Trust Based Congestion Control Framework for Wireless Sensor Networks

Thesis submitted by

Arpita Bhattacharyya (Chakraborty)

Doctor of Philosophy (Engineering)

Department of Electronics and Telecommunication Engineering,

Faculty Council of Engineering & Technology

Jadavpur University

Kolkata, India

2016

JADAVPUR UNIVERSITY
KOLKATA – 700 032, INDIA

INDEX No. 57/12/E

1. Title of the Thesis:

Trust Based Congestion Control Framework for Wireless Sensor Networks

2. Name, Designation & Institution of the Supervisor/s:

Prof. M. K. Naskar
Professor
Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata – 700 032

Dr. A. Karmakar
Associate Professor
Department of Electronic Science
University of Calcutta
Kolkata 700 009

3. List of Publications:

International Conferences

1) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Congestion Aware Protocol Suite for Energy Efficient Routing in Wireless Sensor Networks Using Trust Based Framework (CET-PS)", in the Proceedings of the Sixth International IEEE Conference CODEC 2015, Kolkata, December 2015.

2) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Trust Based Congestion Aware Hybrid Ant Colony Optimization Algorithm for Energy Efficient Routing in Wireless Sensor Networks (TC-ACO)", in the Proceedings of the Fifth International IEEE Conference ICoAC -13, Anna University, Chennai, December 2013. **Best Paper Award.**

3) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Trust Based Fuzzy Algorithm for Congestion Control in Wireless Multimedia Sensor Networks (TFCC)", in the Proceedings of the Second International IEEE Conference, ICIEV -13, May 2013, Dhaka, Bangladesh. **Best Presenter Award.**

4) A. Chakraborty, A. Raha, S. Maity, A. Karmakar, M. K. Naskar, "A Fuzzy Based Trustworthy Route Selection Method using LSRP in Wireless Sensor Networks (FTRSP)", in the Proceedings of the Second ACM International Conference CCSEIT-12, Coimbatore, October 2012.

5) A. Raha, A. Chakraborty, M. K. Naskar , O. Alfandi, D. Hogrefe, "A Novel Indirect Trust based Link State Routing Scheme using a Robust Route Trust Method for Wireless Sensor Networks", in the Proceedings of the Fifth International IEEE Conference, NTMS- 2012, May, Istanbul, 2012.

National Conference

1) A. Chakraborty, A. Raha, M. K. Naskar, "Trust Management in Wireless Sensor Networks - A Survey" , in the proceedings of the National Conference on Frontiers of Communication and Instrumentation Engineering (COIN-2011), Techno India, Technical Co-Sponsors : IEEE Kolkata Chapter, IEI and ISA Kolkata Section, November, 2011.

International Journal

1) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "Trust Integrated Congestion Aware Energy Efficient Routing for Wireless Multimedia Sensor Networks (TCEER)", International Journal of Computing and Information Technology (CIT), Vol. 23, No.2, Pp. 95 – 109, Published by University Computing Centre, Zagreb, Croatia, June 2015.

2) A. Raha, A. Paul, A. Chakraborty, A. Karmakar, M. K. Naskar, "A Genetic Algorithm Inspired Load Balancing Protocol for Congestion Control in Wireless Sensor Networks using Trust based Routing Framework (GACCTR)", International Journal Computer Network and Information Security (IJCNIS), Published Online by Modern Education and Computer Science Press (MECS), Hong Kong, DOI: 10.5815 / ijcnis.2013.09.02, July 2013.

3) A. Chakraborty, S. Kundu, S. Mukherjee, A. Karmakar, M. K. Naskar, "FPGA Implementation of a Fuzzy Coprocessor for Trust Based Congestion Aware Data Routing in Wireless Sensor Networks (TCR-FC)", Wireless Networks, Springer (Communicated).

4. List of Patents:

Nil

5. List of Presentations in National / International / Conferences / Workshops:

1) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Congestion Aware Protocol Suite for Energy Efficient Routing in Wireless Sensor Networks Using Trust Based Framework (CET-PS)", Sixth International IEEE Conference CODEC, Kolkata, December 2015.

2) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Trust Based Congestion Aware Hybrid Ant Colony Optimization Algorithm for Energy Efficient Routing in Wireless Sensor Networks (TC-ACO)", Fifth International IEEE Conference ICoAC-13, Anna University, Chennai, December 18-20, 2013.

Best Paper Award.

3) A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Trust Based Fuzzy Algorithm for Congestion Control in Wireless Multimedia Sensor Networks (TFCC)", Second International IEEE Conference, ICIEV-13, Dhaka, May 2013**.**

Best Presenter Award

4) S. Mitra, J. Banerjee, A. Chakraborty, M. K. Naskar, "Data Gathering in Wireless Sensor Network using Realistic Power Control", Fifth International ACM Conference on Communication, Computing & Security (ICCCS2011), NIT Rourkela, Feb 2011.

5) A. Chakraborty, A. Raha, M. K. Naskar, "Trust Management in Wireless Sensor Networks: A Survey", National Conference on Frontiers of Communication and Instrumentation Engineering (COIN-2011), Techno India, Technical Co-Sponsors: IEEE Kolkata Chapter, IEI and ISA Kolkata Section, November 2011.

CERTIFICATE FROM THE SUPERVISOR/S

This is to certify that the thesis entitled "Trust Based Congestion Control Framework for Wireless Sensor Networks", submitted by Smt Arpita Bhattacharyya (Chakraborty), who got her name registered on 4th September 2012 for the award of Ph. D. (Engg.) degree of Jadavpur University is absolutely based upon her own work under the supervision of Prof. (Dr.) M. K. Naskar and Dr. A. Karmakar and that neither her thesis nor any part of the thesis has been submitted for any degree / diploma or any other academic award anywhere before.

1. _____

Prof. (Dr.) M. K. Naskar

2. _____

Dr. A. Karmakar

Dedicated to my parents

Acknowledgement

I would like to express my sincere gratitude and deepest thanks to my supervisor Prof. M. K. Naskar for his valuable guidance, continuous support and encouragement during my research work. I appreciate all his immense knowledge, contributions of time, suggestions, motivations and patience for my research. He provided me all facilities to carry out the research in his Advanced Digital and Embedded Systems Laboratory at Jadavpur University. I am also extremely grateful to my co-supervisor, Dr. A. Karmakar for his enormous support in all the time of my research. He always motivated me and made himself available to clarify my doubts. Without his help it would not be possible to conduct this research. I consider it as a great opportunity to do my doctoral programme under the guidance of my supervisor and co-supervisor and to learn from their research expertise.

I gratefully acknowledge the contributions of UG students of Jadavpur University, Mr. Arnab Raha, (at present, Research Scholar, School of Electrical and Computer Engineering, Purdue University, USA.), Mr. Srinjoy Ganguly (presently, at IIM Ahmedabad), Mr. Soham Mukherjee and Mr. Sayak Kundu for their helps in conducting simulation of the proposed algorithms.

I would like to thank Prof. P. Venkateswaran, Head of the Department, Electronics and Telecommunication Engineering, Jadavpur University. I would like to thank all members of the board of research studies for their support at various phases of the program and their insightful comments which incented me to widen my research from various perspectives.

I would like to convey my deepest thanks to Dr. Swarup Mitra, Dr. Shaik Sahil Babu, Mr. Rajarshi Midhya, Mr. Rathindra Nath Biswas and Mr. Asim Maity for sharing their valuable knowledge and useful discussions whenever required.

I gratefully acknowledge the support and encouragement received from Prof. A. K. Datta, Head of the Department, Electronics and Communication Engineering, Techno India Salt Lake, Prof. (Dr.) Chandan K. Bhattacharyya, Principal, Techno India Salt Lake and Prof. (Dr.) R. Paladhi, Director, Techno India Salt Lake, throughout my

research work. I would like to appreciate the help and support received from my colleagues at Techno India, Salt Lake.

I would like to covey my thanks to all my dear friends for their continuous support and motivation during my research work.

I am deeply indebted to my parents, Dr. Arun Prokash Bhattacharyya and Mrs. Anima Bhattacharyya for their unconditional love, support, patience, tolerance, motivation and encouragement for my research. I gratefully acknowledge the contributions of my sister Mrs. Archita Bhattacharyya (Saha) who continuously motivated me, even during hard time. I would like to appreciate the support received from my brother in law Biswajit Saha and nephew Mr. Upayan Saha.

Finally, I would like to thank my husband Mr. Shakti Chakraborty and my beloved sons Mr. Shounak Chakraborty and Mr. Sreejit Chakraborty for their invaluable patience, endless support and encouragement, without which it would not be possible to conduct this research.

I convey my sincere thanks to all.

Jadavpur University,
Kolkata.

Arpita Bhattacharyya (Chakraborty)

Abstract

Wireless Sensor Networks (WSNs) consist of tiny, low cost sensor nodes with limited processing capability and communication bandwidth which are deployed in large number, especially in remote and hostile areas. Typically, sensor nodes are battery powered and replacement or recharge of battery is not feasible in most of the cases. The self-organized sensor nodes form a temporary network without predefined network infrastructure and centralized network administration. The unique features of WSNs make them popular in a variety of application areas such as climatic data gathering, seismic and acoustic underwater monitoring, industrial control and monitoring, intelligent agriculture, surveillance and security, military applications, health care and many more.

Sensor nodes are generally operated in standby mode and then, suddenly become active after detecting event in the surrounding area. The resulting effect is the generation of large and sudden impulses of data which would be delivered to the sink or the base station without disrupting the performances of the sensing operations. When large numbers of sensor nodes are active simultaneously in transmit mode, there is a high chance of packet collisions followed by network congestion. It is one of the major bottleneck in resource constrained WSNs, especially for large networks, where the traffic loads exceed the available capacity of the resources. Congestion causes buffer overflow, packet drops and re-transmission of packets due to the limited buffer size of the sensor node. It reduces precious battery life of the sensor nodes due to the additional computation and communication overhead, which in turn decreases network lifetime, degrades network performance and Quality of Services. Moreover, the special characteristics of WSNs make them vulnerable to various types of security attacks and some security attacks have direct impact on network congestion. For example, HELLO flood attacks, Jamming attacks, Sybil attacks and Node replication attacks aggravate congestion by flooding the network with fake messages, jamming intermittently, re-transmitting same message several times and

creating false node identification respectively. In this way, the congestion created by the faulty malicious nodes plays an important role in the overall congestion of the WSNs.

The researchers throughout the globe have proposed several energy efficient routing algorithms to maximize the network lifetime. But unfortunately, most of the conventional data gathering and routing algorithms do not consider the congestion phenomenon and the impact of the security threats. Some congestion control algorithms for WSNs are reported in the literature, but most of them do not consider the role of the malicious nodes. The traditional cryptographic security solutions are not suitable for resource limited WSNs. Nowadays, the concept of trust is considered as an alternative method, which has been used for getting secured and trustworthy data routing. The integration of trust and congestion control in data routing is the new idea, where the faulty malicious nodes are detected and isolated from the data routing path, causing reduction of network congestion. This is the largely open research domain and very few research efforts have been reported in this direction. To cater this problem, the present research study on trust based congestion control has been undertaken which has a lot of potential to develop in the near future.

In this thesis, the merits of the trust based congestion control over the conventional congestion control algorithms, in terms of network lifetime and throughput, have been argued. In this context, the types of congestion and their effects, basic principle of congestion control and challenges of congestion control in WSNs have been discussed. The concept of trust is discussed in brief and the relation of congestion control with trust management is explained.

In *Chapter 1*, problem overview, objective and the motivation of the present research study is discussed.

Chapter 2 includes literature survey and review of the existing congestion control algorithms. The commonly occurred security attacks in WSNs are discussed in brief and the existing trust evaluation methods are also studied extensively in this chapter.

In *Chapter 3,* two new trust integrated data routing algorithms, ITLSRP and FTSRP are presented by using Link State Routing Protocol as the basic data routing scheme.

In *Chapter 4,* a new congestion control protocol is proposed, where selection of routing path is modeled on the basis of traffic sharing with the help of Genetic Algorithm.

In *Chapter 5,* two new fuzzy algorithms (TFCC and TCEER) for trust based congestion control in Wireless Multimedia Sensor Networks are proposed where multimedia sensor nodes are deployed for multimedia applications.

In *Chapter 6,* a new approach for congestion aware energy efficient data routing algorithm (TC-ACO) is presented by utilizing Ant Colony Optimization technique.

In *Chapter 7,* CET-PS algorithm is discussed, in which three of our previously proposed trust based congestion aware data routing algorithms (TFCC, TCEER and TC-ACO) are integrated into a single protocol suite, where the routing path is selected adaptively on the basis of the congestion status of the sensor node and the parameter called Composite Protocol Efficiency (CPE).

In *Chapter 8,* a new trust based congestion control algorithm (TCR-FC) is proposed, where dedicated state-of-the-art Fuzzy Co-processor (FCP) architecture is designed for fuzzy computation. The proposed model is implemented in Field Programmable Gate Array.

Finally, in *Chapter 9,* the thesis is concluded by summarizing the overall findings. It also includes suggestions for the future direction of research in this domain.

In brief, the main contributions of the thesis are listed as:

- Detailed study of the existing congestion control algorithms and trust based security solutions, explanation of the requirement of trust based congestion control in WSNs.
- Proposal of two new trust integrated data routing algorithms (ITLSRP and FTSRP).
- Proposal of six novel algorithms (GACCTR, TFCC, TCEER, TC-ACO, CET-PS and TCR-FC) for trust based congestion control in WSNs.
- Verification of the merits of the proposed trust based congestion control models over the conventional congestion control algorithms, in terms of network lifetime and throughput of the network, using simulation and performance analysis.

Table of Contents

List of Figures

Chapter 4

Chapter 5

List of Tables

Chapter 6

Chapter 8

List of Abbreviations

ACO	Ant Colony Optimization
AD	Average Delay
ADC	Analog to Digital Converter
ATSR	Ambient Trust Sensor Routing
BLIFE	Battery Life
BS	Base Station
BTR	Best Trustworthy Route
BRS	Best Route Selection
CCI	Complementary Congestion Index
CEE	Composite Energy Efficiency
CET-PS	Congestion-Energy-Trust Protocol Suite
CMOS	Complementary Metal Oxide Semiconductor
CN	Congestion Notification
CODA	Congestion Detection and Avoidance
COG	Centre of Gravity
CONFIDANT	Cooperation Of Nodes, Fairness In Dynamic Ad-hoc Networks
CPE	Composite Protocol Efficiency
CPTE	Composite Packet Transmission Efficiency
CPTR	Control Packet Transmission Ratio
CSMA	Carrier Sense Multiple Access
DPTR	Data Packet Transmission Ratio
DT	Direct Trust
DTLSRP	Direct Trust-dependent Link State Routing Protocol
EDM	Energy Distance Metric
EE	Energy Efficiency
ESRT	Event–to-Sink Reliable Transport
FCP	Fuzzy Coprocessor
FIS	Fuzzy Inference System
FL	Fuzzy Logic
FLC	Fuzzy Logic Controller
FPGA	Field Programmable Gate Array

FTRSP	Fuzzy based Trustworthy Route Selection Protocol
GA	Genetic Algorithm
GMTMS	Geometric Mean based Trust Management System
GPSR	Greedy Perimeter Stateless Routing
H	High
HC	High Congestion
HCC	High Complementary Congestion Index
HCCP	Hybrid Congestion Control Protocol
HCEE	High Composite Energy Efficiency
HCPE	High Composite Protocol Efficiency
HCPTE	High Composite Packet Transmission efficiency
HD	High Delay
HDL	Hardware Description Language
HE	High Energy
HT	High Trust
IT	Indirect Trust
ITLSRP	Indirect Trust based Link State Routing Protocol
L	Low
LC	Low Congestion
LCC	Low Complementary Congestion Index
LCEE	Low Composite Energy Efficiency
LCPE	Low Composite Protocol efficiency
LCPTE	Low Composite Packet Transmission Efficiency
LD	Low Delay
LE	Low Energy
LT	Low Trust
LSP	Link State Packet
LSRP	Link State Routing Protocol
LLC	Link Level Congestion
LUT	Look Up Table
M	Medium
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network

MC	Medium Congestion
MCC	Medium Complementary Congestion Index
MCEE	Medium Composite Energy Efficiency
MCPE	Medium Composite Protocol Efficiency
MCPTE	Medium Composite Packet Transmission Efficiency
ME	Medium Energy
MF	Membership Function
MRA	Maximum Retransmission Attempts
MT	Medium Trust
NLC	Node Level Congestion
NP	Node Potential
OOR	Optimal Operating Region
PCCP	Priority-based Congestion Control Protocol
PEGASIS	Power Efficient Gathering in Sensor Information System
PRR	Packet Reception Ratio
PSFQ	Pump Slowly Fetch Quickly
PTE	Packet Transmission Efficiency
QCCP-PS	Queue-based Congestion Control Protocol with Priority Support
QoS	Quality of Service
RAU	Rate Adjustment Unit
RS	Route Selection
RT	Route Trust
RTCM	Route Trust Congestion Metric
SMTR	Selection of the Most Trusted Route
SPINS	Security Protocols for Sensor Networks
TAODV	Trusted Ad hoc On Demand Distance Vector
TARA	Topology Aware Resource Adaptation
TARF	Trust Aware Routing Framework
TCEER	Trust-integrated Congestion-aware Energy Efficient Routing
TCM	Trust Congestion Metric
TCR-FC	Trust-based Congestion –aware Routing with Fuzzy Coprocessor
TDMA	Time Division Multiple Access
TDSR	Trust-aware Dynamic Source Routing

TDT	Total Direct Trust
TFCC	Trust-based Fuzzy algorithm for Congestion Control
TILSRP	Trust Integrated Link State Routing Protocol
TM	Trust Metric
TRANS	Trust Routing for Location Aware Sensor Networks
TRC	Traffic Rate Controller
TT	Total Trust
TTH	Trust Threshold
VHCC	Very High Complementary Congestion Index
VHCEE	Very High Composite Energy Efficiency
VHCPE	Very High Composite Protocol Efficiency
VHCPTE	Very High Composite Packet Transmission Efficiency
VL	Very Low
VLC	Very Low Congestion
VLCC	Very Low Complementary Congestion Index
VLCEE	Very Low Composite Energy Efficiency
VLCPE	Very Low Composite Protocol Efficiency
VLCPTE	Very Low Composite Packet Transmission Efficiency
VLD	Very Low Delay
VLE	Very Low Energy
VLSI	Very Large Scale Integration
VLT	Very Low Trust
WCCP	WMSN Congestion Control Protocol
WSN	Wireless Sensor Network
WMSN	Wireless Multimedia Sensor Network

CHAPTER 1

Introduction

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are self-configured ad-hoc networks without centralized administration, comprising of large number of inexpensive, battery powered, tiny sensor nodes which are deployed randomly in a physical space, indoor or outdoor, even in remote and hostile areas [IFA02], [DCD04]. The sensor nodes, also called motes, are capable of sensing events from the surrounding environment, able to convert analog data to the digital format and then transmit data in collaborative manner to the sink node or base station which has higher processing capability, power and transmission range. Finally, the task manager or the user from the external network can collect the processed data directly from the base station through the wireless radio link [IFA02], [DCD04].

The dense networks of spatially distributed wireless sensing techniques offer unique advantages over the traditional centralized approaches which include [FZL04]:

- Signal to noise ratio improvement by reducing average distances from sensor to source of signal or target.
- Increased energy efficiency in communications due to multi-hop topology of the network.

- Improved flexibility due to infrastructure less configuration and self-organizing capabilities.
- Improved robustness, flexibility, fault tolerance and scalability due to the redundancy in the network.
- Rapid deployment characteristics in remote and hostile environment.
- Additional relevant information from other sensors can be aggregated during multi-hop transmission through in-network processing.
- Cost effective solution provider.

However, realization of WSN faces the following challenges [IFA02], [DCD04], [FLZ04]:

- Sensor nodes have small sensing areas of few meters.
- Poor communication bandwidth.
- Limited memory, limited storage space and limited computational capabilities.
- Constraints on power consumption – wireless sensor nodes are battery powered and replacement of battery is very difficult when deployed in inaccessible places.
- Limited support for networking - There are no universal routing protocols or central registry services.
- Network topology is changed frequently due to fading and node failures. Each node acts as a router as well as an application host.
- Limited support for software development - It is real time and distributed in nature which involves dynamic collaboration among sensor nodes. It should handle multiple competing events.
- Unreliable – sensor data packets are transferred from source node to destination node through wireless connection. Hence it is unreliable in nature.
- Prone to security attacks and physical damages – sensor nodes are deployed outside in hostile environment. Unlike other types of networks, it is often impossible to prevent security attacks and physical damages of the sensor nodes.

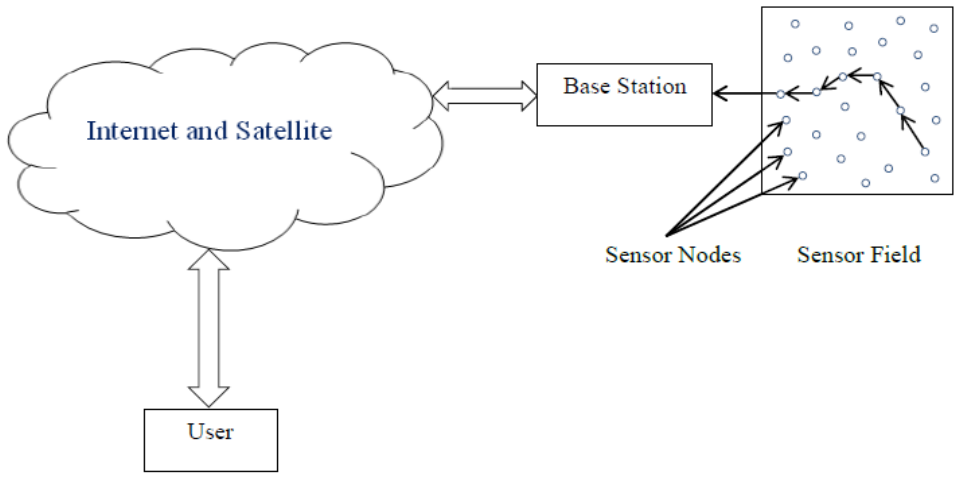


Figure 1.1: Wireless Sensor Network.

