Ghosh, and S. Chakrabarti. Co-operative routing for wireless sensor networks using network coding. *IET Wireless Sensor Systems*, 2(2):75–85, June 2012.

[67] Christina Fragouli and Emina Soljanin. Network coding fundamentals. *Foundations and Trends® in Networking*, 2(1):1–133, 2007.

[68] Mohammad Abu Alsheikh, Dinh Thai Hoang, Dusit Niyato, Hwee-Pink Tan, and Shaowei Lin. Markov decision processes with application in wireless sesnor networks: A survey. *IEEE Communications Surveys and Tutorials*, 17(3):1239–1267, April 2015.

[69] Zhichu Lin and Mihaela van der Schaar. Autonomic and distributed joint routing and power control for delay-sensitive applications in multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, 10(1):102–113, January 2011.

[70] Jie Hao, Zheng Yao, Kui Huang, Baoxian Zhang, and Cheng Li. An energy-efficient routing protocol with controllable expected delay in duty-cycled wireless sensor networks. *IEEE International Conference on Communications (ICC), 2013*, pages 6215–6219, November 2013.

[71] Yahong Guo, Xiaohang Guo, Yanqing Zhang, Jinghua Zhu, and Jinbao Li. Opportunistic routing in multi-power wireless sensor networks. *Advances in Wireless Sensor Networks*, pages 83–96, 2013.

[72] Arupa Mohapatra, Natarajan Gautam, Srinivas Shakkottai, and Alex Sprintson. Network Coding Decisions for Wireless Transmissions With Delay Consideration. *IEEE Transactions on Communications*, 62(8):2965–2976, August 2014.

[73] Muhammad Ali Murtaza and Muhammad Tahir. Optimal Data Transmission and Battery Charging Policies for Solar Powered Sensor Networks using Markov Decision Process. *IEEE Wireless Communications and Networking Conference (WCNC) 2013*, July 2013.

[74] R. Madan and S. Lall. An energy-optimal algorithm for neighbor discovery in wireless sensor networks. *Mobile Networks and Applications*, 11(3):317–326, 2006.

[75] Luca Stabellini. Energy optimal neighbor discovery for single-radio single-channel wireless sensor networks. *IEEE International Symposium on Wireless Communication Systems*, pages 583–587, 2008.

[76] Sudip Misra, Rashmi Ranjan Rout, T. Raghu Vamsi Krishna, Patel Manish Kumar Manilal, and Mohammad S. Obaidat. Markov decision process-based analysis of rechargeable nodes in wireless sensor networks. *in Proc. ACM Spring Simulation Multiconference (SpringSim '10)*, 2010.

[77] Koushik Kar, Ananth Krishnamurthy, and Neeraj Jaggi. Dynamic Node Activation in Networks of Rechargeable Sensors. *IEEE/ACM Transactions on Networking*, 14(1):15–26, February 2006.

[78] Jesus Fernandez-Bes, Jesus Cid-Sueiro, and Antonio G. Marques. An mdp model for censoring in harvesting sensors: Optimal and approximated solutions. *IEEE Journal on selected areas in communications*, 33(8):1717–1729, August 2015.

[79] Rashmi Ranjan Rout, Mothku Sai Krishna, and Shalinee Gupta. Markov Decision Process-Based Switching Algorithm for Sustainable Rechargeable Wireless Sensor Networks. *IEEE Sensors Journal*, 16(8):2788–2797, April 2016.

[80] Peyman Neamatollahi, Mahmoud Naghibzadeh, and Saeid Abrishami. Fuzzy-Based Clustering-Task Scheduling for Lifetime Enhancement in Wireless Sensor Networks. *IEEE Sensors Journal*, 17(20):6837–6844, October 2017.

[81] Sariga Arjunan and Pothula Sujatha. Lifetime maximization of wireless sensor network using fuzzy based unequal clustering and ACO based routing hybrid protocol. *Applied Intelligence*, pages 1–18, November 2017.