The pictorial view of a typical WSN is presented in Fig. 1.1 where the sensor nodes are deployed randomly in a square sensor field.

1.1.1 Types of Wireless Sensor Networks

WSNs face various challenges depending upon the deployment conditions and physical environment. Considering that WSNs are classified into five categories [JYB08], as given below:

- **Terrestrial WSNs** – Terrestrial WSNs consist of hundreds to thousands sensor nodes and one or more sink nodes, deployed randomly or pre planned way, into the target land area. This is the oldest version of the WSN.

- **Underground WSNs** – Underground WSNs are used to monitor the environment of the target underground areas, where the sensor nodes are buried underground or inside a cave or mine. The additional sink node is placed above the ground for connecting WSNs to the external world. Compared to the terrestrial WSNs, underground WSNs are more expensive in terms of equipment, deployment and maintenance.

- **Underwater WSNs** – Underwater WSNs consist of sensor nodes and autonomous vehicles deployed underwater. The wireless communications are established inside the water through transmission of acoustic waves.

- **Multimedia WSNs** - Wireless Multimedia Sensor Networks (WMSNs) is a new, emerging field of WSNs which contain multimedia sensor nodes having low cost CMOS cameras, microphones and other sensor devices for retrieving video and audio streams, still images and scalar sensor data from the physical environment [IFA07]. WMSNs are heterogeneous in nature and generate large volume of high bit rate multimedia data. This type of networks are characterized by new challenges like high bandwidth demand, data processing and compression techniques, congestion control and Quality of Service provisioning.

- **Mobile WSNs** – Mobile WSNs consist of sensor nodes that can move by itself and sense data from the environment. Localization and navigation are the new challenges for Mobile WSNs.

Depending upon the mode of operation and functionality, WSNs can be classified into two types.

- **Proactive Networks:** WSNs where the nodes periodically switch on their sensors to sense the environment and transmit the data of interest at regular interval of time are known as proactive networks. They are well suited for applications requiring periodic data monitoring.

- **Reactive Networks**: WSNs where the nodes react immediately to the sudden and drastic changes in the value of a sensed attribute are known as reactive networks. They are well suited for time-critical applications.

1.1.2 Applications of Wireless Sensor Networks

The state-of-the art sensor technology has immense potential for a variety of applications in diverse fields. The two most important application domains are:

- Monitoring and surveillance applications.
- Tracking applications.

Some major application areas in WSNs [JYB08], [IFA02] are listed in Table 1.1.

Table 1.1: Major application areas in WSNs

Monitoring / Surveillance Applications	**Tracking Applications**
Military area monitoring, surveillance	Object tracking
Indoor and outdoor environment monitoring, Climatic data gathering and weather monitoring	Vehicle tracking
Smart home monitoring and security surveillance	Traffic tracking
Industrial monitoring, process automation and structural monitoring	Tracking of Human beings
Health care and wellness monitoring	Military enemy tracking
Water / waste water monitoring	Animals tracking
Seismic and acoustic underwater monitoring / volcano monitoring	Passive localization and tracking
Inventory location monitoring	
Habitat and Animal monitoring	
Plant monitoring, agriculture monitoring	

1.1.3 Architecture of Wireless Sensor Networks

A typical sensor node consists of four basic components, namely, sensing unit, processor and storage unit, transceiver unit and power supply unit [IFA02], [ABV02]. The components of the sensor node are represented in Fig. 1.2.

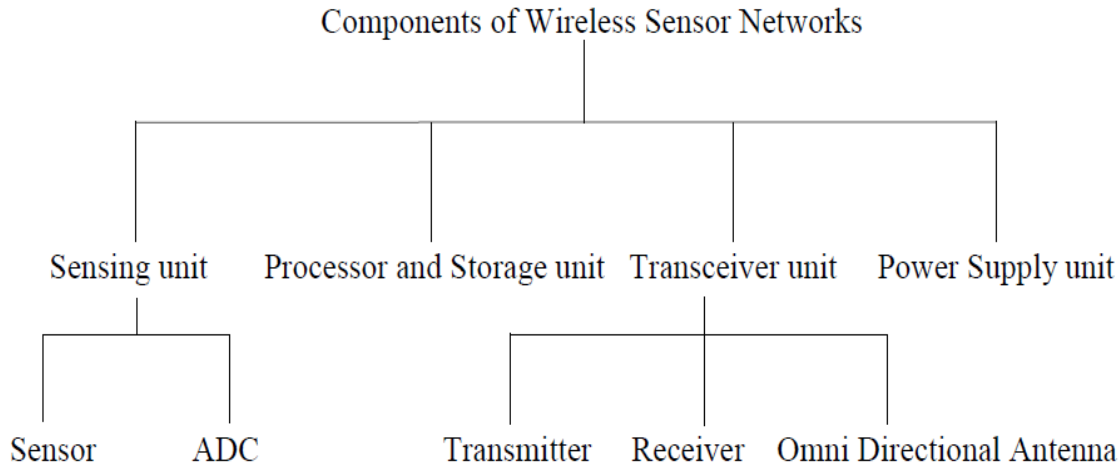


Figure 1.2: Basic components of WSNs.

The sensing unit consists of two sub units such as sensor and ADC (Analog to Digital Converter). The analog signals obtained from the sensor are converted to digital data through ADC. The data is processed by the processor unit which is associated with a small storage unit. The processor and storage unit are responsible for the control of the sensors and the execution of communication protocols. Transceiver consists of radio transmitter, receiver and an omnidirectional antenna which connects the node with the network. The non-rechargeable battery is generally used as the power unit which provides power to each component.

The pictorial view of 2.4 GHz Crossbow make IRIS Mote module is shown in Fig. 1.3, which has improved features compared to the previous MICA motes in terms of radio range and memory space. IRIS mote has IEEE 802.15.4 standard compliant RF transceiver, supporting 250 Kbps data rate radio. Outdoor line-of-sight radio range of the IRIS Mote is approximately 500 meters between nodes without amplification. The processor board is based on the Atmel low power microcontroller ATmega 1281 which runs MoteWorks from its internal flash memory [ISM00], [ISM01].

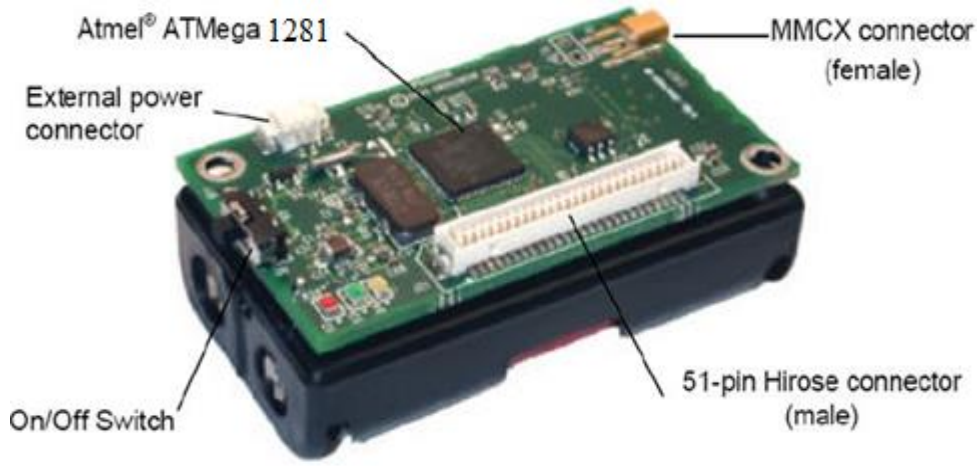


Figure 1.3: IRIS mote module.

IRIS mote module is supported by MoteWorks WSN platform which is based on the open source TinyOS operating system [PLD06]. TinyOS is specially designed for low power resource constraint devices like wireless sensor networks [PLD06]. It features component-based architecture which enables rapid innovation and implementation while minimizing code size as required by severe memory constraints. The TinyOS system, libraries and applications are written in a new language, called nesC (network embedded systems C) which is a component based event driven programming language used to build applications for the TinyOS platform [NPL04]. NesC is C dialect and support C syntax TinyOS uses nesC rather than C because it provides linguistic support for the TinyOS execution and programming model.

1.2 Problem Overview

Since the last decade, WSNs are getting more and more interests among the scientific researchers, throughout the globe as a new emerging technology which has significant benefits to the society. With rapid advancement of technology, the characteristics of the WSNs are improved gradually and new directions of research are evolved. The trust based congestion control is one of such new research domain which is very much relevant in today's world. Congestion and security attacks are very common in resource constrained WSNs. The effect of congestion is characterized by buffer overflow, packet drops, retransmission of packets, unacceptable packet delays, drastic drops in network

throughput and deterioration of Quality of Service (QoS). Even, in the worst case, the network may be collapsed, where almost no packets are delivered.

Generally, sensor nodes are operated in idle mode for energy conservation and then suddenly become active to the detected events in the surrounding area [CYW03]. When large number of sensor nodes are active and try to transmit data to the base station simultaneously, packet collisions and network congestion are quite common phenomenon. Moreover, WSNs are prone to security attacks due to its broadcast nature and infrastructure less configuration [CKD03]. The faulty malicious nodes of the sensor networks aggravate congestion by flooding the network with fake messages, jamming intermittently, retransmitting the same message several times and creating false node identification.

In literature, a large number of data routing algorithms for WSNs are available. But, most of them do not consider congestion related problems and impact of security attacks. This dissertation addresses the congestion problems encountered in WSNs, focusing on the congestions obtained from the misbehaving activities of the malicious nodes.

1.3 Motivation and Objective of the Thesis

The traditional cryptographic security protocols are complex in nature and the overhead are high. So, they are not suitable for resource constraint WSNs [SSB11], [CKD03], [ARM11], [MOM08]. Nowadays, a relatively new concept, known as trust is considered by the researchers, which is light weight and used to get the secured trustworthy routing, instead of the traditional security protocols. Again, the conventional congestion control algorithms are not applicable in sensor networks due to the limitation of resources [CYW03], [JZL10]. So, the researchers have proposed some light weight congestion control algorithms, specially designed for the sensor networks. But, most of the existing congestion control algorithms mainly focus on either traffic load control or resource control or both of them. The congestion occurred due to the misbehavior of the malicious nodes are not discussed much in the literature. To cater this problem, the present research study has been undertaken.

The main objective and challenges posted in this thesis are the integration of trust based security solutions and congestion control to the data routing algorithm to enhance the performance of WSNs. This work moves a step ahead and attempts to develop trust based congestion control data routing algorithm in WSNs. It is a new concept and very few research works are reported in this direction. The goal of the present research study is to analyze the compliance and merits of the trust based congestion control by comparing with the existing congestion control algorithms for WSNs. In this context, the following works have been proposed:

- Detection of malicious nodes using the concept of trust.
- Design new model for trust based congestion control in WSNs, which would expect enhancement of network lifetime and throughput by reducing computation and communication overhead.
- Design of a new algorithm where traffic loads are distributed among various routes from source node to sink node for congestion control.
- Performance analysis and comparative study of the proposed trust based congestion control algorithms with the existing solutions.

The following sections include the brief discussions on the concept of congestion control and trust that are related to the present research topic.

1.3.1 Definition of Congestion

Network congestion is one of the major issues in all kinds of communication networks. As per Oxford English Dictionary, the word congestion means “the state of being crowded and full of traffic”. Network congestion describes a situation when the offered traffic load exceeds the available capacity of the network. As described in [LLP07], “When too many packets are contending for the same link, the queue overflows and packets dropped. When such drops become common events, the network is said to be congested”. As referred in [AST89], “When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered and the number delivered is proportional to the number sent. However, as traffic increases, the routers are no longer able to cope up with the situation and they begin losing packets. When

too many packets are present in the subnet, performance degrades. This situation is called congestion". As written in [BAF07], "An important issue in a packet switched network is congestion. Congestion in a network may occur if the load on the network, that is, the numbers of packet sent to the network is greater than the capacity of the network- the number of packets a network can handle". Simply speaking, congestion is defined as the situation when the total demand of the resources in a network is greater than the available resources of the network at a given time. Mathematically, congestion occurs, when ∑ Demands > Total Resources [ANA07].

1.3.2 Effect of Congestion

The factors for getting congestion in the communication networks are listed below:

- If the number of packets sent to the network is greater than the number of packets a network can handle,
- If resources (e.g. bandwidth) are scarce and highly in demand, and/or
- The processing and transmission speeds lag behind the speed of the incoming traffic.

The effect of congestion is characterized by,

- drastic drop in network throughput,
- packet loss,
- unacceptable packet delays,
- deterioration of Quality of Service (QoS).

The network performance as a function of the load is illustrated in Figs. 1.4 and 1.5 [ANA07]. When the load is light, packets delivery is linearly proportional to the packet sent and delay is almost unchanged. After the load reaches the network capacity (called the knee point), the packet delivery would not increase much with the load. Instead packets are queued in the buffer space. At this point, the throughput of the network suddenly drops and the packet delivery is highly decreased. The point where packet gets discarded due to buffer overflow is called cliff point.

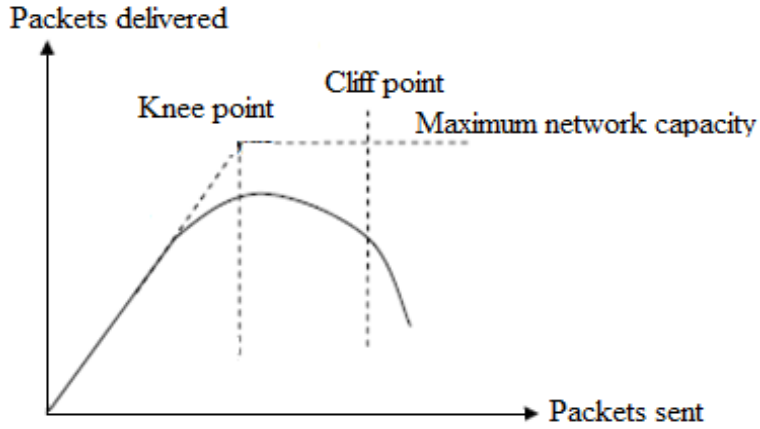


Figure 1.4: Packet delivery versus packets sent.

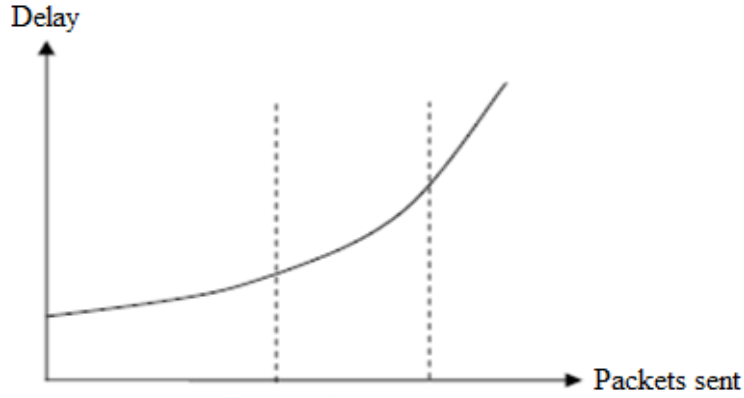


Figure 1.5: Delay versus packets sent.

1.3.3 Congestion Control and Resource Allocation

Congestion control and resource allocation are two sides of the same coin [LLP07]. The term "congestion control" is the process which describes the efforts made by the network nodes to prevent or to respond the overload conditions [LLP07]. It refers to the techniques and mechanisms that can either prevent congestion before it happens, or remove congestion after it has happened [BFA07]. The principal aspects of the congestion control are given by any one or both of the following mechanisms,

- reduce the flow of traffic entering the network, and / or
- re-direct the traffic flow away from the congested points.

The term "resource allocation" or "resource control", on the other hand, means effective and fair allocation of resources among the users [AST89]. Some sort of

congestion control is obtained through proper resource allocation. But, in practice, resources are scare relative to the demand and the network becomes congested. Hence, congestion control mechanism is needed to recover from the state of the network congestion.

1.3.4 Congestion Control and Flow Control

Congestion control and flow control are often confused due to their close relationship, but there is a basic difference between them. Flow control is the process of keeping a fast sender from over running a slow receiver [AST89]. It refers to a set of procedures that tells the sender how much data it can transmit to the receiver before waiting for the acknowledgement [BAF07]. If the rate of the data absorbed by the receiver is less than the rate of the data transmitted by the sender, flow control mechanism is imposed which basically depends upon the direct point to point feedback to the sender from the receiver [LLP07]. For example, flow control is needed when the powerful computer (sender) is running faster than the low end machine (receiver). In contrast, congestion control is a global issue, involving the behavior of all senders that tend to diminish data sending rate to the network due to the lack of resources at some points of the network. In practice, some algorithms use flow control mechanisms for congestion control of the network.

1.3.5 Basic Principle of Congestion Control

Depending upon the mode of operations, congestion control techniques are broadly classified into two categories, viz.,

- **Open Loop or Preventive Congestion Control** [BFA07]: It is the proactive congestion avoidance tools in which the system controls the network parameters before the occurrence of congestion. Once the system starts to run, midcourse corrections are not made. The open loop congestion control algorithms are divided into two sub classes.

    - ❑ Open loop congestion control, handled by the source.
    - ❑ Open loop congestion control, handled by the destination.

- **Closed Loop Congestion Control** [BFA07]: It is reactive congestion control mechanism in which the system controls the network parameters after realizing congestion. Closed loop solutions are based on the concept of a feedback loop. The closed loop congestion algorithms are also divided into two subcategories [BFA07], as given below :

    - Explicit feedback algorithms**,** in which packets are sent from the point of congestion for warning the source.
    - Implicit feedback algorithms**,** in which the existence of congestion is realized by the source by making local observations, such as the time required to receive the acknowledgement.

In general, closed loop congestion control has three broad steps [AST89], as given below:

- Monitor the system to detect the congestion (Congestion detection).
- Transmit this information to the places where action can be taken (Congestion notification).
- Adjust system operation to correct the problem (Rate adjustment).

The parameters commonly used to monitor congestion in the network are given by:

- Percentage of discarded packets for lack of buffer space.
- Average queue lengths.
- Average packet latency.
- Number of packets time out and re-transmit.

If the number in all the above cases increases, congestion increases.

1.3.6 Types of Congestion in Wireless Sensor Networks

Concerning the reason of occurrence, the congestion in Wireless Sensor Networks is categorized into two broad types:

- **Link Level Congestion (LLC)** –At Wireless Sensor Networks, wireless channels are shared by various sensor nodes that have employed Carrier Sense Multiple Access (CSMA) protocols for getting access to the wireless medium. When several active nodes within the range of one another try to access the wireless channels at the same time, data packet collisions may occur. This type of congestion is called Link Level Congestion (LLC) or the Channel Congestion (CC) [CSV13]. It is also referred to "Type H1 congestion". It increases packet service time and decreases link utilization and overall network throughput, which in turn causes wastage of energy of the sensor nodes. The explicit local synchronization among neighboring nodes can reduce this type of congestion, but cannot eliminate it completely because non-neighboring nodes can still interfere with transmission [CSV13].

- **Node Level Congestion (NLC)** – When the packet arrival rate is more than the packet service time, the buffer space of the individual sensor nodes overflow due to the higher queuing delays. The resulting effect is Node Level Congestion. Since it is induced by the overflow of the buffer of the sensor nodes, it is alternatively known as the Buffer Congestion. NLC is also referred to "Type H2 congestion" [CSV13]. The consequence effect of NLC is the packet loss and increased latency. The re-transmission attempts are thereby increased that may led to the consumption of additional energy and decrease of the network lifetime.

1.3.7 Challenges for Congestion Control in Wireless Sensor Networks

The key distinguishing aspects of the congestion control in resource constraint WSNs, compared to the conventional congestion control protocols for data packet routing are characterized by the following major issues:

- Limited power and processing capabilities of the sensor nodes.
- Distributed nature of WSN processing.
- Event driven properties of WSN traffic.
- Dynamical change of WSN topology due to the node transitioning from sleep mode to active mode and vice versa.

Since the network congestion has direct impact on energy efficiency and QoS, the congestion in WSNs must be controlled, either to avoid it or to mitigate it. So, there is a need of a comprehensive set of congestion control mechanisms specially designed to best fit the unique constrains and requirements of sensor networks and their emerging applications.

1.3.8 Concept of Trust

The concept of trust is borrowed from the human society and inherent meaning is not much different from its literary meaning [ARM11]. In our everyday life, interactions between persons depend upon some kind of trust relationships between them. In general, trust in WSN is the directional relationship between two nodes which shows the degree of trustworthiness in forwarding packets [MOM08]. Technically, it is a mathematical tool representing the degree of reliability or the level of confidence of one node on the other in performing network related activities [MOM08], [TZH10]. The main idea behind the trust based system is to identify and exclude misbehaving nodes, termed as the malicious nodes, in order to minimise the damage caused by the inside attackers. The nodes having trust values higher than a predefined threshold are called trusted nodes or benevolent nodes which perform the normal activities of the sensor network. The trust value of one node with respect to its neighboring node within the radio range is calculated dynamically, by observing the behavior of the node during the previous data transfer through this node and the recommendation received from the other trusted nodes, on the basis of some node characteristics, known as **Trust Metric (TM)** [SSB11], [SAM11]. Depending upon the application of the sensor networks, different TMs are used for the calculation of trust values. Example of some commonly used TMs are given as, number of data packet forwarded, number of control packet (message) forwarded, data packet precision, control packet (message) precision, availability based on beacon / hello messages, consistency of reported (sensed) values / data, reputation, packet address modified, remaining battery life, delay in packet transmission and so on. Trust is broadly classified into two types **Direct Trust (DT)** and **Indirect Trust (IT)** [MOM08], [SSB11], [SAM11], [TZH10]. The parameter **Total Trust (TT)** is a function of DT and IT. Depending upon the nature of application, sometimes DT is given more importance than IT and vice versa.

- **Direct Trust (DT) -** The trust evaluated from the direct experiences of a node on the other node in the neighborhood is called Direct Trust.
- **Indirect Trust (IT) –** Indirect Trust of a node with respect to another node is evaluated from the information or the recommendation of the later one as given by other neighbors. It depends upon the indirect experience of the node on the other node.

Security is different from trust, although in some cases these two terms are used interchangeably. In security protocols, no one is considered as trusted node and authentication is required before every actions. On the other hand, the concept of trust believes everyone is trusted somehow and so authentication is not required [MOM08]. This makes the trust based protocols less complex and light weight as compared to the security algorithms and traditional cryptographic solutions [MOM08], [ARM11].

1.3.9 Trust Based Congestion Control in Wireless Sensor Networks

Trust based congestion control is the new research domain, where the concept of trust is used as a tool to solve the congestion problems in WSNs. The faulty malicious nodes are detected from their trust values. Next, they are blocked, so that they could not take part in further data packet routing activities. As a result, the congestion obtained from the faulty malicious nodes is eliminated, which would expect significant reduction in overall network congestion.

1.4 Justification of the Work

Congestion control and avoidance is one of the major research fields in resource restricted WSNs, because congestion deteriorates network performance and increases energy consumption. Integration of trust and congestion control with data routing algorithms would provide the energy efficient solutions of the problem discussed. Under the light of this, study and research of the congestion control using trust as a tool is important which is largely open and needs to describe elaborately. The present thesis aims to design and implement new protocols for trust based congestion aware energy efficient data routing in WSNs and to compare the performance of trust based

congestion control with other existing congestion control algorithms in terms of network lifetime and throughput.

1.5 Contribution

The contribution of the thesis includes the following:

- A concise survey of the existing congestion control algorithms for WSNs.
- A review of the security attacks and concept of trust management applicable in WSNs.
- Literature survey for the existing trust based congestion control algorithms and its implications in network lifetime and throughput.
- Proposal of two new trust integrated data routing algorithms (ITLSRP and FTRSP).
- Proposal of genetic algorithm inspired load balancing protocol (GACCTR) for congestion control in WSNs using trust based routing framework.
- Proposal of two new fuzzy related trust based congestion control algorithms (TFCC and TCEER) for Wireless Multimedia Sensor Networks.
- Proposal of trust based congestion aware hybrid ant colony optimization model (TC-ACO) for WSNs.
- Proposal of CET-PS algorithm, a congestion aware protocol suite for energy efficient routing in WSNs using trust based framework.
- Proposal of a Fuzzy Coprocessor and its FPGA implementation for trust based congestion aware data routing in WSNs.
- The performance of the above mentioned proposed models are compared with other similar existing protocols to show the merits of the proposed schemes.

1.6 Organization of the Thesis

This chapter has briefly introduced WSNs and the challenges associated with their deployment in unattended and harsh environment. The typical congestion related problems occurred in sensor networks are discussed and the relation of the congestion with the security attacks are explained. The concept of trust is discussed in brief and the needs for the trust based congestion control approaches to solve these issues are

explained. This chapter also includes motivation and justification of the present research and the contributions of the thesis.

The rest of the thesis is organized in the following way:

Chapter 2 presents literature survey which includes overview of security attacks in WSNs, different approaches for trust calculations and overview of the existing trust based routing protocols. An extensive study of the existing congestion control algorithms for WSNs is also included in this chapter.

Chapter 3 presents two new trust integrated data routing algorithms ITLSRP and FTSRP respectively where Link State Routing Protocol (LSRP) is used as the basic data routing scheme. In the proposed ITLSRP algorithm, geometrical mean based indirect trust evaluation mechanism is considered for the calculation of trust of the individual sensor nodes. On the other hand, a new fuzzy logic based trust evaluation model is proposed in FTSRP protocol. The algorithms for the selection of the best trustworthy route are proposed and the performance of the proposed schemes are compared with the similar existing trust integrated routing protocols to show their advantages over them.

Chapter 4 proposes a new congestion control protocol for balanced distribution of traffic among the different paths existing between the source node and the sink node in accordance to the different route trust values. This probabilistic method of data transmission through the various alternate routes can be appropriately modeled with the help of Genetic Algorithms. The proposed GACCTR protocol is mainly targeted in selecting the trustworthy routes and it prevents concentration of the entire data traffic through a single route eliminating any possible occurrence of bottleneck. The merits of the GACCTR protocol in comparison to the existing routing protocols are justified through the simulation results.

Chapter 5 includes two new fuzzy algorithms (TFCC and TCEER) for trust based congestion control in Wireless Multimedia Sensor Networks where the faulty nodes are identified and blocked from the data packet routing by using the concept of trust. In TFCC algorithm, TMs of all the nodes are derived by using a two-stage Fuzzy inferencing scheme. The congestion of the sensor node is controlled by regulating the

rate of the traffic flow on the basis of the priority of the traffic. In TCEER algorithm, the parameter Node Potential is computed on the basis of the trust value, congestion status, residual energy and the distance of the node from the base station using Fuzzy Logic. The source node selects node with highest potential in its one hop radio range for data transmission which is light weight as well as energy efficient. The merits of the proposed TFCC and TCEER algorithms are discussed by comparing with existing protocols.

Chapter 6 presents TC-ACO algorithm where we have proposed a congestion aware, energy efficient, data routing approach by utilizing Ant Colony Optimization techniques. The concept of trust is used to detect the faulty nodes, which are then isolated from the data routing path. The merits of the proposed TC-ACO scheme are verified through simulations and they are compared with other similar protocols.

Chapter 7 presents the algorithm in which three of our previously proposed trust based congestion aware data routing algorithms (TFCC, TCEER and TC-ACO) are integrated into a single protocol suite (CET-PS), where the routing path is selected adaptively on the basis of the congestion status of the sensor node and the parameter called Composite Protocol Efficiency (CPE). The simulation results of the proposed CET-PS algorithm are presented to demonstrate the effectiveness of the protocol compared to the standalone mode implementation of the individual protocols.

Chapter 8 proposes a new trust based congestion control algorithm (TCR-FC), where dedicated state-of-the-art Fuzzy Co-processor (FCP) architecture is designed for fuzzy calculations. It is implemented in Xilinx Spartan 3 Field Programmable Gate Array.

Chapter 9 concludes the thesis by summarizing the overall findings and also by suggesting the future direction of research in this area.

CHAPTER 2

Literature Survey

2.1 Introduction

This chapter introduces a survey of congestion control mechanisms and deployment of trust management in Wireless Sensor Networks. An intensive study of the existing trust based congestion control related to the present research topic is reviewed.

The rest of this chapter is organized as follows: in Section 2.2 the overview of the commonly occurred security attacks in Wireless Sensor Networks is presented, Section 2.3 includes different approaches of trust evaluation methods, in Section 2.4 the existing trust based routing algorithms is described. The existing congestion control mechanisms available in literature are discussed in Section 2.5 and finally, the summary of the chapter is given in Section 2.6.

2.2 Overview of Security Attacks in WSNs

Network security is one of the most pressing issues in all wireless networks including Wireless Sensor Networks. A security attack is defined as an attempt to get unauthorized access to a service or information. It is often impossible to prevent the sensor nodes from being physically accessed by attackers. This is called **node capture**. Once a node is compromised, the attacker is capable of stealing the key materials contained within that node. The normal nodes are called **benevolent nodes** or **legitimate nodes**. The originator of an attack is called **intruder** or **adversary** or simply **attacker**. The

limitation or weakness of a system that could be exploited by the attackers is known as **vulnerability**. The compromised nodes are called the **malicious nodes** or **selfish nodes**. The goal of the security services in WSN is to protect the information and resources from the security attacks. It works in three steps as given below:

- Prevention of attack means the attack will fail,
- Detection of attack and report, so that action can be taken, and
- Recovery means stop an attack and repair the damage.

The security requirements in WSNs include:

- **Availability**, which ensures that the desired network services are available even in the presence of denial-of-service attacks.
- **Authorization,** which ensures that only authorized sensor can be involved in providing information to network services.
- **Authentication,** which ensures that the communication from one node to another node is genuine.
- **Data Confidentiality,** which ensures that a given message cannot be understood by anyone other than the desired recipients. It is one of the most basic security primitives used in almost every security protocol.
- **Data Integrity,** which ensures that the message transmitting from one node to another node is not modified by an adversary. Unauthorized modification of data is not allowed.
- **Data Freshness,** which implies that the data is recent and ensures that no adversary could replay old messages.

Some commonly occurred security attacks and their existing solutions in Wireless Sensor Networks are described in [TKD10], [CKD03], [JSE09] and [APJ04]. The attackers may be categorized as,

- Mote class attackers, and
- Laptop class attackers.

The **mote class attackers** have same resources with all other nodes in the network and it has access to few sensor nodes. The **laptop class attackers** may have access to more powerful devices like laptop or workstation with more battery power, more

computational capacity and memory, high bandwidth, high power radio transmitter and receiver and sensitive antenna. The degree of harm by an attacker with laptop class devices may be several times greater than that of mote class attackers.
In other way, the attackers are also classified as

- Inside attacker, and
- Outside attacker.

The **outside attackers** have unauthorized access to the network. In contrast, **inside attacker** is a node having authorized access to the network and has somehow, gone bad by malicious node or adversaries.

The Security attacks are broadly classified as:

- Passive attack, and
- Active attack.

Passive attack – Monitoring and listening of the unauthorized outside attackers are known as Passive attack. Some common passive attacks on sensor network are as described below:

- **Passive information gathering** - An intruder with powerful receiver and properly designed antenna may pick up the data stream from the network.
- **Monitoring and Eavesdropping** – It is one of the most common attack to the privacy. The adversary could easily discover the communication contents by snooping to the data.
- **Traffic Analysis** – By analyzing traffic pattern, adversary could guess the communication contents.
- **Camouflage Adversaries** – someone may introduce malicious nodes which are in hide within the sensor networks. Later these nodes behave as a normal node and misroute the packets.

Active Attacks – When attackers modify the data stream in the communication channel, it is known as Active attacks. Various types of Denial of Service (DoS) attacks may be possible in different layers, which are briefly described below.
Physical layer attacks – Physical attack may destroy the node permanently. Jamming attack and tampering attack are two common attacks in physical layer.

- **Jamming attack** – This is one of the DoS attack in which adversary interferes with the radio frequencies that are used by the nodes. The intermittent jamming of the network causes delay in data transmission. Even the entire network may be disrupted by powerful jamming attacker.
- **Tampering**- The attacker can extract the cryptographic key from the captured node, tamper with the associated circuitry, modify the program and even replace them with a malicious sensor under the control of the attacker. Since in many applications, sensor nodes are deployed in outside environment at unattended condition, tampering could also destroy the node permanently.

Link Layer (MAC) attack – At link layer, DoS attacks could be continuous channel access, exhaustion or collision between network packets.

- **Continuous channel access / exhaustion / collision** – a collision occurs when two nodes attempt to transmit on the same frequency simultaneously. In this attack, malicious nodes continuously transmit a large number of RTS (Request To Send) packets or ACK (Acknowledgement) control message over the network. It leads into multiple collisions of the packets causing resource exhaustion and draining out of power.

Network layer attacks/ routing attacks – The routing protocols for WSNs are prone to different security attacks in network layer as described below:

- **Spoofed, altered and replayed routing information** - this attack targets the routing information exchanged between the nodes. Adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error message, and increase end to end latency.
- **Selective forwarding attack** – malicious nodes may refuse to forward certain messages selectively and drop the packets.
- **Sybil attack** - in the network generally nodes have unique identity. In sybil attack, adversary can exists with multiple identities at different places of the network at the same time by creating fake identities.
- **Sinkhole attack** – in this type of attack, adversary tries to attract all the traffic from a particular area routed through a compromised node which creates a sinkhole.

- **Wormhole attack** – attackers record the packets at one location in the network and tunnels those to another location. An adversary could convince nodes that they are only one or two hops away from the base station via wormhole adversary although the nodes are multiple hops from the base station.
- **Hello flood attack** – In WSNs, broadcasting of Hello packets are used in routing protocols for finding neighbors within radio range of the sender. Hello flood attacker is basically laptop class attacker which sends Hello packets to every node in the network so that all nodes will respond to the hello message and waste their energy. The victim nodes think that attacker is within its one hop radio communication range and try to send information to the base station through the attacker.
- **Node replication attack** – the attacker captures a node from the network, generates clone of it and then mounts all these nodes within the network with similar identity. So different nodes exist in the network with same identity.

Transport layer attacks- the attack in the transport layer are flooding attack and de synchronization attack.

- **Flooding attack** – the attacker may repeatedly make new connection request until the resource required by each connection are exhausted or reach a maximum limit.
- **De synchronization** refers to the disruption of an existing connection. An attacker may, for example, repeatedly spoof messages to an end host, causing that host to request the retransmission of missed frames.

Different types of security attacks which are very common in Wireless Sensor Networks are illustrated in Fig. 2.1.

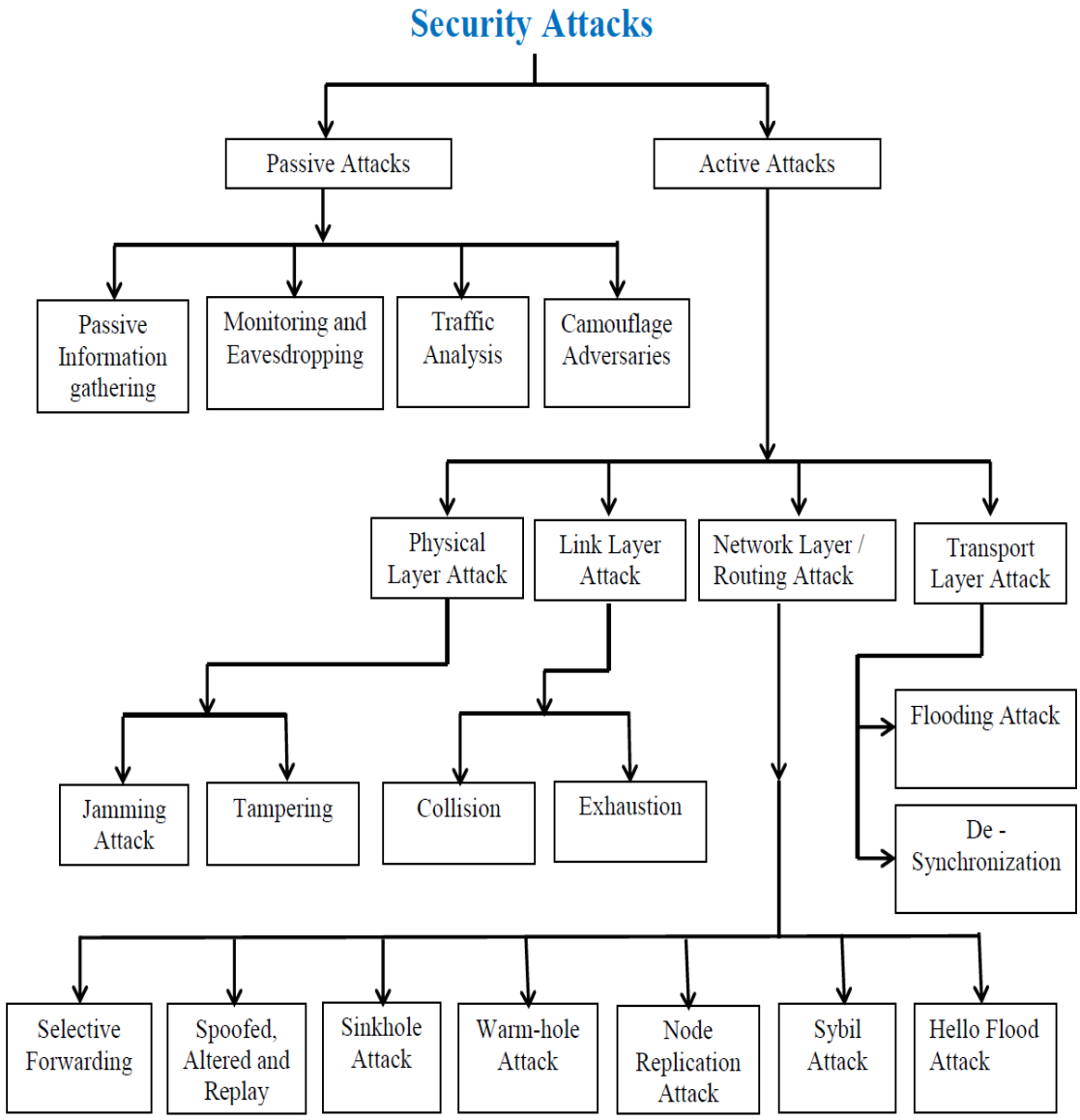


Figure 2.1: Security attacks in Wireless Sensor Networks.

Some security attacks have direct impact in network congestion. For example, Hello Flood attack, Jamming attack, Sybil attack and Node Replication attack aggravate network congestion by flooding the network with fake messages, jamming intermittently, re transmitting same message several times and creating false node identification respectively. It causes additional computation and communication overhead and the resulting effect is increase in battery consumption and decrease in network lifetime.

2.3 Different Approaches of Trust Calculations

Unfortunately, traditional cryptographic security protocols are not suitable for resource constrained Wireless Sensor Networks as they require large overhead of memory, high processing capabilities of sensor nodes and communication bandwidth. Moreover, cryptographic security algorithms cannot detect physically captured nodes, malicious nodes and communication failure nodes. Also, adversary force overtaking is not detected by the cryptographic methods. Trust management is the alternative security solution, suitable for Wireless Sensor Networks which is light weight and can be implemented for the detection of the malicious nodes. Different approaches for trust calculations and their pros and cons are studied in the literature. The most important models that are commonly used are given below.

2.3.1 Momani's Trust Computation Model

Total Trust calculation with traditional weighting approach is used in Momani's model [MOM08], where A is considered as the Direct Trust (experience), B is the Indirect Trust (recommendation) and C is the Total Trust. The Direct Trust A of node N_1 on node N_2 is given by the equation

$$A = \sum_{i=1}^{m} W_i * T_{N_{1_i}}(N_2). \quad \ldots . (2.1)$$

It is the sum of the trust values of N_1 on N_2 for different Trust Metrics (TMs) with different weights (W_i).

The Indirect Trust B of node N_1 on N_2 is the average of Direct Trust of k neighbors on N_2 and is given by

$$B = \frac{1}{k} * \sum_{i=1}^{k} T_i(N_2) \quad \ldots . (2.2)$$

$$total\ trust\ C = F(A, B)$$
$$C = A * W_A + B * W_B \quad \ldots . (2.3)$$

The weights W_A and W_B are chosen as per the application of the sensor nodes. Sometimes Direct Trust is given more importance than the Indirect Trust and vice versa.

2.3.2 Geometric Mean Based Trust Computation Model

In geometric mean based trust computation model (GMTMS) [SSB11], Direct Trust is calculated on the basis of the geometric mean of the Quality of Services or Trust Metrics of the nodes. It is represented in the following equation.

$$DT_{N_1}(N_2) = \left[\prod\nolimits_K (m_{N_1, N_2, K})\right]^{\frac{1}{K}} \quad \ldots . (2.4)$$

Here, $m_1, m_2, m_3 \ldots\ldots m_{10}$ are the TMs of node. The $DT_{N1}(N_2)$ in the above equation is the DT value of node N_1 on N_2, calculated for k different types of Trust Metrics.

GMTMS [SSB11] has certain advantages over the other models. In Momani's model [MOM08], if one of the TM values for data packet transmission is zero and the rest of the TMs have high values, the overall trust value of the node may be above the trust threshold. In this case the node appears to be trustworthy, which is not correct. The above difficulties can be avoided with geometric mean based calculations.

2.3.3 Probabilistic Trust Management Architecture

Probabilistic trust management architecture is proposed in [MKD09], where the history of interactions between nodes, i.e. record of interactions is embedded in a node A about node B. This history entry is defined by the evaluation of interaction, type of interaction and the time in which interaction has happened. Trust computation is done before each interaction between nodes. The trust computation module selects the desired entry in the history of interactions module, then decides whether to pursue with direct or indirect computation to evaluate trust values. This decision is made by computing a certain level of confidence that one node has in the trust evaluation on another node. If the confidence level is low, then the recommendations obtained from other nodes will be added for the computation of trust.

2.3.4 Behavior Based Trust Calculation

The behavior based trust calculation presented in [HTS06] is a variant of reputation-based trust in e-Commerce [LXL03] to reduce the risk of dishonest customer or vendor. The trust degree is the core of behavior based trust frame work. After every

task, the behavior of the node is evaluated in routing, data processing and in other tasks. The evaluation result is combined with old trust degree to form a new one. The behavior evaluation methods are task-specific, and differ from each other. To a given task, the evaluation result may be accurate or inaccurate.

2.3.5 Fuzzy Logic Based Trust Model

Fuzzy logic based trust model is proposed in [TKM08], in which trust level of sensor node, trustworthiness T and untrustworthiness U are calculated assuming that the wireless sensor network has the reputation value of each sensor node. In reputation components, minimum T, maximum T, minimum U and maximum U are defined between each pair of nodes. From this trust T and un-trust U are acquired. Using that T and U, the evaluation level of the network is acquired (T/ (T+U)).

2.4 Overview of Trust Based Routing Protocols

2.4.1 ARIADNE

ARIADNE [YCH00] is a secure on demand routing protocol for ad hoc networks. It uses highly efficient symmetric cryptographic primitives and per-hop hashing function. It prevents the attackers or compromised nodes from tampering the uncompromised routes consisting of uncompromised nodes, and also prevents different types of Denial-of-Service attacks.

2.4.2 ATSR

Ambient Trust Sensor Routing (ATSR) is a fully distributed Trust Management System [TZH10], where the reliability of the node is evaluated using the concept of trust. In this approach, nodes monitor the behavior of their neighbors with respect to different trust metrics and calculate Direct Trust value of each neighbor. It also takes Indirect Trust information, i.e., information from its neighbors called Reputation. Total Trust is calculated by combining Direct and Indirect Trust information. Finally, the routing decisions are taken on the basis of the geographical information (distance of the node from the base-station) and Total Trust information. Disadvantage of this approach is that nodes need to calculate the DTs and ITs for all other nodes, and it requires high processing and storage capabilities.