[82] Padmalaya Nayak and Anurag Devulapalli. A Fuzzy Logic-Based Clustering Algorithm for WSN to Extend the Network Lifetime. *IEEE Sensors Journal*, 16(1):137–144, January 2016.

[83] Timothy J.Ross. *Fuzzy Logic with Engineering Applications*. Wiley, 3rd edition, 2010.

[84] Yong Ho Kim, Sang Chul Ahn, and Wook Hyun Kwon. Computational complexity of general fuzzy logic control and its simplification for a loop controller. *Fuzzy Sets and Systems*, 111(2):215–224, April 2000.

[85] Indranil Gupta, D. Riordan, and Srinivas Sampalli. Cluster-head election using fuzzy logic for wireless sensor networks. *3rd Annual Communication Networks and Services Research Conference (CNSR'05)*, May 2005.

[86] Jong-Myoung Kim, Seon-Ho Park, Young-Ju Han, and Tai-Myoung Chung. CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks. *10th International Conference on Advanced Communication Technology*, April 2008.

[87] unpei Anno, Leonard Barolli, Arjan Durresi, Fatos Xhafa, and Akio Koyama. Performance Evaluation of Two Fuzzy-Based Cluster Head Selection Systems for Wireless Sensor Networks. *Mobile Information Systems*, 4(4):297–312, December 2008.

[88] Ge Ran, Huazhong Zhang, and Shulan Gong. Improving on LEACH Protocol of Wireless Sensor Networks Using Fuzzy Logic. *Journal of Information and Computational Science*, 7(3):767–775, March 2010.

[89] Hakan Bagci and Adnan Yazici. An energy aware fuzzy unequal clustering algorithm for wireless sensor networks. *International Conference on Fuzzy Systems*, July 2010.

[90] Qingjian Ni, Qianqian Pan, Huimin Du, Cen Cao, and Yuqing Zhai. A Novel Cluster Head Selection Algorithm Based on Fuzzy Clustering and Particle Swarm Optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(1):76–84, January 2017.

[91] Roomana Yousaf, Rizwan Ahmad, Waqas Ahmed, and Abdul Haseeb. Fuzzy Power Allocation for Opportunistic Relay in Energy Harvesting Wireless Sensor Networks. *IEEE Access*, 5:17165–17176, September 2017.

[92] Fred Stann and John Heidemann. RMST: reliable data transport in sensor networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003*, pages 102–112, June 2003.

[93] Yogesh G. Iyer, Shashidhar Gandham, and S. Venkatesan. STCP: a generic transport layer protocol for wireless sensor networks. *Proceedings 14th International Conference on Computer Communications and Networks, 2005*, pages 449–454, October 2005.

[94] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):862–872, April 2005.

[95] Hongwei Zhang, AnishArora, Young-riChoi, and Mohamed G. Gouda. Reliable bursty convergecast in wireless sensor networks. *Computer Communications*, 30(13):2560–2576, September 2007.

[96] Faisal Karim Shaikh, Abdelmajid Khelil, Azad Ali, and Neeraj Suri. TRCCIT: Tunable reliability with Congestion Control for Information Transport in Wireless Sensor Networks. *The 5th Annual ICST Wireless Internet Conference (WICON) 2010*, pages 1–9, April 2010.

[97] Tuan Le, Wen Hu, Peter Corke, and Sanjay Jha. ERTP: Energy-efficient and Reliable Transport Protocol for data streaming in Wireless Sensor Networks. *Computer Communications*, 32:1154–1171, January 2009.

[98] Seung-Jong Park, Ramanuja Vedantham, Raghupathy Sivakumar, and Ian F. Akyildiz. GARUDA: Achieving Effective Reliability for Downstream Communication in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 7(2):214–230, February 2008.

[99] Lorenzo Keller, Emre Atsan, Katerina Argyraki, and Christina Fragouli. SenseCode: Network coding for reliable sensor networks. *ACM Transactions on Sensor Networks*, 9(2):25:1–25:20, March 2013.