2.4.3 TAODV

Trusted Ad hoc On Demand Distance Vector (TAODV) is an extended AODV routing protocol where data routing is obtained by considering the Trust Metrics into account [XLL04]. First, a trust recommendation mechanism is introduced and then the routing decision rules of AODV are modified to take trust into account. It has the disadvantage that the entire network architecture would become complicated due to trust information exchange mechanism.

2.4.4 Trusted GPRS

Trusted GPSR is the modified Greedy Perimeter Stateless Routing protocol [AAP07], where the trust levels of the nodes are integrated in the data routing algorithm. Each time a node sends out a packet it waits, until it overhears its neighboring node forwarding the packet to the next node. The node maintains the trust value of its neighbor on the basis of the correct and prompt forwarding information. This information is then taken into account in the routing decisions. It has the disadvantage that it increases transmission latency of the system.

2.4.5 SPINS

Security Protocols for Sensor Networks (SPINS) [APR01] provides mechanism for secure communication suitable for WSNs. It consists of two building blocks, namely, SNEP (Secure Network Encryption Protocol) and µTESLA (micro version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol). The µTESLA protocol provides authenticated data broadcasting and the SNEP provides one-to-one data confidentiality, data authentication and data freshness with an overhead of 8 bytes. It has the disadvantage that architecture does not address the problem of compromised sensor nodes.

2.4.6 TDSR

Trust-aware Dynamic Source Routing (TDSR) [SMT00] algorithm consists of two modules, viz., the Watchdog module and the Pathrater module which are incorporated in the Dynamic Source Routing protocol for security. The Watchdog module is responsible for detecting selfish nodes that do not forward packets. For this, each node in the network buffers every transmitted packet for a limited period. During this period, the node enters into promiscuous mode in order to overhear whether the next node has forwarded the packet or not. On receiving feedback from the Watchdog module, Pathrater module assigns different ratings to the nodes. These ratings are then used to select the routes consisting of nodes with the highest forwarding rate. The disadvantage of the algorithm is that it is complex and it increases the transmission latency of the system.

2.4.7 TARF

Trust Aware Routing Framework (TARF) for Wireless Sensor Networks algorithm [GZW02] integrates trustworthiness and energy efficiency in making the routing decisions. In order to route a data packet to the Base Station (BS), the node needs to select the neighboring node to which it should forward the data packet. In addition to data packet transmission, other two types of routing information need to be exchanged are broadcast message from the BS about undelivered data packets and energy cost report message from each node. Each node maintains a neighborhood table with trust values and energy cost values for certain known neighbors. Energy watcher and Trust Manager are the two components that run on each node. In TARF, a node keeps track of the trustworthiness of its neighbor and thus selects a trusted route. The disadvantage of the algorithm is that it is complex and it increases the transmission latency of the system.

2.4.8 CONFIDANT

Cooperation Of Nodes, Fairness In Dynamic Ad-hoc Networks (CONFIDANT) [TZH09] algorithm provides two new building blocks, reputation system and a trust manager to the Watchdog and Pathrater scheme. The trust manager evaluates the events reported by the Watchdog module and issues signals to the other nodes regarding malicious nodes. The signal recipients are maintained in a friends-list.

The reputation system maintains a black-list of the nodes and shares it with the friends-list nodes. It is a punishment based scheme by not forwarding packets to the nodes whose trust level drops below the certain threshold level. The disadvantage of this algorithm is that it is complex with high overhead and latency.

2.4.9 TRANS

Trust Routing for Location Aware Sensor Networks (TRANS) [STP04] is a location centric architecture for trust routing in WSNs. The two main components of TRANS are given as TRM (Trust Routing Module) and ILAM (Insecure Location Avoidance Module) respectively. The TRM is installed in the sink and all the sensor nodes whereas ILAM is installed in the sink only. The routes are selected on the basis of the trust information, not on the hop count to avoid insecure locations. The sink sends message only to the trusted neighbors for the destined location. The corresponding neighbors forward the packet to their trusted neighbors that have the nearest location to the destination. Thus, the packet reaches the destination through the trusted sensor nodes. The disadvantage of this protocol is that it is complex and needs extra component or hardware.

2.4.10 DTLSRP

Direct Trust-dependent Link State Routing Protocol (DTLSRP) [SAM11] computes Total Direct Trust (TDT) of the sensor nodes by geometric mean of Trust Metrics under consideration. A predefined trust threshold value (TTH) is considered on the basis of the application of the WSNs. The nodes having TDT higher than TTH are considered as benevolent nodes and the rest are selfish nodes. After selecting benevolent nodes, Link State Routing protocol is used to find all the available paths from the source node to the sink. The Route Trust (RT) is calculated for each path. The routing path with highest RT is chosen as best routing path. DTLSRP simulations perform better result with respect to the other protocols such as ATSR algorithm [TZH10], CONFIDANT algorithm [TZH09] etc. This method is allowed to find the shortest path without applying Dijskra's algorithm. However it does not include Indirect Trust (IT).

2.5 Overview of Congestion Control Algorithms

The performance of Wireless Sensor Networks is highly dependent on the network congestion, and therefore, the congestion control techniques are needed for achieving high throughput and long network lifetime. Despite of the adequate number of congestion control protocols proposed in the literature, effective congestion control in Wireless Sensor Networks still remains challenging. The study and review of the existing congestion control algorithms are presented in [MAK14], [JZL10], [RCC10], [ALG15] and [CHS12]. In the following sections, the brief discussions of some important congestion control and congestion avoidance protocols related to this research work have been included.

2.5.1 CODA

Congestion Detection and Avoidance (CODA) in sensor networks is energy efficient congestion control algorithm proposed by C. Y. Wan et al. [CYW03], which comprises of three key mechanisms as described below:

- Congestion detection by monitoring channel loading conditions and current buffer occupancy at each receiver nodes. Since listening to the channel at all the time incurs high energy costs, CODA [CYW03] deploys a sampling scheme that activates local channel monitoring at the appropriate time.

- Open loop hop by hop back pressure - Once the congestion is detected, open loop hop by hop back pressure message is generated and broadcasted to all one hop downstream nodes. When a node receives the back pressure message, it decides whether or not to further propagate the back pressure message depending on its own local network conditions. On receiving back pressure signals, nodes can throttle their sending rates or drop packets on the basis of the local congestion policy. The term depth of congestion is used to indicate the number of hops that the backpressure message has traversed before a non-congestion node is encountered.

- Closed loop multisource regulation – In case of persistence congestion, closed loop multisource regulation acts over the multiple sources to a single sink. If the

source event rate is less than the maximum theoretical throughput of the channel, the source regulates itself. When this value is exceeded, the source is more likely to contribute to the congestion and the closed loop congestion control is activated. In that case, the source triggers sink regulations by setting a regulate bit in the event packets, which is forwarded towards the sink. Reception of packets with the regulate bit set force the sink to send ACK message to all the sources associated with a particular data events. As long as there is no failure to receive ACK at the sources, they can maintain their own rates, otherwise rate reduction at the sources are forced to control congestion.

Limitations:

- Though CODA can reduce congestion in WSNs to a great extent, it does not eliminate congestion.
- CODA does not put enough emphasis on fairness.
- The role of the malicious nodes in network congestion is not discussed.

2.5.2 ESRT

In Event–to-Sink Reliable Transport **(**ESRT) algorithm [YSO03], sensor nodes change their sending rate on the basis of the feedback from the sink in terms of reliability level or congestion detection. Each node sets a Congestion Notification (CN) bit in the packets as soon as the buffer reaches a threshold level. The sink periodically calculates the new reporting rate for all the sources on the basis of the reliability measurement, CN bit and the old reporting rate. It is then broadcasted by the sink. The primary objective of ESRT algorithm is to configure the network so that it operates as close as Optimal Operating Region (OOR).

Limitations:

- In ESRT, reliability is defined as the number of received packets. But in actual cases, malicious nodes may corrupt the packets during transition or they may send additional fake messages. In WSNs, low cost sensor nodes are prone to failure and often behave as the malicious nodes. But, in ESRT, the role of the malicious nodes is not discussed.

- There is no congestion control mechanism at the intermediate nodes. Sink is responsible to all rate adjustment in the network.
- The assumption that a sink is capable to broadcast all the source nodes is not reasonable.

The summary of ESRT protocol operations are described in five states as shown in Table 2.1.

Table 2.1: Summary of ESRT protocol operation

Network State	Description	ESRT Action
NC, LR	No Congestion, Low Reliability	Reporting rate is increased to reach an acceptable rate.
NC, HR	No Congestion, High Reliability	Decrease reporting rate conservatively,
C, HR	Congestion, High Reliability	Decrease reporting rate aggressively to state (NC, HR) to relieve congestion.
C, LR	Congestion, Low Reliability	Decrease reporting rate exponentially, Relieve congestion as soon as possible.
OOR	Optimal Operating Region	Reporting rate remains unchanged

2.5.3 Other Important Congestion Control Algorithms

In Pump Slowly Fetch Quickly (PSFQ) algorithm [CYW05], data is distributed from a source node by sending data at a relatively slow speed called “pump slowly” and allowing the nodes that experience data loss to recover missing segments from their local immediate neighbors aggressively called “fetch quickly”. Priority Based Congestion Control (PCCP) algorithm [CWK06] is a hop by hop congestion control algorithm in the upstream direction, in which node priority index is considered and node congestion is measured by using packet inter arrival time and service time. A new Queue-based Congestion Control Protocol with Priority Support (QCCP-PS) is presented in [MHY08], where queue length and priority index are taken as the indication of

congestion degree. Omar Banimelhem et al. [OBS12] have proposed a Grid-based Multipath with Congestion Avoidance Routing (GMCAR) protocol, which is suitable for grided sensor networks. The entire sensor network field is divided into several grids. In each grid, one master node is selected which is responsible for delivering data from any nodes within the grid and routing data from other master nodes in its neighbor grids. In [BHK04], B. Hull et al. have combined three congestion control techniques into a strategy called fusion which improves network efficiency, fairness and channel loss rate. The first technique is hop-by-hop flow control, in which local congestion is reduced via back pressure. The second technique is source rate limiting scheme and third one is the prioritizing MAC layer that gives priority to the backlogged nodes over non-backlogged nodes to access the shared medium. In Hybrid Congestion Control Protocol (HCCP), Jang-Ping Sheu et al. [JPS09], have presented a distributed algorithm that mitigates congestion by allocating appropriate source rate to the sink node. It considers packet delivery rate and buffer size of each node. Kang et al. [JKY07] have proposed the Topology Aware Resource Adaptation (TARA) protocol. It measures the buffer occupancy and channel loading in order to detect congestion. If the congestion level of a node hits the upper limit, it declares congestion and becomes a hot spot node. The hot spot node locates two other low congestion level nodes, called distributor node and merger node respectively. A detour path is established starting at the distributor node and ending at the merger node. The congestion control algorithm adapted in TARA [JKY07] requires knowledge about the whole network, which makes the protocol impractical for large scale network. A fuzzy based congestion control for WMSNs is proposed in SUIT algorithm [CSZ14], where some packets of the frames are dropped from the source. S. M. Aghdam et al. [SMA13], have presented WCCP (WMSN Congestion Control Protocol) for Wireless Multimedia Sensor Networks, where congestion is avoided by adjusting the sending rate of the source nodes and by distributing the departing packets from the source nodes. In WCCP [SMA13], the queue length of intermediate nodes are monitored to detect congestion and less important frames are ignored to reduce congestion.

2.5.4 Trust Based Congestion Control Algorithms

Trust based congestion control is the new research domain where trust is used as a tool for reducing congestion, caused by the malicious nodes. A few research efforts have been

done in this domain, but still some challenges are left unaddressed. Zarei et al. [MZA09] have proposed a fuzzy logic based trust estimation scheme for congestion control in WSNs. The algorithm described in [MZA10] is basically a modification of the protocol as presented in [MZA09], where a pre-defined trust threshold value is considered for decision making. Although, it improves network throughput compared to the conventional congestion control algorithms, there is a scope of improvement, because remaining energy of the nodes are not considered in [MZA10]. A trust routing protocol based on congestion control in MANET is proposed in [RRM09], where the load is distributed among the nodes having higher trust.

2.6 Summary

In this chapter, the literature survey for the current solutions on congestion control and avoidance in WSNs are included. The commonly occurred security attacks and different methods of trust calculations are also discussed in brief. It is noticed that majority of the congestion control algorithms employ traffic control for congestion mitigation in WSNs. Few algorithms consider resource control as well. But, most of them do not discuss the impact of the security attacks for aggravation of network congestion. In many critical applications, especially for large Wireless Sensor Networks and Wireless Multimedia Sensor Networks, where large volume of data is needed to transfer through the network, the traffic control and resource control are not sufficient. For that isolation of faulty malicious nodes creating congestion would be a good solution. It is widely open research area and hence, the present research works mainly focus on the design of trust based congestion control in Wireless Sensor Networks.

CHAPTER 3

Trust Integrated Data Routing Algorithms for Wireless Sensor Networks

3.1 Introduction

Wireless Sensor Networks (WSNs) are self-configured ad- hoc networks without centralized administration which consist of spatially distributed, low cost tiny sensor nodes that can monitor physical or environmental conditions at different locations, even in remote areas, by means of wireless radio links. Wireless Sensor Networks can be deployed to support numerous application areas like climatic data gathering, seismic and acoustic underwater monitoring, industrial control and monitoring, intelligent agriculture, area surveillance, indoor security, military applications, health care and many more. Due to its unique characteristics, WSNs are prone to be vulnerable to various types of security attacks. However, one of the fundamental requirements of Wireless Sensor Networks is sensing and routing of message in a secured way, from the source node to the sink or the base station with energy efficiency. Unfortunately, traditional cryptographic security protocols are not suitable for resource constrained Wireless Sensor Networks as they require large overhead of memory, processing speed and communication bandwidth. However, security has a strong relationship with trust and the concept of trust and reputation play an important role to design trustworthy Wireless Sensor Networks with low overhead and processing speed. Although some research papers are available for computing trust

value of the nodes, a very limited works about their inclusion in the data routing path have been reported so far. Some of the related algorithms are proposed in ATSR [TZH10], DTLSRP [SAM11] and TILSRP [ARM11] respectively, which are briefly discussed in Chapter 2 (Literature Survey).

In this chapter, two new trust integrated data routing algorithms are proposed. In the first algorithm (ITLSRP protocol), Indirect Trust (IT) of the individual nodes are estimated with the help of geometrical mean based trust management system and the parameter Route Trust (RT) is calculated. Finally, the best trustworthy route of the network is selected for data transfer. On the other hand, the second algorithm (FTRSP algorithm) is a new trust oriented routing protocol, in which, the trust of the individual sensor nodes in the network are computed using Fuzzy Logic and then the most trusted route is selected by using the proposed route selection protocol.

The rest of the chapter is organized as follows: Section 3.2 briefly describes the Link State Routing Protocol, Section 3.3 represents the proposed ITLSRP algorithm and its simulation results and Section 3.4 explains the proposed FTSRP algorithm with the corresponding simulation results. Finally Section 3.5 summarizes the work.

3.2 Link State Routing Protocol

As described in [LLP07] and [AST89], Link State Routing is one of the most effective intra-domain routing protocol. Nowadays, variants of Link State Routing are used widely in different applications. The idea behind Link State Routing basically consists of five steps as given below.

Step 1: Learning about the neighbors - It is assumed that each node is capable to find out all of its neighbors and learn their network addresses. To achieve this goal, the node sends a special HELLO packet on each point-to-point line. The node on the other end sends back a reply telling its identity that must be globally unique.

Step 2: Measuring line cost - Every node knows how to reach its directly connected neighbors and measure the delay or cost to each of its neighbors. The delay can be determined directly by sending a special ECHO packet over the line that the other side is required to send back immediately. The sending node can estimate the delay by

measuring the round-trip time and divided it by two. The same test can be conducted several times for reducing measurement errors.

Step 3: **Building Link State Packet (LSP)** – the packet called LSP is constructed by the node which contains the following data

- The ID of the node who creates the LSP
- List of directly connected neighbors of that node, with the cost of the link to each one.
- A sequence number
- A time to live for the packet (age)

Step 4: Distributing the Link State Packets - The reliable distribution of LSP is the most crucial part of the algorithm. Reliable flooding is the process of making sure that all the nodes participating in the routing protocol get a copy of LSP from all the other nodes. The basic idea of reliable flooding is that a node sends LSP to all of its directly connected nodes. On receiving LSP, the nodes on the other side send it out to all of their direct links. The process continues until the information has reached all the nodes in the network.

Step 5: Compute the shortest path – The shortest path of a node from another node can be obtained by using Dijkstra's algorithm. In the proposed ITLSRP and FTSRP algorithms, Link State Routing Protocol is used with the exclusion of step 5. Instead of using Dijkstra's algorithm for finding the shortest route, we have implemented our algorithm to get the most trustworthy route as the routing path.

3.3 Proposed Algorithm 1:

A Noble Indirect Trust based Link State Routing Protocol (ITLSRP*)* using a Robust Route Trust Method for Wireless Sensor Networks

In this trust model, every node in the network will monitor all events and record them in the database stored in the memory, in terms of Trust Metric (TM) quantities. Next, the Link State Routing Protocol is applied to find out all possible routes from the source to the sink. Indirect Trust values of the nodes in the routing path are computed thereafter. The parameter called Route Trust (RT) of all routes from the source to the

sink is calculated by using Indirect Trust of the nodes of the corresponding routes. Finally, the route having the highest value of RT is selected as the routing path. The algorithm is explained by considering a particular example. Since the proposed scheme considers only the Indirect Trust of the node, it is light weight and energy efficient as well.

3.3.1 Notion of Direct and Indirect Trust

As described in GMTMS algorithm [SSB11], the concept of Direct Trust (DT) and Indirect Trust (IT) of the sensor nodes are explained pictorially in Figure 3.1. The DT of a node with respect to its one hop neighboring node is computed by analysing the Trust Metric (TM) values obtained during previous data transfer between these two nodes. On the other hand, IT of a node with respect to its one hop neighbouring node is evaluated by analysing the opinion (TM values) of the node, received from other one hop neighboring nodes. So, DT is associated with the direct experience between two nodes whereas IT deals with the indirect information received from other nodes. In Fig. 3.1, the trust value of node I on node J is calculated. The nodes A, B, C, D and E represent one hop neighbouring nodes. The dashed arrow indicates the indirect information of node J given by the neighbors to the node I. The solid arrow indicates the direct experience.

The node I evaluates its DT on node J by the Eq. (3.1),

$$DT^{I,J} = \left[\prod_K \left(m_{I,J,k}\right)\right]^{\frac{1}{K}} \quad \ldots . \text{(3.1)}$$

where, $DT^{I,\,J}$ is the Direct Trust value of node I on node J, calculated for K different Trust Metrics (k = 1 to K), $m_{I,\,J,\,k}$ is k^{th} Trust Metric quantity of node I on node J.

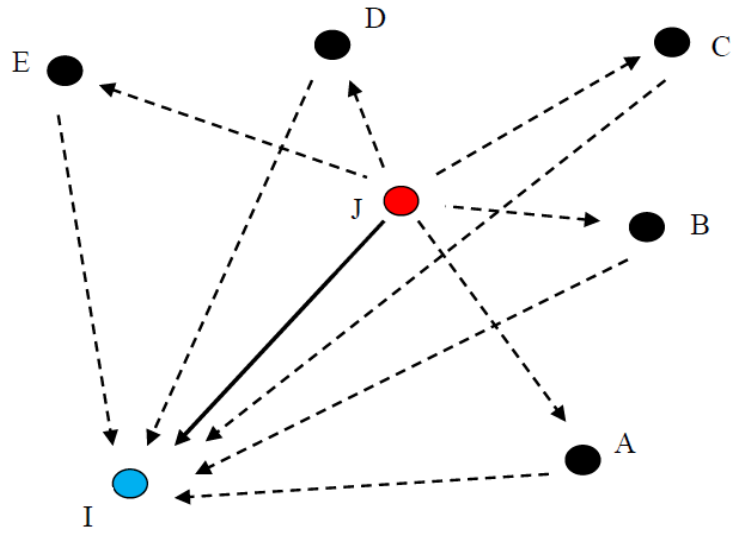


Figure 3.1: Direct Trust (DT) and Indirect Trust (IT) evaluation.

The IT of node I on node J, can be calculated from the DTs on J with respect to its neighbors, sent by the neighboring nodes. The Indirect Trust of node I on node J is defined as the geometric mean of the DTs of neighbour nodes A, B, C, D, and E on J. This is represented by the Eq. (3.2),

$$IT^{I,J} = \left[\prod_L \left(DT^{N_j,J}\right)\right]^{\frac{1}{L}} \quad \ldots . \quad (3.2)$$

$DT^{N_j,J}$ is a set of Direct Trust on node J, given by N_j neighboring nodes (j = 1 to L).

3.3.2 Proposed ITLSRP protocol

The proposed ITLSRP protocol consists of four steps as given below:

Step 1: Consider an arbitrary Wireless Sensor Network. Apply LSRP for a particular source and sink node and get all possible routes.

Step 2: Calculate Indirect Trust values of the nodes comprising the routes with respect to the source node.

Step 3: Evaluate Route Trust (RT) of all routes using IT values of the nodes of the corresponding routes.

Step 4: Select the Most Trusted Route (SMTR) and get the final routing path.

It is assumed that there are total 64 wireless sensor nodes organized as an 8 X 8 grid. In the rest of this section, the color of the nodes will represent their identities except when mentioned directly with the node numbers as caption. For example, a node having the color green will be referred to as a green node. Nodes sharing some common traits have been denoted by the same color. In Fig. 3.2, the green nodes represent all the concerned nodes which are taken into account in the present scenario. Initially, all the nodes are assumed to be homogeneous and uniform in hardware and routing capabilities. However, their parameters may vary randomly with time. Every packet of information to be routed comprises of a source and a sink which are denoted by orange and red colours respectively, as shown in Fig. 3.3. A stream of packets are sent from the source (node number 1: N1, represented by orange color) to the sink (node number 35: N35, represented by red color), as shown in Fig. 3.4. It is already mentioned that the basic routing protocol employed here is the Link State Routing Protocol (LSRP) which finds all possible routes from the source to the sink. The LSRP is particularly attractive in the case of Wireless Sensor Networks due to the limited hardware and software features. The three routes from the source to the sink selected by LSRP are shown in Fig. 3.4. The grids showing nodes within the routes with communication ranges are represented in Fig. 3.5.

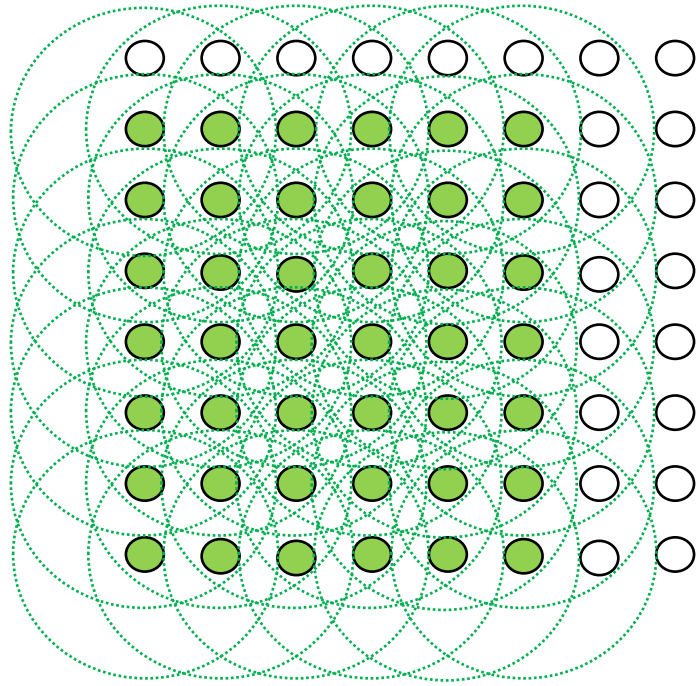


Figure 3.2: Grid (square matrix) network topology of sensor nodes.

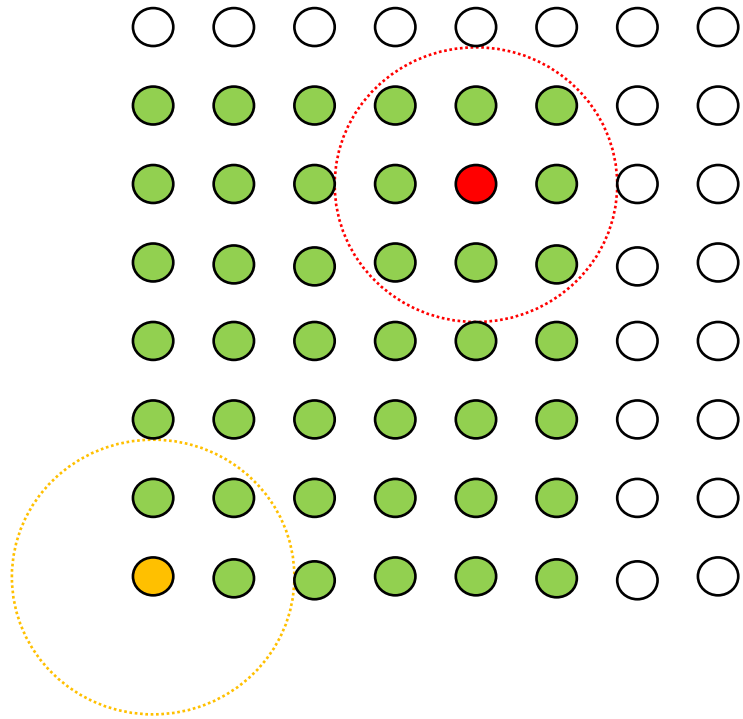


Figure 3.3: Network showing source and sink.

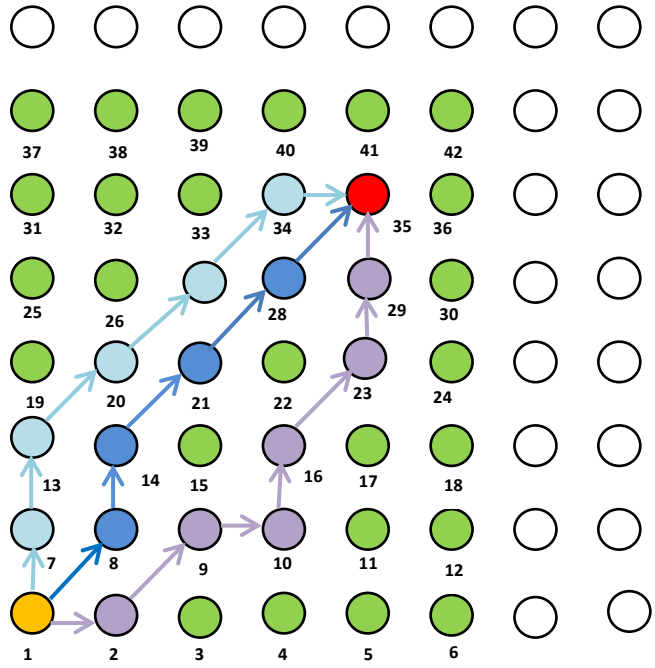


Figure 3.4: Three routes from source to sink selected by LSRP.

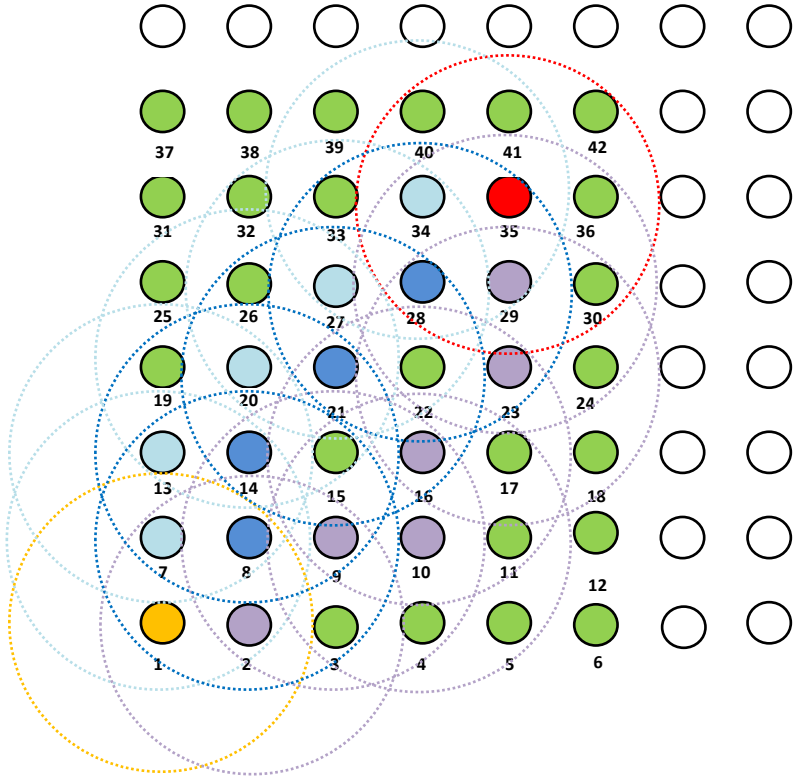


Figure 3.5: Grid showing nodes in the routes with communication ranges.

3.3.2.1 Calculation of Indirect Trust

As discussed in the previous section, three different routes are shown in Fig. 3.4, as obtained by using LSRP. Next step is the calculation of Indirect Trust (IT) of the nodes. It is to be noted here that the Indirect Trust of the node is obtained from the Direct Trust of the node with respect to its one hop neighbors.The one hop neighbour means the cluster of those nodes who can communicate directly with that node or in other words who are within the communication range of that particular node. In the proposed algorithm, communication ranges are assumed to be uniform which are shown by the dotted circles in Figs. 3.2, 3.3, 3.5, 3.6, 3.8 and 3.9 respectively. The three arbitrary scensarios for the calculation of Direct Trusts of nodes with respect to one hop neighbors are shown in Fig. 3.6, which are marked in yellow colour.

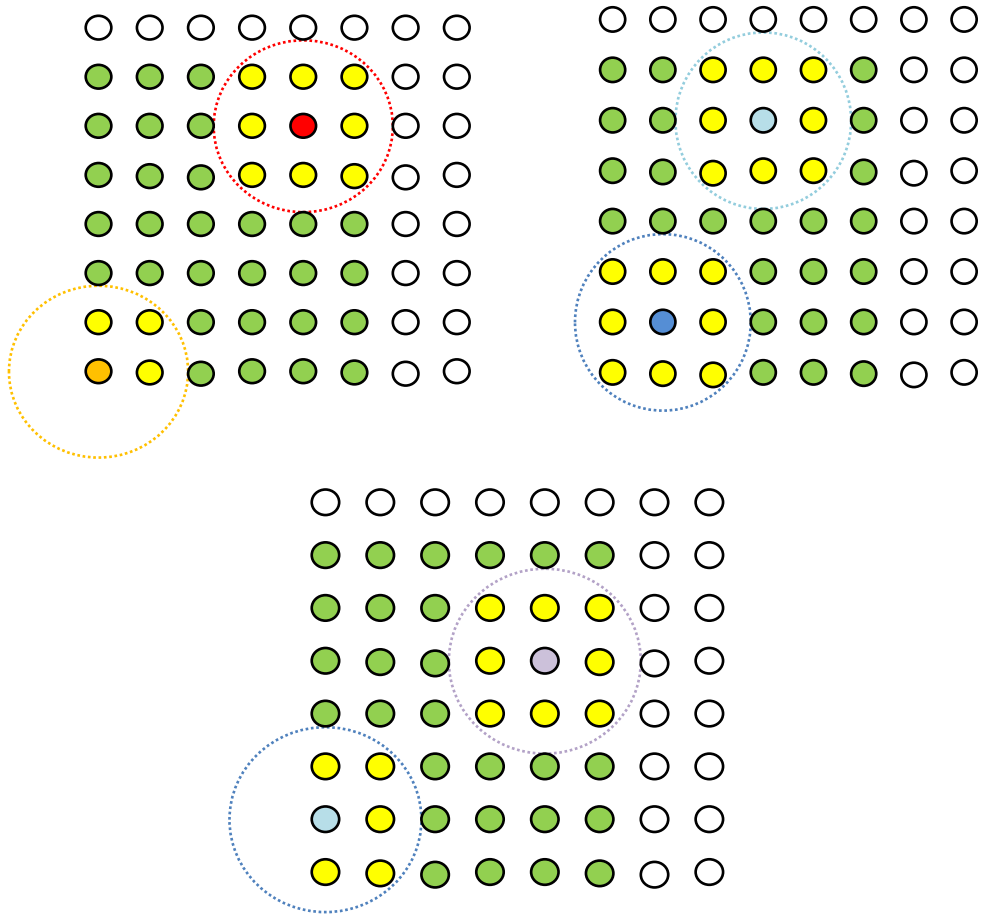


Figure 3.6: Calculation of DT of nodes with respect to one hop neighbors.

The Direct Trust is calculated from the Trust Metric values as per the Eq. (3.1). Next, Indirect Trust values of the nodes with respect to the source node is calculated from the Direct Trust values of other one hop neighboring nodes. The required expression for the calculation of Indirect Trust of an arbitrary node N_X with respect to the source node N_1 is represented by Eq. (3.3),

$$IT_1(X) = \frac{\left[\left\{\frac{1}{N}\sum_{i=1}^{N}\left\{DT_{Ne_{Ri}}(X) \times W_{Ne_{Ri}}\right\}\right\} \times \left\{\left\{\frac{1}{M}\sum_{j=1}^{M}\left\{DT_{Ne_{Sj}}(X) \times W_{Ne_{Sj}}\right\}\right\}\right]^{\frac{1}{2}} + \left\{\left\{\frac{1}{P}\sum_{k=1}^{P}\left\{DT_{Ne_{Ok}}(\mathrm{X}) \times W_{Ne_{Ok}}\right\}\right\}\right\}}{2} \quad \ldots . \quad (3.3)$$

The explanation of the different notations used in Eq. (3.3) is explained in Table 3.1, W_{NeRi}, W_{NeSj} and W_{NeOk} represent the weightage given to the Direct Trust of N_X with respect to i^{th} one hop neighbors $N_{eRi}(X)$, j^{th} one hop neighbors $N_{eSj}(X)$ and k^{th} one hop neighbors $N_{eOk}(X)$ respectively. $N_{eRi}(X)$ and $N_{eSj}(X)$ represent i^{th} one hop neighbor of N_X from whom N_X receives data while forwarding them from source to sink and j^{th} one hop neighbor of N_X to whom N_X sends data while forwarding them from source to sink. The nodes $N_{eRi}(X)$ and $N_{eSj}(X)$ can be involved in more than one route in accordance with the LSRP protocol. $N_{eOk}(X)$ represents k^{th} one hop neighbor of N_X which does not occur within any of the routing paths. The nodes N_X, $N_{eRi}(X)$, $N_{eSj}(X)$ and $N_{eOk}(X)$ are shown pictorially in Fig. 3.7.

Table 3.1: Explanation of Notations

Notation used in Eq. (3.3)	**Explanation**
$\mathbf{Ne_X}$	Total number of one hop neighbors of N_X
$\mathbf{Ne_{Ri}(X)}$ (i varies from 1 to N)	i^{th} one hop neighbor of N_X from whom N_X receives data while forwarding them from source to sink.
$\mathbf{Ne_{Sj}(X)}$ (j varies from 1 to M)	j^{th} one hop neighbor of N_X to whom N_X sends data while forwarding them from source to sink.
N	Total number of neighbor nodes involved in the routing. Hence, $N = Ne_{Ri}(X) + Ne_{Sj}(X)$
$\mathbf{Ne_{Ok}(X)}$ (k varies from 1 to P)	k^{th} one hop neighbor of N_X which does not occur within any of the routing paths. Hence, $Ne_{Ok}(X) = Ne_X - N$
$\mathbf{DT_{NeRi}(X)}$	Direct Trust of N_X with respect to i^{th} one hop neighbors $N_{eRi}(X)$
$\mathbf{DT_{NeSj}(X)}$	Direct Trust of N_X with respect to j^{th} one hop neighbors $N_{eSi}(X)$
$\mathbf{DT_{NeOk}(X)}$	Direct Trust of N_X with respect to k^{th} one hop neighbors $N_{eOk}(X)$
$\mathbf{W_{NeRi}}$	Weightage given to the Direct Trust of N_X with respect to i^{th} one hop neighbors $N_{eRi}(X)$
$\mathbf{W_{NeSj}}$	Weightage given to the Direct Trust of N_X with respect to j^{th} one hop neighbors $N_{eSj}(X)$
$\mathbf{W_{NeOk}}$	Weightage given to the Direct Trust of N_X with respect to k^{th} one hop neighbors $N_{eOk}(X)$

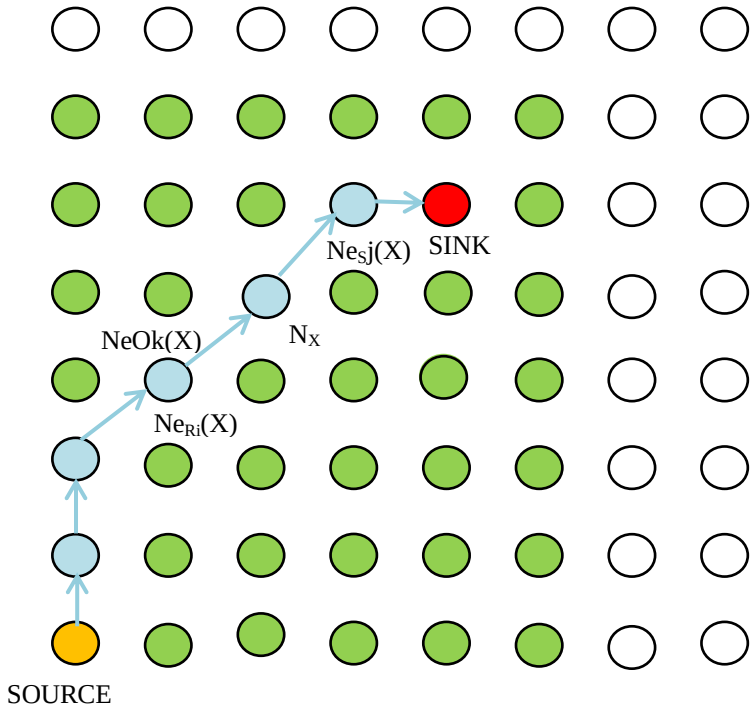


Figure 3.7: Representation of nodes N_X, $N_{eRi}(X)$, $N_{eSj}(X)$ and $N_{eOk}(X)$ in route 3.

3.3.2.2 Assignment of Weightages

The proposed ITLSRP algorithm considers equal weightage on Direct Trust of $N_{eRi}(X)$ and $N_{eSj}(X)$ and hence, $W_{NeRi} = W_{NeSj}$. The assignment of weightages are given as,

$$W_{NeRi} = W_{NeSj} = \frac{\alpha}{N} \text{ and } W_{NeOk} = \frac{\beta}{P} \quad \ldots . (3.4)$$

where, α is the magnitude of total weightage given to the DT of $N_{eRi}(X)$ and $N_{eSj}(X)$, β is the magnitude of total weightage given to the DT of $N_{eOk}(X)$, N is the total number of one hop neighboring nodes involved in the routing and P is the total number of one hop neighboring nodes that are not involved in the routing. Here, for convenience, $\alpha + \beta = 1$. The values of α and β can be regulated, depending on the application of the network. For most of the cases, $\alpha > \beta$. A possible value of α may be 0.75, so, $\beta = (1- 0.75) = 0.25$, then Eq. (3.4) can be written as,

$$W_{Ne_{Ri}} = \frac{3}{4n_X}, W_{Ne_{Sj}} = \frac{3}{4n_X} \text{ and } W_{NeOk} = \frac{1}{4n_X} \quad \ldots . (3.5)$$

3.3.2.3 Calculation of Route Trust

The individual Route Trust value of R routes, derived from LSRP is denoted by RT_R. In the given example, three routes are obtained upon the implementation of LSRP protocol as shown by different coloured nodes in Fig. 3.4. Sensor nodes and their communication ranges in the above mentioned three routes are represented in Fig. 3.8.

Route 1 : N1(Source) → N8→ N14→ N21→ N28→ N35 (Sink),
Route 2 : N1(Source) → N2→ N9→ N10→ N16→ N23→ N29→ N35 (Sink),
Route 3 : N1(Source) → N7→ N13→ N20→ N27→ N34→ N35 (Sink),

The Route Trust (RT) values are calculated by multiplying Indirect Trusts of all nodes (with respect to the source) belonging to the route. In the generic form, let us assume that a series of 10 nodes excluding the source and sink belong to a definite route. Let these nodes be denoted by R_q, where q varies from 1 to 10. Then, Route Trust calculated for this route is given by the equation (3.6) as shown below.

$$RT = \prod_{q=1}^{10} IT_1(R_q) \quad \ldots . (3.6)$$

For the given example, the Route Trusts are evaluated as follows:

$RT_1 = \prod(IT_1(8), IT_1(14), IT_1(21), IT_1(28))$
$RT_2 = \prod(IT_1(2), IT_1(9), IT_1(10), IT_1(16), IT_1(23), IT_1(29))$
$RT_3 = \prod(IT_1(7), IT_1(13), IT_1(20), IT_1(27), IT_1(34))$

The multiplication of Indirect Trust of individual nodes belonging to the route is taken, instead of addition, because it has certain advantage over the addition method. If a certain node of the route has very low Indirect Trust, even zero Indirect Trust, the RT of the route may be high in case of addition method, when the Indirect Trust values of other nodes belonging to the route are high. So, this is the case where we get high RT of the route, although it has some nodes with very low trust values. This problem is not appeared in the proposed model, where multiplication of Indirect Trust is considered for calculation of RT of the route.

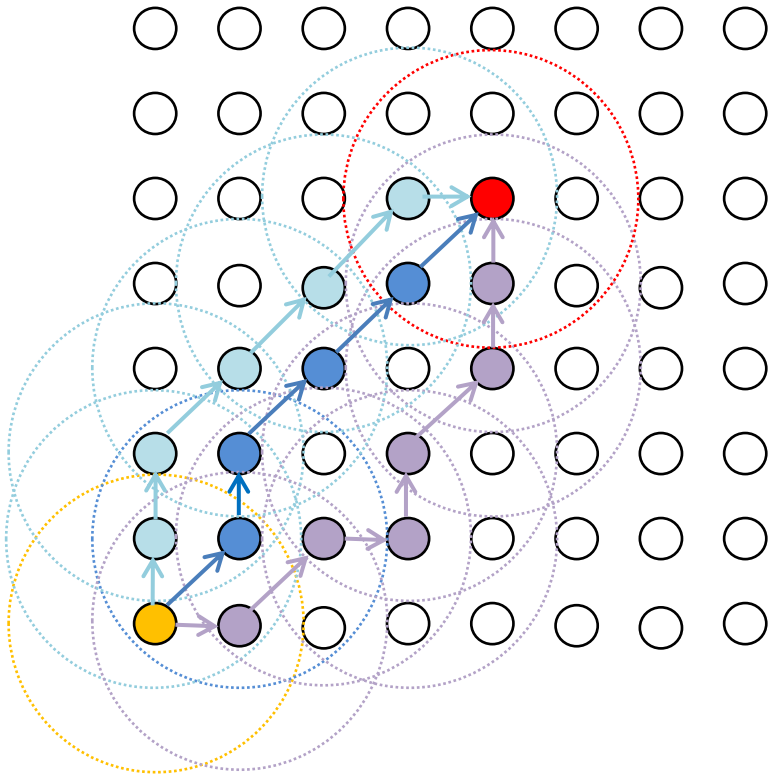


Figure 3.8: Sensor nodes in three different routes with communication ranges.