[100] Qi Zhang and Anfeng Liu. An unequal redundancy level-based mechanism for reliable data collection in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 1(258):1–22, October 2016.

[101] Stefano Basagni, Alessio Carosi, Emanuel Melachrinoudis, Chiara Petrioli, and Z. Maria Wang. Controlled Sink Mobility for Prolonging Wireless Sensor Networks Lifetime. *Wireless Networks*, 14:831–858, 2008.

[102] Faisal Karim Shaikh and Sherali Zeadally. Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 55:1041–1054, March 2016.

[103] Shenqiu Zhang, Alireza Seyedi, and Biplab Sikdar. An Analytical Approach to the Design of Energy Harvesting Wireless Sensor Nodes. *IEEE Transactions on Wireless Communications*, 12(8):4010–4024, August 2013.

[104] Satyajayant Misra, Nahid Ebrahimi Majd, and Hong Huang. Approximation Algorithms for Constrained Relay Node Placement in Energy Harvesting Wireless Sensor Networks. *IEEE Transactions on Computers*, 63(12):2933–2947, December 2014.

[105] Volodymyr Pryyma, Damla Turgut, and Ladislau Boloni. Active time scheduling for rechargeable sensor networks. *Computer Networks*, 54:631–640, 2010.

[106] Zhu Ren, Peng Cheng, Jiming Chen, David K. Y. Yau, and Youxian Sun. Dynamic activation policies for event capture in rechargeable sensor network. *IEEE transaction on parallel and distributed systems*, 2010.

[107] Mor Harchol-Balter. *Performance Modeling and Design of computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[108] Pandurang Kamat, Wenyuan Xu, Wade Trappe, and Yanyong Zhang. Temporal privacy in wireless sensor networks: Theory and practice. *ACM Transactions on Sensor Networks*, 5(4):1–24, November 2009.

[109] Jong-Suk Ahn, Jong-Hyuk Yoon, and Kang-Woo Lee. Performance and Energy Consumption Analysis of 802.11 with FEC Codes over Wireless Sensor Networks. *Journal of Communications and Networks*, 9(3):265–273, September 2007.

[110] Wei Shen, Tingting Zhang, Filip Barac, and Mikael Gidlund. PriorityMAC: A Priority-Enhanced MAC Protocol for Critical Traffic in Industrial Wireless Sensor and Actuator Networks. *IEEE Transactions on Industrial Informatics*, 10(1):824–835, February 2014.

[111] Yuxin Liu, Anfeng Liu, Yuxuan Li, Zhetao Li, Young june Choi, Hiroo Sekiya, and Jie Li. APMD: A fast data transmission protocol with reliability guarantee for pervasive sensing data communication. *Pervasive and Mobile Computing*, 41:413–435, April 2017.

[112] Arijit Roy, Pushpendu Kar, and Sudip Misra. Detection of Dumb Nodes in a Stationary Wireless Sensor Network. *2014 Annual IEEE India Conference (INDICON)*, pages 1–6, December 2014.

[113] Pushpendu Kar, Arijit Roy, Sudip Misra, and Mohammad S. Obaidat. Energy-Efficient Connectivity Re-establishment in WSN in the Presence of Dumb Nodes. *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1485–1490, June 2015.

[114] Arjit Roy, Pushpendu Kar, Sudip Misra, and Mohammad S. Obaidat. D3: distributed approach for the detection of dumb nodes in wireless sensor networks. *International Journal of Communication Systems*, 30(1), January 2014.

[115] Yuanyuan Zeng, Deshi Li, and Athanasios V Vasilakos. Real-time data report and task execution in wireless sensor and actuator networks using self-aware mobile actuators. *Computer Communications*, 36(9):988–997, 2013.

[116] Hamed Yousefi, Ali Dabirmoghaddam, and Kambiz Mizanian. Score Based Reliable Routing in Wireless Sensor Networks. *IEEE Conference on Information Networking*, pages 1–5, January 2009.