3.3.2.4 Selection of the Most Trusted Route (SMTR)

The route selection is obtained by using Select the Most Trusted Route (SMTR) algorithm. Here, RT values of all possible routes from source to sink are examined and the route with highest RT is selected as the routing path. The flow diagram of the proposed ITLSRP and SMTR are shown in Fig. 3.9.

In the given example, highest RT is obtained on route 1 and hence it is selected as the final routing path, as shown in Fig. 3.10.

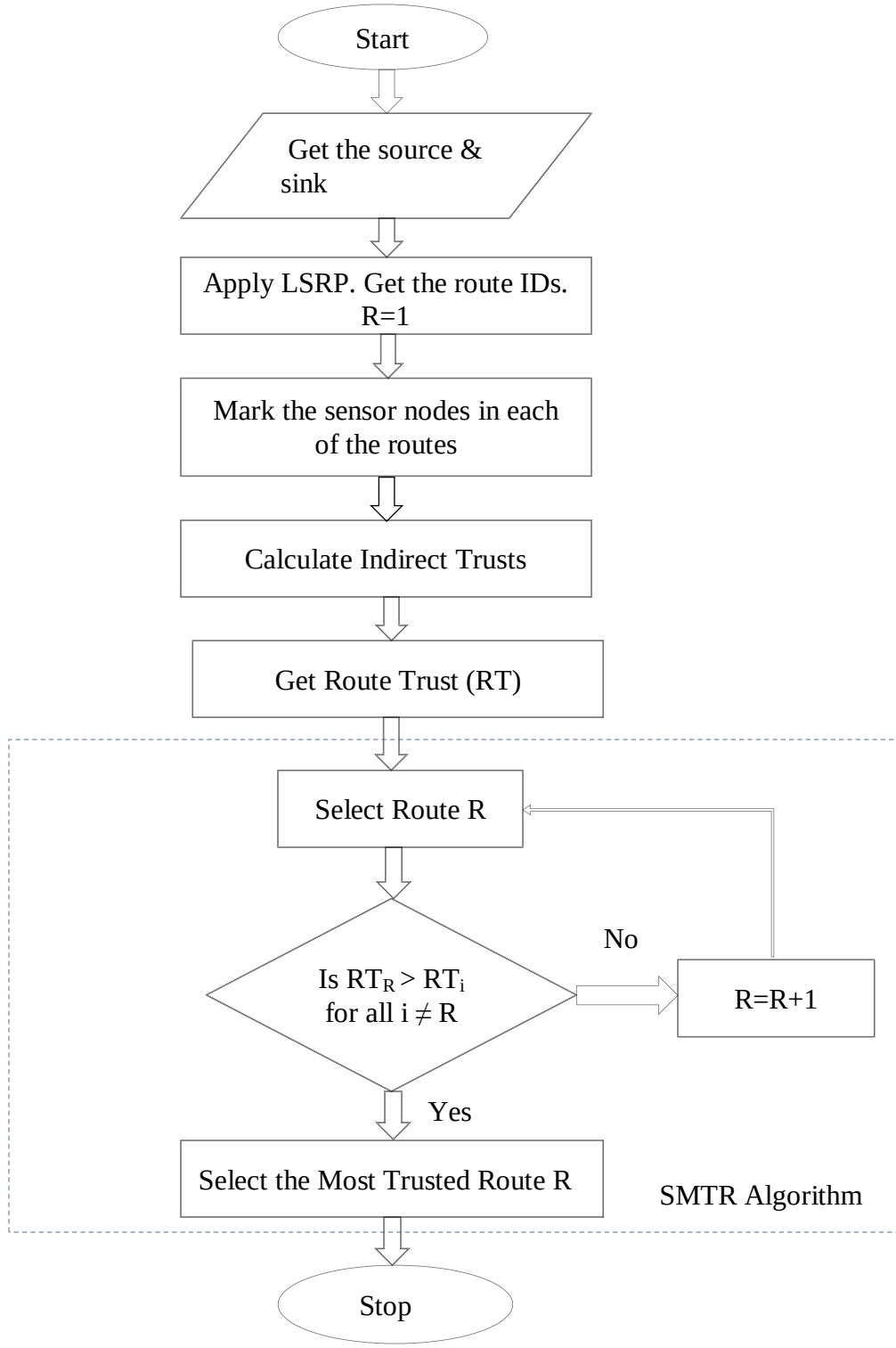


Figure 3.9: Flow diagram of ITLSRP and SMTR algorithm.

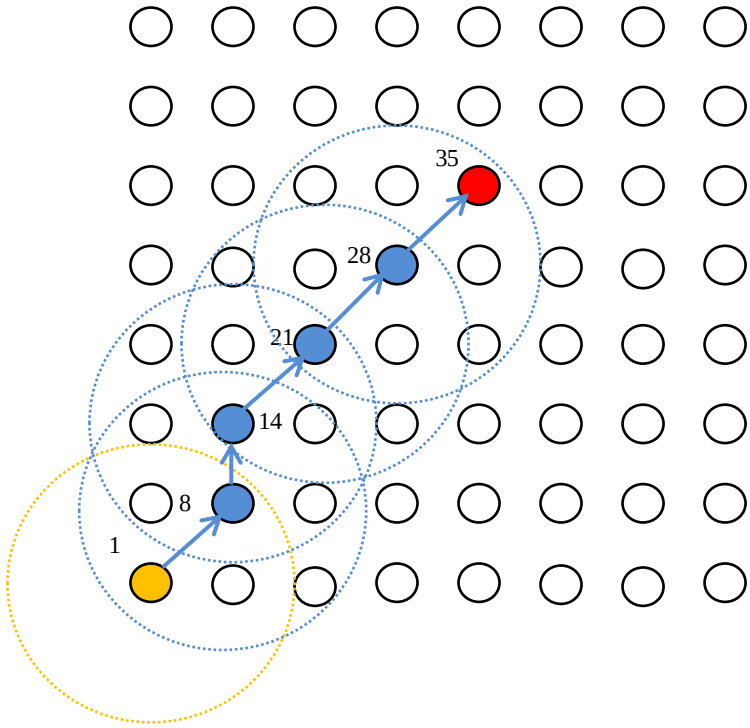


Figure 3.10: Final routing path (Route 1).

3.3.3 Simulation Results of ITLSRP Algorithm

The proposed ITLSRP algorithm is implemented in TinyOS based IRIS motes under the conditions given below:

- A symmetric square distributed field of 30 homogeneous sensor nodes were taken.
- Initially, all the nodes were fully powered.
- Equivalent Trust values of 0.5 were given in prior to all the above mentioned nodes and they sent messages in proportional to those values.

The proposed ITLSRP algorithm is compared with the previously existing protocols like ATSR [TZH10], DTLSRP [SAM11] and TILSRP [ARM11] respectively. In each case, better packet transmission and delay performance have been observed. For initialisation, it was assumed that the default trust value of each of the nodes with respect to any other sensor node is 0.5. This is a trivial assumption and the subsequent periodic updating of the routing tables will have a cumulative effect on the routing

performance. The bar graphs as given in Fig. 3.11 represents the comparison of the successful packet transmission for the proposed ITLSRP with the existing protocols ATSR [TZH10], DTLSRP [SAM11] and TILSRP [ARM11] respectively for five different initial trust values (0.1, 0.25, 0.5, 0.75 and 0.9) of the nodes under consideration. Here, five sets of simulation experiments are conducted independently and it is noted that the number of successful packet transmission increases in the proposed ITLSRP compared to the other existing protocols for each case of initial trust value. Similarly, Fig. 3.12 represents comparison graphs for delay in transmission of packets from source to sink versus initial trust values of the nodes. In these plots the symbol A, D, T and I represent the routing protocols ATSR [TZH10], DTLSRP [SAM11], TILSRP [ARM11] and the proposed ITLSRP respectively. The results represent a snapshot after a time of 1 day from the initiation.

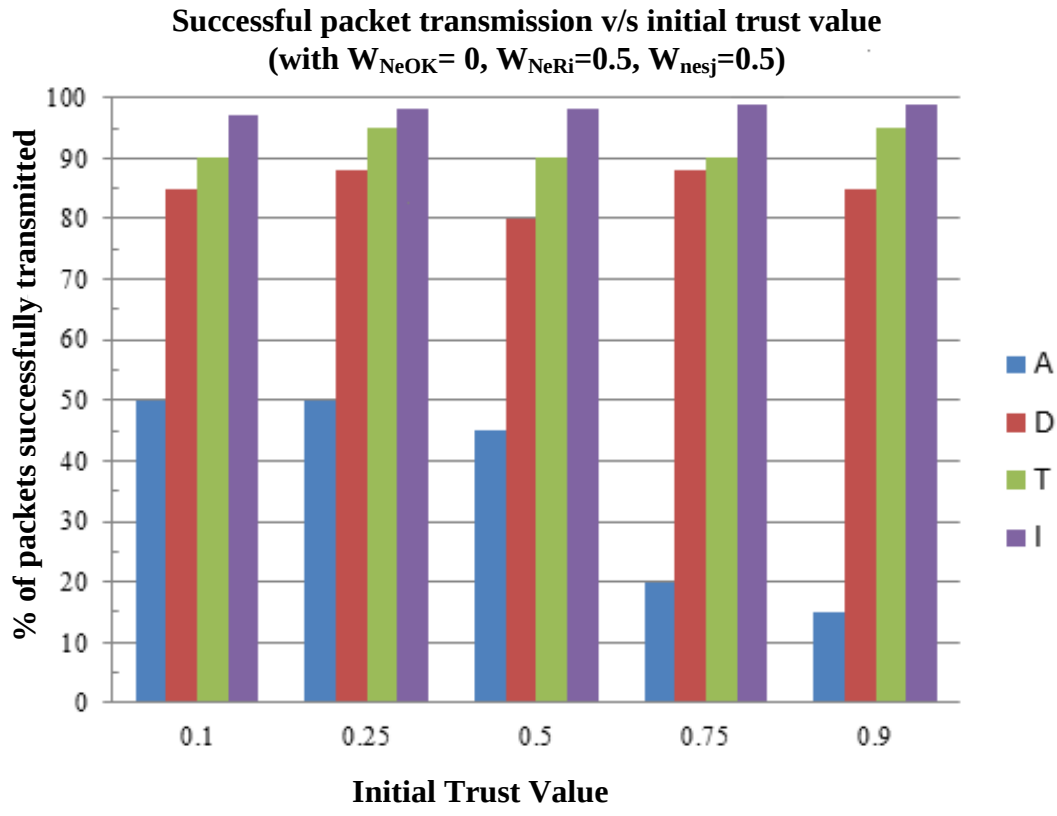


Figure 3.11: Successful packet transmission versus initial trust value.

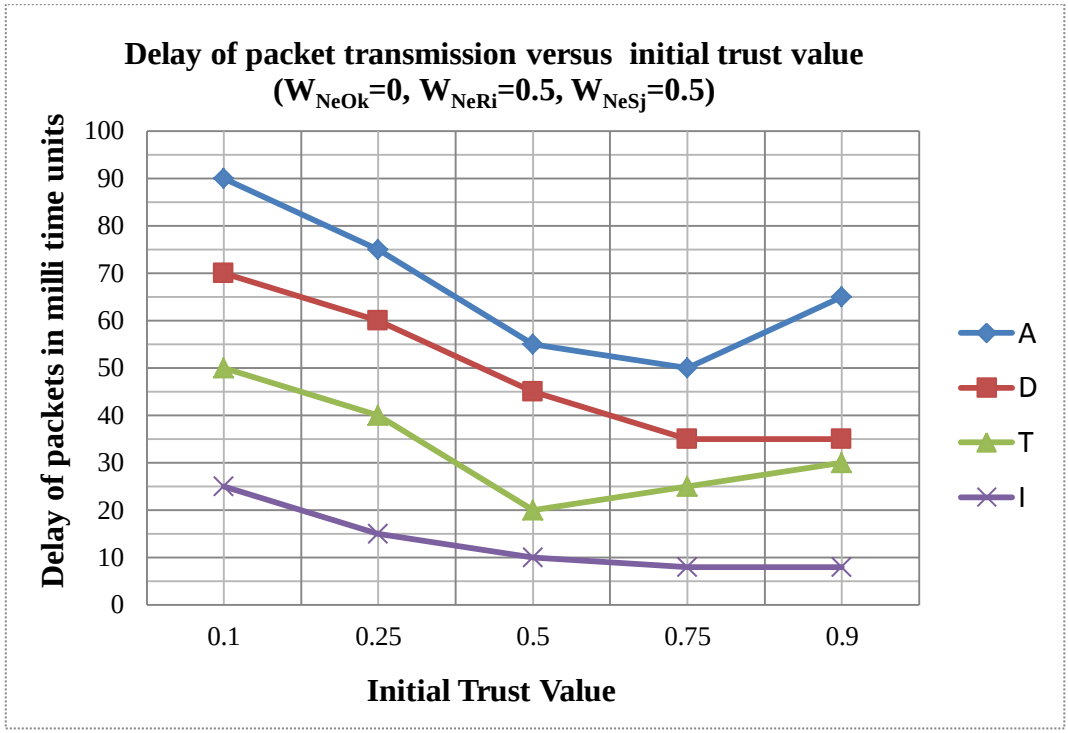


Figure 3.12: Delay in packet transmission versus initial trust value.

It can be appropriately concluded that the proposed ITLSRP algorithm shows better results compared to other similar protocols for all values of TTH. The introduction of this type of generic algorithm provides a number of merits over the dedicated Direct Trust methods. By controlling the weightage given to the Direct and Indirect Trusts in the Route Trust calculations of different paths, the robustness and versatility of the trust based algorithms are justified, which is also verified by the simulation results.

3.4 Proposed Algorithm 2:

A Fuzzy Based Trustworthy Route Selection Protocol (FTRSP) using LSRP in Wireless Sensor Networks

The proposed Fuzzy based Trustworthy Route Selection Protocol (FTRSP) is divided into two parts. In part one, individual Direct Trust (DT) of the sensor nodes within the network is calculated using Fuzzy Logic Controller. In part two, the Best Trustworthy Route (BTR) from source to sink is evaluated using LSRP.

3.4.1 FTRSP: Part One

A new Fuzzy Logic (FL) based trust evaluation model for the Wireless Sensor Network is presented here.

3.4.1.1 Basics of Fuzzy Logic and Fuzzy Logic Controller

The concept of Fuzzy Logic (FL) was conceived in 1965 by Prof. Lotfi Zadeh, University of California, Berkley [AMK05], [JBI06], [VBR95]. It was conceived as a way to process imprecise data. However, its usefulness was not seen at that time due to insufficient small computing capability prior to that time. In 80's, Japanese and European had aggressively implemented Fuzzy logic in real products whereas the US manufacturers had not been so quick to embrace this technology. Nowadays, the applications of Fuzzy logic are widely used in various automatic control systems and others. The Fuzzy Logic is inherently robust and less rigid than the traditional logic system.

The concept of Fuzzy Logic is much closer to human thoughts and natural language. Fuzzy logic starts with the concept of fuzzy set. The logic based on True and False is sometimes inadequate when describing human reasoning. There are many sets in the world that are defined by a non-distinct boundary, for example, the set of high mountains, the set of low level measurements, set of long persons, weather report regarding high temperature etc. Prof. Zadeh decided to extend two valued logic, defined by the binary pair {0, 1}, to the whole continuous interval [0, 1], thereby introducing a gradual transition from falsehood to truth. A grade of membership is considered so that the transition from membership to non-membership is gradual rather than abrupt. A fuzzy set is defined as a set without crisp or clearly defined boundary. It can contain elements with only a partial degree of membership. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value or the degree of membership from 0 to 1. The input space is termed as the universe of discourse. Thus, the elements of a fuzzy set are taken from a universe of discourse or universe, in short [AMK05], [JBI06], [VBR95]. In Fuzzy Logic world, the numeric data is expressed into word language, known as the Linguistic Variable. The Fuzzy Logic provides a simple way to arrive at a definite conclusion in terms of Linguistic Variable, on the basis of the vague, ambiguous, imprecise, noisy or missing

input information. The controller that is designed using the Fuzzy Logic is termed as the Fuzzy Logic Controller (FLC). The model of Fuzzy Logic Controller proposed by Mamdani consists of four components, viz., Fuzzifier, Fuzzy rules, Fuzzy Inference mechanism and Defuzzifier respectively [AMK05], [JBI06], [VBR95]. The Fuzzifier converts crisp input data to the suitable input Fuzzy sets. Fuzzy Inference mechanism derives Fuzzy output by combining Fuzzy rules into a mapping routine from input to output of the system. Finally Defuzzifier converts the output fuzzy set to a crisp value. The most popular model of Fuzzy Logic Controller consists of four components, namely Fuzzifier, Fuzzy rule, Fuzzy Interference Engine and Defuzzifier. The general architecture of the Fuzzy Logic Controller is presented in Fig. 3.13.

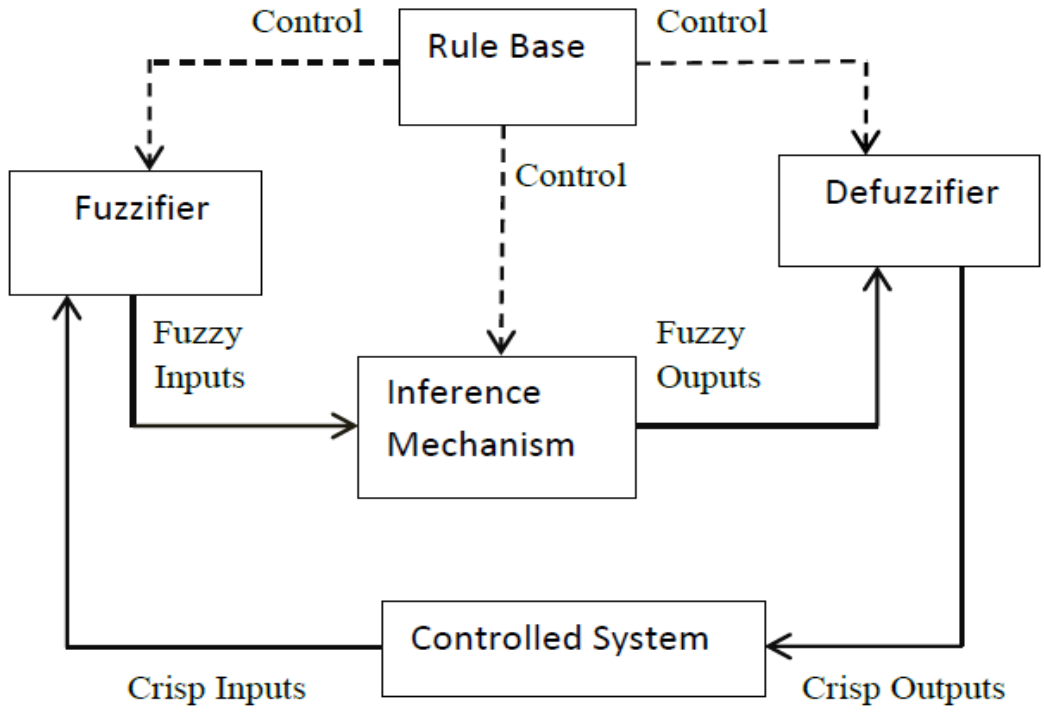


Figure 3.13: General architecture of the Fuzzy Logic Controller.

In the proposed FTRSP algorithm, most commonly used Mamdani's model is considered for designing FLC which is represented in five steps as described below:

Step 1: Defining Inputs and Outputs for the FLC

The range of values taken by the inputs and outputs are called the **universe of discourse**. In Step 1, we need to define the universe of discourse for all the inputs and outputs of the FLC. These values are called **crisp value.**

Step 2: Fuzzification of Inputs

Fuzzifier or Fuzzification converts input data into suitable Fuzzy sets. In Step 2, all the crisp values of the variables are converted to Fuzzy variables using membership functions. Different types of membership functions are used. The triangular and trapezoidal membership functions are most commonly used in practice. There are some general guidelines to determine the range of the Fuzzy variables as related to the crisp inputs.

- Distribute the Fuzzified values symmetrically across the universe of discourse.
- Use an odd number of Fuzzy sets for each variables so that some set is assumed to be in the middle.
- Overlap adjacent sets typically by 15% to 25%.

Step 3: Set up Fuzzy Membership Functions for the outputs

In Step 3, outputs are defined in terms of Fuzzy variables.

Step 4: Creating Fuzzy Rule Base

In this step, a set of Fuzzy rules are constructed that describe the operation of the FLC. The rule base consists of a collection of **IF-THEN rules**. The **Inference mechanism** combines these rules into a mapping routine from the inputs to the outputs of the system, to derive a reasonable output conclusion, in terms of Fuzzy sets. The efficiency of the FLC depends on the number of Fuzzy IF-THEN rules used for computation.

Step 5: Defuzzification of Outputs

A suitable Defuzzification strategy is required to convert the output Fuzzy set to the crisp output values. Several methods for Defuzzification are used in practice.

The most commonly used methods are

- Center of area (COD) or centroid method,
- Fuzzy OR method, and
- Weighted Average method.

3.4.1.2 Trust Evaluation Model Using Fuzzy Logic Controller

In the proposed FTRSP model, FLC is designed considering three input parameters, called Data Packet Transmission Ratio (DPTR), Control Packet Transmission Ratio (CPTR) and Battery Life (BLIFE) respectively. The Trust value of the sensor node is considered as the output variable. It is to be noted that the computational complexity of FLC increases with the increase in number of input and output. Hence, for the sake of simplicity in computation and ease of simulation, we have considered only three most relevant input parameters that are used to determine the routing properties of the sensor nodes. The descriptions of input parameters DPTR, CPTR and BLIFE are given in Table 3.2. All the input variables are restricted to positive values.

Table 3.2: Description of Inputs

Inputs	**Description**
Data Packet Transmission Ratio (DPTR)	Ratio of the number of data packets sent by the source node to the number of acknowledgement received.
Control Packet Transmission Ratio (CPTR)	Ratio of the number of control packets sent by the source node to the number of acknowledgement received.
Battery life (BLIFE)	Remaining battery life

Table 3.3: Fuzzy variable ranges for DPTR and CPTR

Crisp Input Range	**Fuzzy Variable Range for DPTR**	**Fuzzy Variable Range for CPTR**
0 – 0.45	VL (Very Low)	VL (Very Low)
0.4 – 0.6	L (Low)	L (Low)
0.55 – 0.75	M (Medium)	M (Medium)
0.7 – 1	H (High)	H (High)

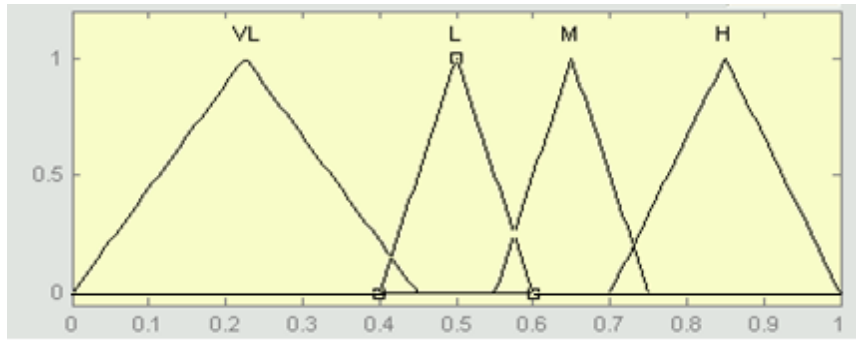
Table 3.4: Fuzzy variable ranges for BLIFE

Crisp Input Range	**Fuzzy Variable Range for BLIFE**
0 – 1.25 V	VL (Very Low)
1- 1.6 V	L (Low)
1.5 – 2.25 V	M (Medium)
2.15 – 3 V	H (High)

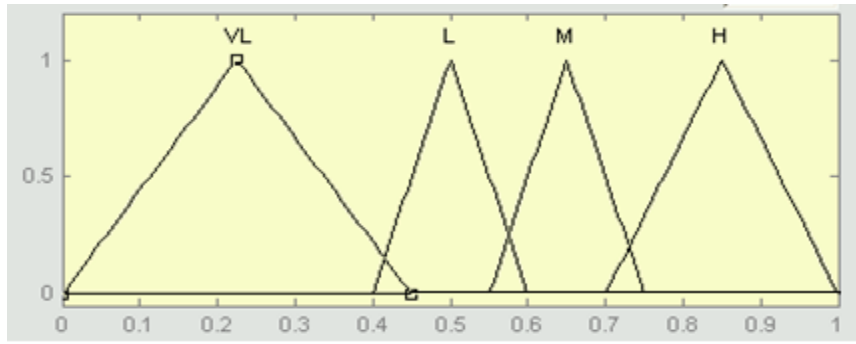
Table 3.5: Fuzzy variable ranges for Trust

Crisp Output Range	**Fuzzy Variable Range for Trust**
0 – 0.45	VLT (Very Low Trust)
0.4 – 0.6	LT (Low Trust)
0.55 – 0.75	MT (Medium Trust)
0.7 – 1	HT (High Trust)

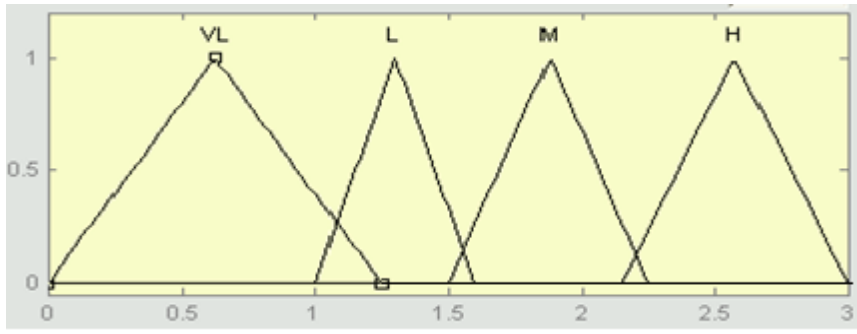
Table 3.3 represents assignment of ranges (crisp input range and Fuzzy variable range) for DPTR and CPTR respectively. Similarly, Table 3.4 shows same for the BLIFE. In the proposed model, Trust of the individual sensor nodes is taken as the output variable. The crisp range and the Fuzzy membership assignment for the output variable are shown in Table 3.5. The triangular Fuzzy membership functions for input and output variables are shown in Fig. 3.14.



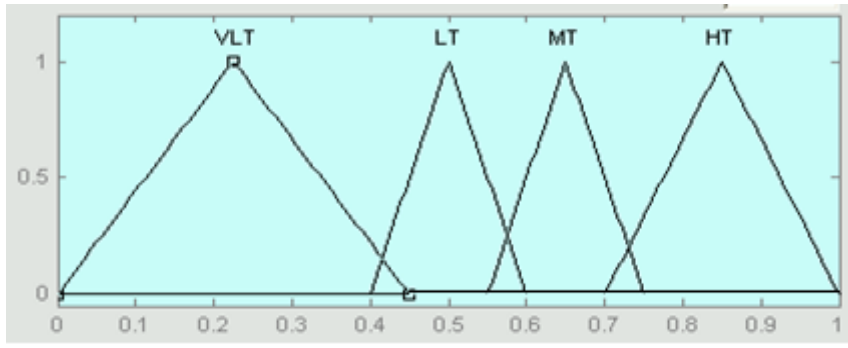
Input variable - Data Packet Transmission Rate



Input variable - Control Packet Transmission Rate



Input variable - Battery Life



Output variable – Trust

Figure 3.14: Fuzzy membership functions for input and output variables.

Table 3.6: Fuzzy Rule Base for inferring trust

Rule	**DPTR**	**CPTR**	**BLIFE**	**Trust**
1 to 16	VL	VL/L/M/H	VL/L/M/H	VLT
17,18	L	VL	VL/L	VLT
19,20	L	VL	M/H	LT
21,22,23	L	L	VL/L/M	VLT
24	L	L	H	LT
25,26	L	M	VL/L	VLT
27,28	L	M	M/H	LT
29,30,31	L	H	VL/L/M	LT
32	L	H	H	MT
33,34	M	VL	VL/L	LT
35,36	M	VL	M/H	MT
37,38,39	M	L	VL/L/M	LT
40	M	L	H	MT
41 to 44	M	M	VL/L/M/H	MT
45	M	H	VL	LT
46,47	M	H	L/M	MT
48	M	H	H	HT
49	H	VL	VL	LT
50	H	VL	L	MT
51,52	H	VL	M/H	HT
53	H	L	VL	LT
54,55	H	L	L/M	MT
56	H	L	H	HT
57	H	M	VL	MT
58,59,60	H	M	L/M/H	HT
61	H	H	VL	MT
62,63,64	H	H	L/M/H	HT

The Fuzzy Rule Base for inferring trust of individual sensor nodes are listed in Table 3.6. Rule 1 can be interpreted as "if DPTR, CPTR and BLIFE of a sensor node are all VL, then inferred trust of the node is VLT". Similarly, Rule 2 is interpreted as "if DPTR is VL, CPTR is VL and BLIFE is L, then the trust is VLT" and so on.

As discussed, three inputs and one output variables are considered here. More numbers of Fuzzy sets could be formed, thus giving a more accurate result. However, choosing the above sets proved to be more practical and easy to implement on TinyOS based IRIS motes rather than choosing a huge number of sets resulting in a very large number of Fuzzy rules. It is practically infeasible for the low memory and low processing capability of the sensor nodes.

3.4.2 FTRSP: Part Two

The proposed FTRSP model implements LSRP as the basic routing protocol. However, as discussed in part two, FTRSP does not require the crisp output value of trust. The Fuzzy trust value is utilized in FTRSP model directly, where Defuzzification of trust is redundant. This method significantly decreases calculation overhead and processing power consumption.

3.4.2.1 Route Selection Procedure

The route selection procedure in FTRSP consists of the following methods:

- All possible routes from source to sink are extracted through LSRP.
- Best Trustworthy Route (BTR) is obtained using **Route Search (RS) Algorithm** and **Best Route Selection (BRS) Algorithm.**

The parameters and notations of the mathematical formulations used in route selection procedure are described in Table 3.7.

Table 3.7: Notation of the mathematical formulation

Parameter	Description
N	Total number of routes available from source to sink, as obtained by implementing LSRP.
n(A)	Cardinality of the set A, i.e. the number of elements of set A.
A_k	k^{th} element of set A.
min(A)	Minimum value element of set A.
max(A)	Maximum value element of set A.
RT_k	Different route ids where k ϵ [1, N]. It is an integer.
n_i	Number of nodes in the route Rt_k where i is an integer.
S_{ik}	i th node of the k th route RT_k, where i ϵ [1, n_k] and k ϵ [1, N] and i, k both are integers.
M	Number of levels to which the trust values of the nodes is divided. As shown in Table 3.5, Trust is divided into 4 levels, namely Very Low Trust (VLT), Low Trust (LT), Medium Trust (MT) and High Trust (HT) respectively.
T_i	Trust level of the node S_{ik} where i ϵ [1, m]. If k < i, then the level T_k represents lesser trustworthiness than the level T_i. For example, T_1, T_2, T_3 and T_4 corresponds to the VLT, LT, MT and HT respectively and $T_1 < T_2 < T_3 < T_4$.
R_k	It is the set representing the trust values of the nodes in the route RT_k .
M	It is the set {min(R_1), min(R_2), . . . , min(R_N) } where min(R_i) is the lowest trust level in the set R_i.
BR	It is the set {RT_m , RT_n , . . . , RT_l }, which represents the set of best possible routes. The proposed algorithm selects one route from the set BR as the Best Trustworthy Route (BTR).
NBR	It is the set {n_m , n_l , . . . } where NBR represents the number of nodes in the route BR.

3.4.2.2 Best Route Selection Algorithm

Input: All possible routes from source to sink as calculated from Link State Routing Protocol

Output: The Best Trustworthy Route (BTR)

Begin

Step 1: Construct the set R_k for the route RT_k. Repeat the same for all routes.

Step 2: Construct the set M and then find max(M).

max(M) represents the highest trust level among the elements of the set M.

Construct the set BR = {RT_m, RT_n, . . . } for all $RT_i \in BR$

where min(R_i) = max(M).

Step 3: If n(BR)=1 and BR={RT_i} then choose RT_i as the best route.

Output this as the only **Best Trustworthy Route (BTR)**.

If n(BR) ≠ 1, min_trust_level= min(R_i) for $RT_i \in BR$ and input_routes= BR.

Step 4: Apply the **Route Search Algorithm** with min_trust_level and input_routes as the input parameters.

Step 5: BR= set returned by the **Route Search Algorithm.**

Step 6: If n(BR) = 1, then choose Rt_i as the **Best Trustworthy Route (BTR)**.

BR={Rt_i} and return it as the output

If n(BR)≠ 1 then go to Step 7

Step 7: Find the set NBR.

Step 8: Update the set BR= {RT_m , RT_n , . . .} for all $RT_i \in BR$, n_i = min(NBR).

Step 9: If n(BR) = 1 and BR={Rt_i} then choose Rt_i as the **Best Trustworthy Route (BTR)**.

If n(BR) ≠ 1 and min_trust_level ≠ maximum trust level then set min_trust_level = next highest trust level to the present min_trust_level and input_routes= {BR} and go to step 3.

If n(BR) ≠ 1 and min_trust_level = maximum trust level present, then output any one of BR as the **Best Trustworthy Route (BTR)**.

End

3.4.2.3 Route Search Algorithm

Input: min_trust_level, input_routes,

Output: selected_routes

Begin

Step 1: num_nodes = { n_1, n_2, . . . } where n_i = number of nodes in the route with trust level=min_trust_level .

Step 2: selected_routes ={ RT_m , RT_n , . . .}

for all RT_i ϵ selected_routes, n_i = min(num_nodes).

End

Examples:

Suppose, in a Wireless Sensor Network the trust values of the nodes are divided into four levels, namely, T_1 (Very Low Trust), T_2 (Low Trust), T_3 (Medium Trust) and T_4 (High Trust) respectively.

Case 1

Let us consider ten routes RT_1, RT_2, RT_3, RT_4, RT_5, RT_6, RT_7, RT_8, RT_9 and RT_{10} be available from source to sink, as obtained from Link State Routing Protocol.

Step 1 of BRS Algorithm yields the sets R_i where i ϵ [1, 10].

R_1= {T_1, T_1, T_2, T_3, T_4}

R_2= {T_1 T_2, T_2, T_3, T_3, T_4}

R_3= {T_2, T_3, T_1}

R_4= {T_1, T_1, T_2, T_2}

R_5= {T_1, T_1, T_1}

R_6= {T_2, T_2, T_3, T_3, T_4}

R_7= {T_2, T_3, T_3, T_3, T_3}

R_8= {T_2, T_2, T_2, T_2, T_2}

R_9= {T_2, T_2, T_3, T_3, T_3}

R_{10}= {T_2, T_3, T_3, T_3, T_4}

Step 2 of BRS Algorithm generates the set BR

M = {Min(R_1), min(R_2), min(R_3), min(R_4), min(R_5), min(R_6), min(R_7), min(R_8), min(R_9), min(R_{10})}

Here, M = {T_1, T_1, T_1, T_1, T_1, T_2, T_2, T_2, T_2, T_2}

Now, Max(M) = T_2 which corresponds to min(R_6), min(R_7), min(R_8), min(R_9) and min(R_{10})

BR= {R_6, R_7, R_8, R_9, R_{10}}

Step 3 of BRS Algorithm

In this case, n(BR) = 5

min_trust_level = min(R_i) = T_2

input_routes = BR = {R_6, R_7, R_8, R_9, R_{10}}

Step 4 of BRS Algorithm

Go to **Route Search Algorithm (RS Algorithm)**

Step 1 of RS Algorithm yields

num_nodes = {2, 1, 5, 2, 1}

that means, the set R_6 has 2 elements (nodes) with Trust level T_2, R_7 has 1 element (node) with Trust level T_2, R_8 has 5 elements (nodes) with Trust level T_2 and so on.

Step 2 of RS Algorithm yields

selected_routes = {R_7, R_{10}}

The sets having minimum number of elements (nodes) with Trust level T_2 are selected. Here, R_7, R_{10} are selected, since both have only one element (node) with Trust level T_2.

Step 5 of BRS Algorithm

BR= set returned by the RS Algorithm

Here, BR = {R_7, R_{10}}

Step 6 of BRS Algorithm

n(BR) = 2

Go to Step 7

Step 7 of BRS Algorithm

NBR = {5, 5}

Since, the number of elements (nodes) is 5 in both of the sets R_7 and R_{10}.

Step 8 of BRS Algorithm

In this case, min(NBR) = 5, so updating BR results no change.

BR = {R_7, R_{10}}

Step 9 of BRS Algorithm

n(BR) = 2

set min_trust_level = T_3

input_routes = BR = {R_7, R_{10}}

Go to Step 4 of BRS Algorithm

Step 4 of BRS Algorithm

Go to **Route Search Algorithm (RS Algorithm)**

Step 1 of RS Algorithm yields

num_nodes = {4, 3}

that means, the set R_7 and R_{10} have respectively 4 element (nodes) and 3 elements (nodes) with Trust level T_3.

Step 2 of RS Algorithm yields

selected_routes = $\{R_{10}\}$

The set R_{10} is selected, since it has less number of elements (nodes) with Trust level T_3, as compared to R_7.

Step 5 of BRS Algorithm

BR= set returned by the RS Algorithm

Here, BR = $\{R_{10}\}$

n(BR) = 1

As there is only one element in the set BR, the algorithm will stop. The set R_{10} corresponds to the route RT_{10}. So, in this way, the r**oute RT_{10} is selected as the Best Trustworthy Route.**

Case 2

Let us consider seven routes with route ids RT_1, RT_2, RT_3, RT_4, RT_5, RT_6 and RT_7 to be available from source to sink as obtained from Link State Routing Protocol.

Step 1 of BRS Algorithm yields the sets R_i where i ϵ [1, 7].

R_1= {T_1, T_1, T_2, T_3, T_4}

R_2= {T_1, T_2, T_2, T_3, T_3, T_4}

R_3= {T_2, T_3, T_4}

R_4= {T_2, T_2, T_3, T_4}

R_5= {T_2, T_1}

R_6= {T_2, T_2, T_3, T_4, T_1}

R_7= {T_2, T_3}

Step 2 of BRS Algorithm generates the set BR

M = {Min(R_1), min(R_2), min(R_3), min(R_4), min(R_5), min(R_6), min(R_7), min(R_8), min(R_9), min(R_{10})}

Here, M = {T_1, T_1, T_2, T_2, T_1, T_1, T_2}

Now, Max(M) = T_2 which corresponds to min(R_3), min(R_4) and min(R_7)

BR= {R_3, R_4, R_7}

Step 3 of BRS Algorithm

In this case, n(BR) = 3

min_trust_level = min(R_i) = T_2

input_routes = BR = {R_3, R_4, R_7}

Step 4 of BRS Algorithm

Go to **Route Search Algorithm (RS Algorithm)**

Step 1 of RS Algorithm yields

num_nodes = {1, 2, 1}

that means, the set R_3 has 1 element (node) with trust level T_2, R_4 has 2 elements (nodes) with trust level T_2 and R_7 has 1 element (node) with trust level T_2

Step 2 of RS Algorithm yields

selected_routes = {R_3, R_7}

The sets having minimum number of elements (nodes) with trust level T_2 are selected. Here, R_3 and R_7 are selected, since both have only one element (node) with trust level T_2.

Step 5 of BRS Algorithm

BR= set returned by the RS Algorithm

Here, BR = {R_3, R_7}

Step 6 of BRS Algorithm

n(BR) = 2

Go to Step 7

Step 7 of BRS Algorithm

NBR = {3, 2}

Since, the set R_3 has 3 elements (nodes) and the set R_7 has 2 elements (nodes) respectively.

Step 8 of BRS Algorithm

In this case, min(NBR) = 2, so updating the BR results

BR = {R_7}

Step 9 of BRS Algorithm

n(BR) = 1 and BR = {R_7}

Since, the set BR has only one element, choose the route RT_7 corresponding to the set R_7 as the **Best Trustworthy Route.**

If two routes have equal number of nodes with the same trust levels, then the route with the least number of nodes, i.e. the shortest route will be selected. Case 2 shows that route RT_3 and RT_7 both have one node with trust level T_2 and T_3 respectively. But the BRS algorithm chooses the route RT_7 as the Best Trustworthy Route, as it has less number of nodes (2 nodes) compared to the route RT_3 (3 nodes).

Advantages

The computational burden for implementing BRS algorithm is significantly less compared to the other methods such as DTLSRP [SAM11] and TILSRP [ARM11], where the route is selected by multiplying the trust values of individual nodes. Since, the sensor nodes have low complexity processors, implementing multiplication of trust values are computationally very expensive. On the other hand, the proposed BRS algorithm only finds the minima of a set and uses searching algorithm. These two algorithms (BRS algorithm and RS algorithm) are very simple to implement even in the low complexity processors of sensor nodes.

The only drawback of the proposed BRS algorithm is that, it may choose a longer route instead of a shorter route, if the lowest trust level of the shorter route is worse than the longer route. As a result, the shorter route will get eliminated in step 2 of the BRS algorithm. This can be overcome, if we consider only those routes, found out by LSRP which has the same number of nodes. For example if a source and a sink has 10 routes between them and the shortest route is 3 hop path, then all routes with 3 hop nodes will be given as the input to the BRS algorithm and the rest of the routes are ignored.

3.4.3 Simulation Results of FTRSP Algorithm

The Fuzzy implementation of the proposed FTRSP has been carried out in MATLAB using Mamdani model of the FIS Editor. The ranges of the input variables, namely Data Packet Transmission Ratio (DPTR), Control packet Transmission Ratio (CPTR)

and Battery Life (BLIFE) are assigned as per the Table 3.3 and Table 3.4 respectively. Similarly, the range of the output variable that is, the trust value of the node is assigned as per the Table 3.5. The snap shot of the Rule Base and the Rule Viewer used in MATLAB Fuzzy Toolbox are shown in Fig. 3.15 and 3.16 respectively. Since three inputs and one output variables are taken, the total sixty four rules are formed which are described in Table 3.6. The crisp value of the trust is simulated in MATLAB by the centroid method of Defuzzification, as shown in Fig. 3.16. It is to be noted that the output crisp trust value of the sensor nodes could be modified by changing the Rule Base in the Rule Editor Toolbox. Although, the Best Trustworthy Route is selected considering the Fuzzy trust value of the nodes, Defuzzification of trust is simulated to show the input output relationship, in this context. The 3 D graph as shown in Fig. 3.17 represents the variation of the trust value with Data Packet Transmission Ratio (DPTR) and Control Packet Transmission Ratio (CPTR). Similarly, Fig. 3.18 shows variation of the trust value with Control Packet Transmission Ratio (CPTR) and the Battery Life (BLIFE) of the individual sensor nodes. The variation of the trust value of the node with the Data Packet Transmission Ratio (DPTR) and the Battery Life (BLIFE) of the node is presented in Fig. 3.19.

The proposed FTRSP algorithm is implemented in TOSSIM simulator based on the TinyOS platform. An arbitrary network topology with 25 sensor nodes in duplex links is considered. The average residual battery life of the nodes and the average delay in transmission of packets for the given network are calculated using the proposed algorithm and the already existing TILSRP algorithm [ARM11]. In case of calculation of delay, it is measured at periodic equal intervals, such as every 1 hour with constant traffic generated throughout that time interval. The performance analysis and the comparison of the proposed FTRSP algorithm with its peer TILSRP are graphically shown in Figs. 3.20 and 3.21 respectively.