[117] Frederick S.Hillier and Gerald J.Lieberman. *Introduction to Operations Research*. McGraw-Hill Higher Education, seventh edition, 2001.

[118] Ayon Chakraborty, Rashmi Ranjan Rout, Aveek Chakrabarti, and Soumya K. Ghosh. On network lifetime expectancy with realistic sensing and traffic generation model in wireless sensor networks. *IEEE Sensors Journal*, 13(7):2771–2779, July 2013.

[119] Ismail Salhi, Y. G. Doudane, S. Lohier, and G. Roussel. Network coding for event-centric wireless sensor networks. *IEEE International Conference on Communications (ICC 2010)*, 2010.

[120] Gentian Jakllari and Stephan Eidenbenz. Link positions matter: A noncommutative routing metric for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 11(1):61–72, January 2012.

[121] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[122] Prasenjit Chanak and Indrajit Banerjee. Fuzzy rule-based faulty node classification and management scheme for large scale wireless sensor networks. *Expert Systems with Applications*, 45:307–321, March 2016.

[123] M.J. Handy, M. Haase, and D. Timmermann. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. *IEEE 4th International Workshop on Mobile and Wireless Communications Network*, pages 368–372, December 2012.

[124] Mohamed Elshrkawey, Samiha M. Elsherif, and M. Elsayed Wahed. An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks. *Journal of King Saud University - Computer and Information Sciences*, 30(2):259–267, April 2018.

[125] Xiaoguo Ye and Weifa Liang. Charging Utility Maximization in Wireless Rechargeable Sensor Networks. *Wireless Networks*, 23(7):2069–2081, October 2017.

[126] Yin Wu and Wenbo Liu. Routing protocol based on genetic algorithm for energy harvesting-wireless sensor networks. *IET Wireless Sensor Systems*, 3(2):112–118, June 2013.

[127] Fayçal Ait Aoudia, Matthieu Gautier, and Olivier Berder. Distributed Computation of Fair Packet Rates in Energy HarvestingWireless Sensor Networks. *IEEE Wireless Communications Letters*, 6(5):626–629, October 2017.

[128] Joaquin Recas Piorno, Carlo Bergonzini, David Atienza, and Tajana Simunic Rosing. Prediction and management in energy harvested wireless sensor nodes. *Wireless VITAE'09*, pages 6–10, May 2009.

[129] Ali Dabirmoghaddam, Majid Ghaderi, and Carey Williamson. Energy- Efficient Clustering in Wireless Sensor Networks with Spatially Correlated Data. *IEEE INFOCOM 2010*, 2010.

[130] Wei Ye, John Heidemann, and Deborah Estrinsss. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, June 2004.

List of Publications

1. **Sai Krishna Mothku** and Rashmi Ranjan Rout, ”Adaptive buffering using Markov Decision Process in tree-based Wireless Sensor and Actor Networks”, *Computers & Electrical Engineering, Elsevier*, Vol.71, page 901-914, 2018.
2. **Sai Krishna Mothku** and Rashmi Ranjan Rout, ”Markov Decision Process and Network Coding for Reliable Data Transmission in Wireless Sensor and Actor Networks”, *Pervasive and Mobile Computing*, *Elsevier*, Vol.56, page 29-44, 2019.
3. **Sai Krishna Mothku** and Rashmi Ranjan Rout, ”A Fuzzy Based Energy-Aware Routing Protocol for a Heterogeneous Sensor Network”, *Journal of Computer Networks and Communications, Hindawi*, Vol. 2019, Article ID: 3237623, 11 pages, 2019, DOI:10.1155/2019/3237623.
4. **Sai Krishna Mothku** and Rashmi Ranjan Rout, ”Fuzzy Logic Based Adaptive Duty Cycling for Sustainability in Energy Harvesting Sensor Actor Networks”, *Journal of King Saud University - Computer and Information Sciences, Elsevier*, DOI: 10.1016/j.jksuci.2018.09.023.