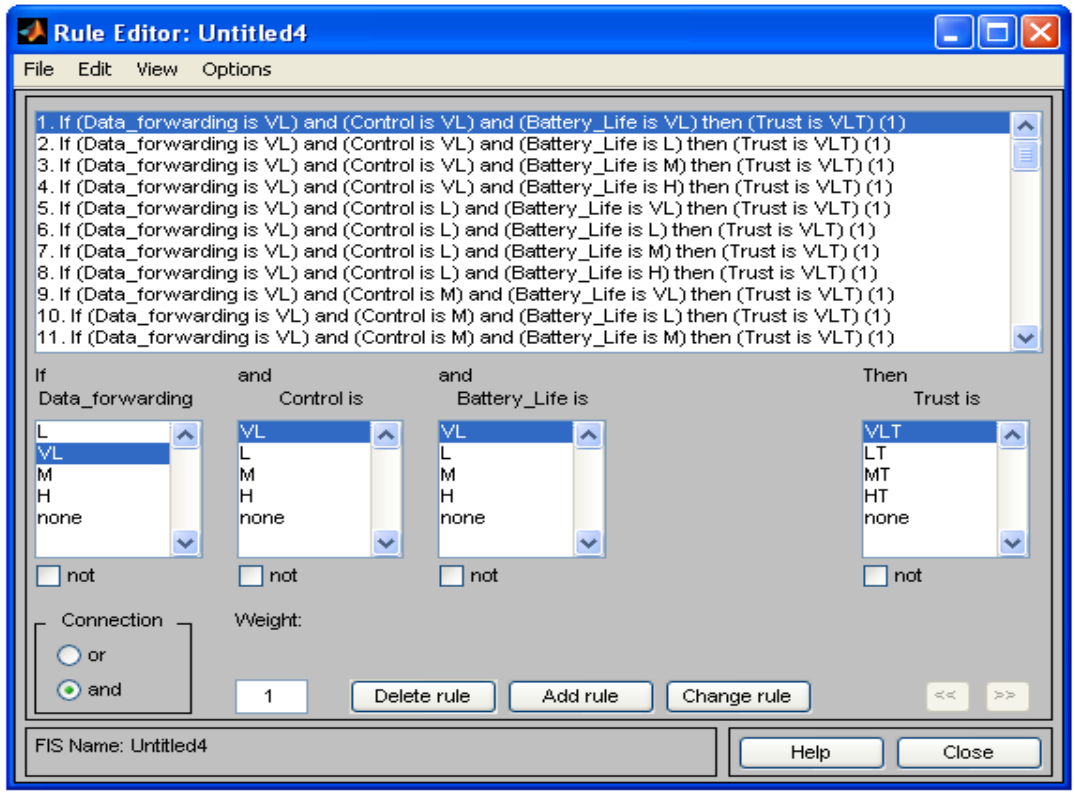


Figure 3.15: Snap shot of the rule base used in MATLAB simulation.

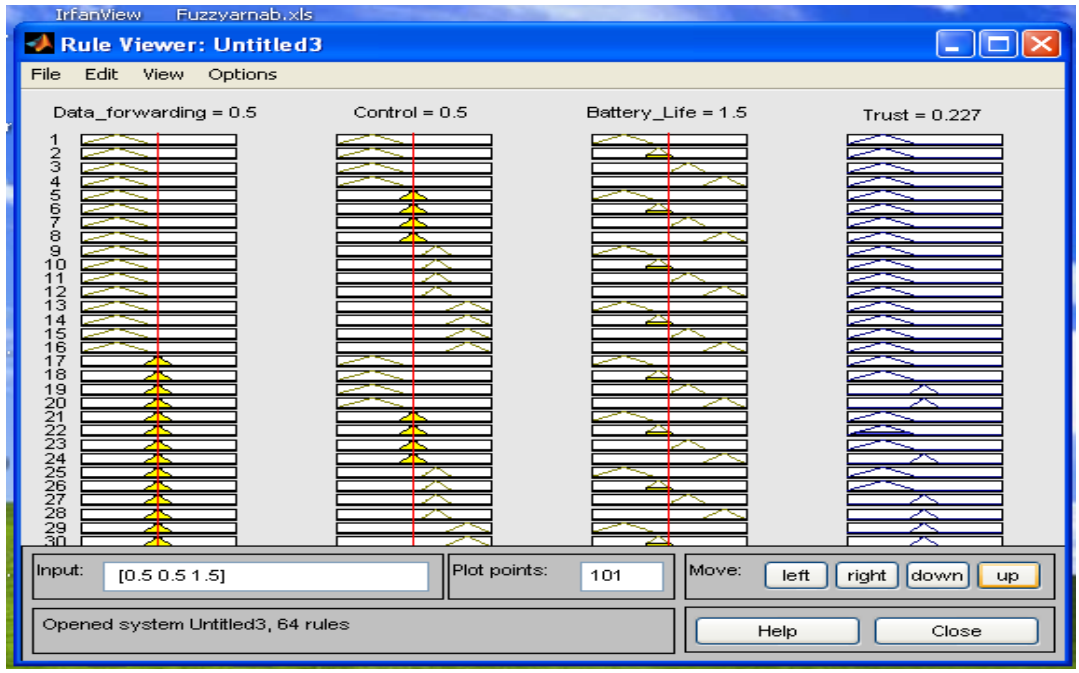


Figure 3.16: Rule Viewer in MATLAB.

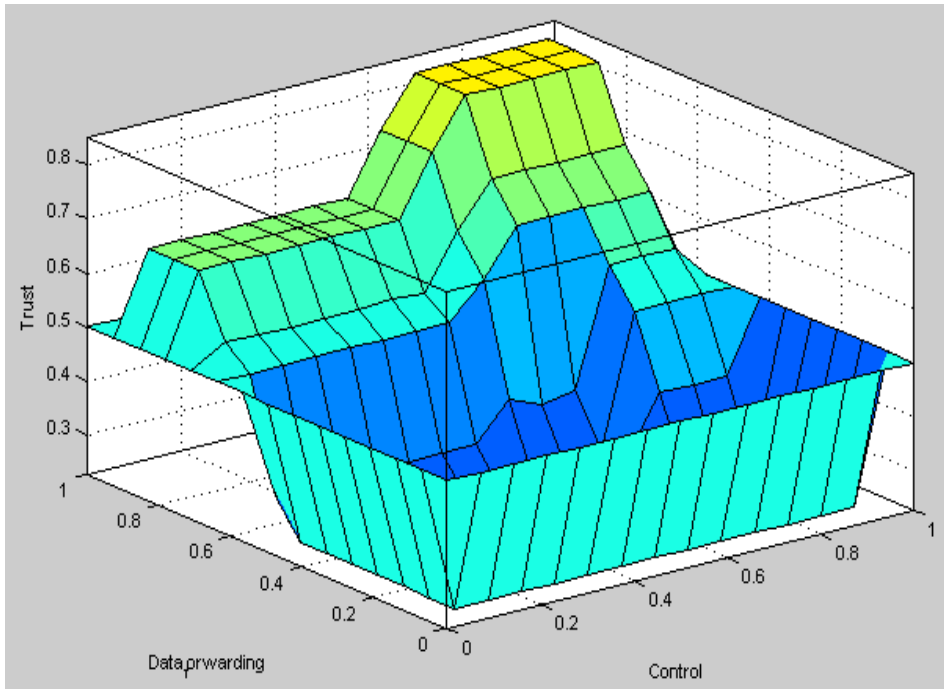


Figure 3.17: Variation of trust with DPTR and CPTR.

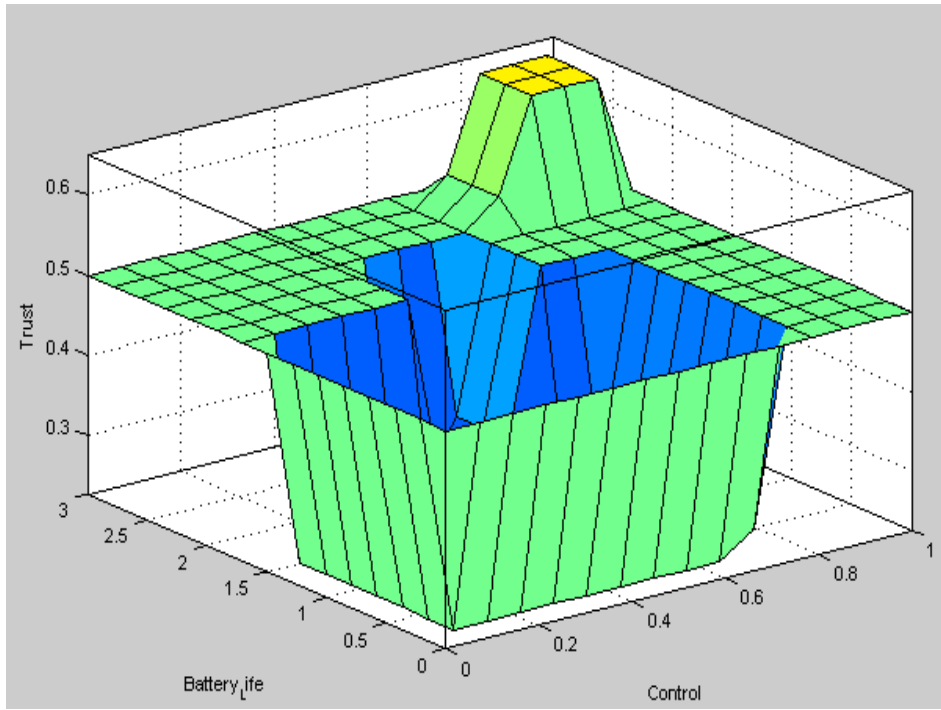


Figure 3.18: Variation of trust with CPTR and BLIFE.

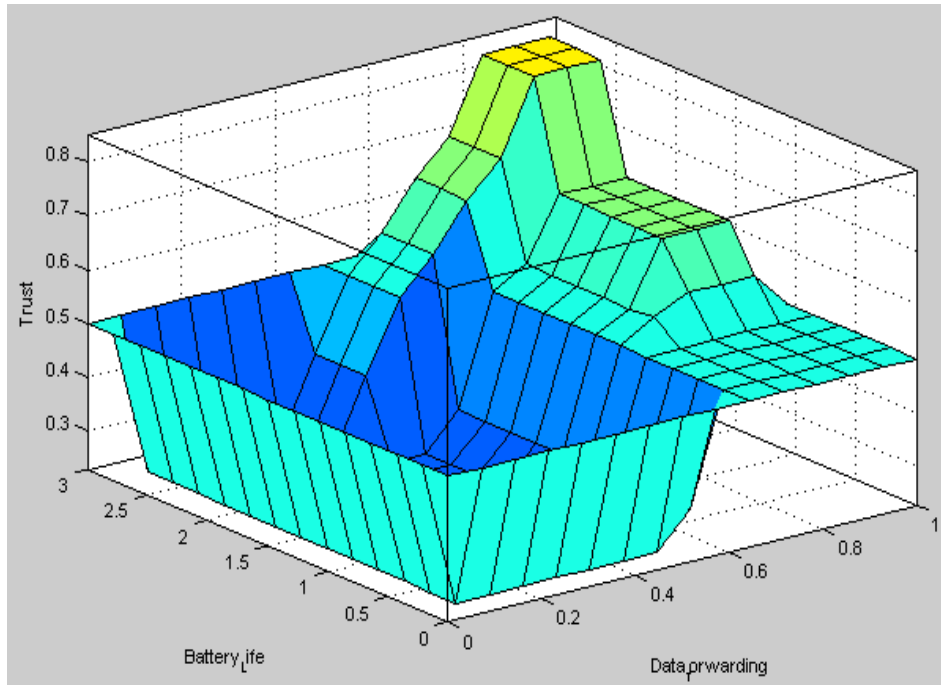


Figure 3.19: Variation of trust with DPTR and BLIFE.

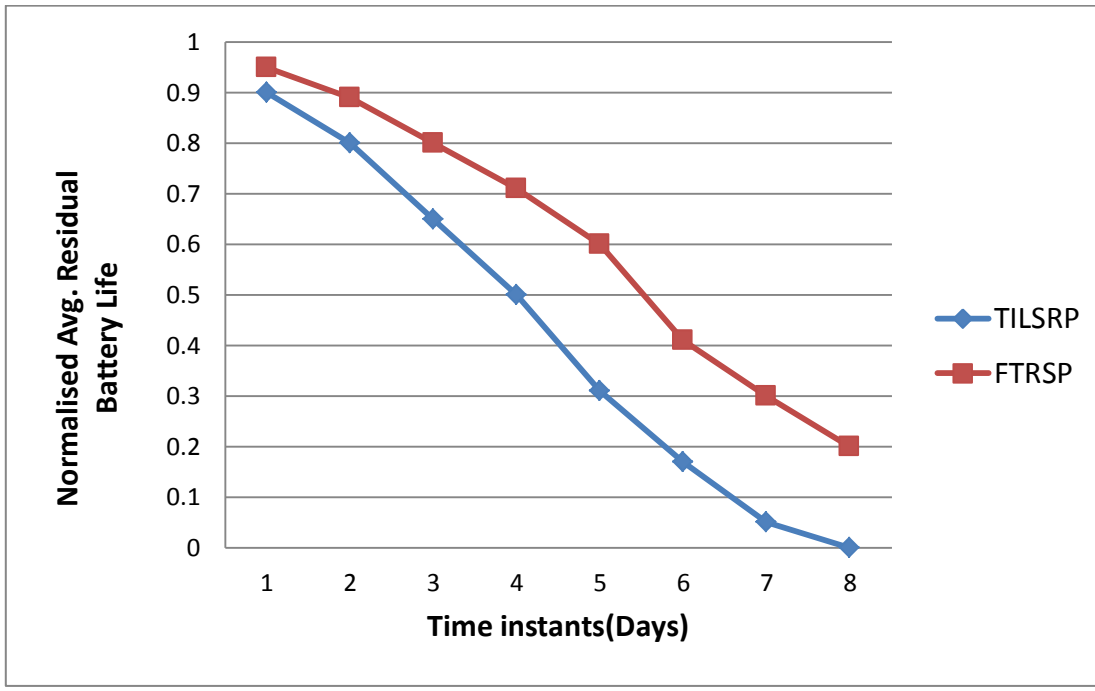


Figure 3.20: Graph showing the average residual battery life.

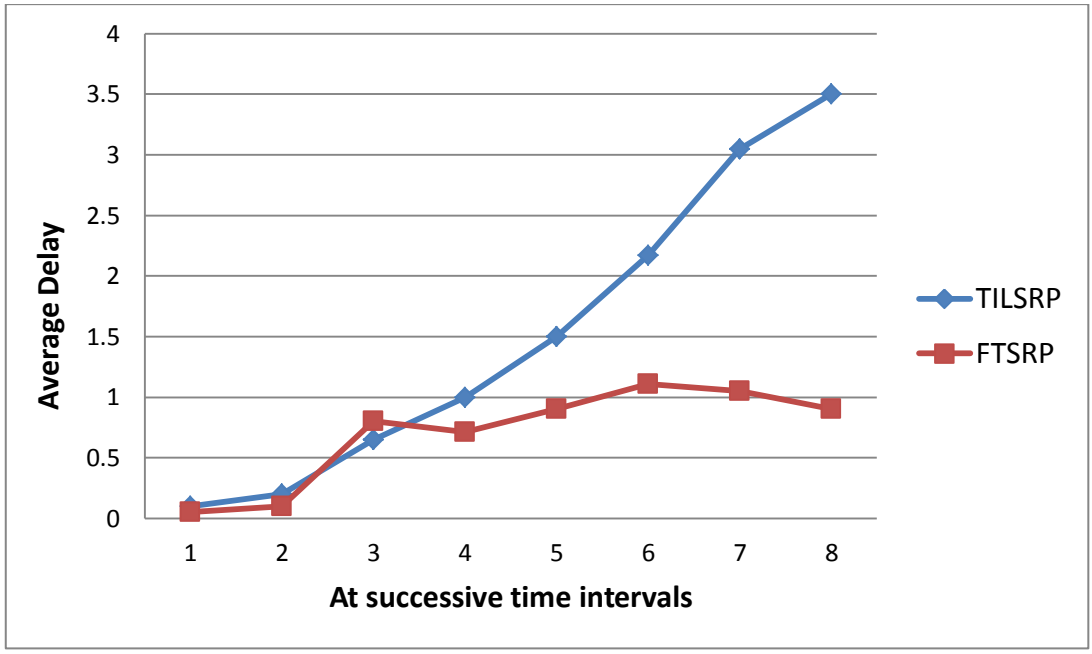


Figure 3.21: Average delay in data packet transmission from source to sink.

It can be appropriately concluded from the simulation results that the proposed FTRSP algorithm shows a much better performance compared to TILSRP algorithm, both with respect to the delay in transmission of packets and the average residual lifetime of the nodes, which directly affects the overall network performance.

3.5 Summary

This chapter describes two new light weight trust based data routing algorithms (ITLSRP and FTRSP) for WSN which show better results compared to the similar existing algorithms, in terms of residual battery life, average delay and the number of successful data packet transmission from the source to the sink. The proposed algorithms eliminate routes having sensor nodes with low trust values. Hence, the selected data routing path is trustworthy and reliable in nature. However, the network congestion and the congestion status of the sensor nodes during data packet transmissions are not considered in the proposed schemes discussed in Chapter 3. The proposed methods to quantify the congestion status and then integration of the trust value and the congestion status of the sensor nodes for getting the more reliable data routing scheme are discussed in the following chapters.

CHAPTER 4

A Genetic Algorithm Inspired Load Balancing Protocol for Congestion Control in Wireless Sensor Networks using Trust Based Routing Framework (GACCTR)

4.1 Introduction

Wireless Sensor Networks (WSNs) can be appropriately defined as a collection of spatially and temporally distributed sensors that are capable of interacting with the environment either by sensing or by controlling the physical parameters. Wireless Sensor Networks, which have been primarily a subject of only research interest till recent years, are nowadays increasing ubiquitously for widespread uses in industry for various types of monitoring, alerting, data gathering and surveillance applications. This is mainly due to the inherent advantages of the Wireless Sensor Networks that provide easy deployment, programmability and re-programmability, self-configurability and their multi-faceted applications to the consumers. However, Wireless Sensor Networks are largely constrained in terms of hardware specifications, processing capability, energy availability and maintainability.

Congestion Control (CC) and Congestion Avoidance (CA) are the two most desirable qualities that are required for the proper functioning of a Wireless Sensor Network. The congestion is generated due to the large amount of data transmission through the network. In densely distributed Wireless Sensor Networks, a large number of sensor nodes are usually deployed, which provide many alternative paths or routes between a particular set of source and sink. Appropriate route selections at various time instants play an important role for congestion avoidance in Wireless Sensor Networks. The entire data is divided into a number of packets which are then distributed among different routes by the source node and are accumulated and re-organized at the destination. The efficient routing of the data through the available paths is very essential for both accurate data communication and lifetime enhancement and this is one of the vital requirements for the energy constrained Wireless Sensor Networks.

Trust management system in Wireless Sensor Networks is new concept which is increasingly becoming essential for the reliability point of view. Even in engineering, the meaning of Trust is not entirely different from its literal meaning. It is attributed to the opinion that a sensor node forms with the other sensor nodes. Depending upon the Trust value or Trust level, a particular node can be considered to be either malicious or benevolent. Malicious nodes usually affect the normal operations of the sensor network adversely, by injecting duplicate packets, forwarding modified or distorted packets, sensing and sending wrong data etc. Due to these malicious nodes, most of the data and control packets get forwarded through a specified node or nodes causing enormous congestion which ultimately followed by the disintegration of the Wireless Sensor Network. Sometimes, the nodes may even process data arbitrarily and can try to communicate with other nodes continuously as long as their batteries are not completely depleted. Presence of a large number of malicious nodes make the Wireless Sensor Network prone to various types of attacks like Wormhole, Sybil, Sinkhole, Denial of Services and Hello flood attacks. Other variants of attacks are tampering, jamming, collision, de-synchronization, traffic analysis and eavesdropping [TKD10], [CKD03]. In most of the cases, Wireless Sensor Networks are highly populated and it becomes extremely difficult to monitor the performance of each and every individual nodes. Hence it becomes imperative to introduce energy aware as well as reliable routing protocols with an additional characteristic of sharing the traffic among the different

reliable paths. So it can be rightly concluded that congestion control and trust based routing are actually interlinked as both have a mutual interdependence. In this chapter a new Genetic Algorithm (GA) inspired load balancing protocol for congestion control in Wireless Sensor Networks using Trust based routing framework (GACCTR) has been proposed. It utilizes the core concepts and fundamental advantages given by GAs for distributing the traffic on the basis of reliability of the nodes. Simplicity and easy implementation are the main factors of selecting GA. As expected, the results show marked improvements in various network characteristics and properties over the existing protocols as mentioned in Section 4.2

The rest of the chapter is organized as follows: Section 4.2 provides an account of some existing literature related to the proposed work. Section 4.3 contains the proposed work along with the major concepts used in the scheme. Section 4.4 represents the simulation results and finally, the chapter is summarized in Section 4.5.

4.2 Related Works

Trust based congestion control in Wireless Sensor Networks is a new concept and has not been reported in literature to a great extent. However, some works based on traditional approach of congestion control for Wireless Sensor Networks are found. Wan et al. [CYW03] have proposed a COngestion Detection and Avoidance method (CODA) for sensor networks in which open loop hop by hop back pressure and closed loop multisource regulation are used. CODA detects congestion by periodically sampling the channel load and current buffer occupancy. Hull et al. [BHK04] integrate three complimentary congestion control strategies that span different layers of the traditional protocol stack: hop by hop flow control, rate limiting and a prioritized MAC protocol. In Event-to-Sink Reliable Transport (ESRT) protocol [YSO03], both congestion detection and reliability level are estimated at the sink. ESRT places interest on events, not on individual pieces of data. Detection of events is related to the number of packets received during a specific interval. Sensor nodes must listen to sink broadcast at the end of each decision interval and updated their reporting rates. They monitor their buffer queues and inform the sink if overflow occurs. The major limitations of ESRT are only single hop operation support and pushing all complexity to the sink. The key idea of Pump Slowly Fetch Quickly (PSFQ) protocol [CYW05] for avoiding congestion is to slow down the

distribution (Pump Slowly) and to recover quick error (Fetch Quickly). The drawback of PSFQ [CTW05] includes several timers setting, highly specialized parameter tuning and complicated internal operations.

All the above works do not consider the presence of malicious nodes in the network. The inexpensive sensor nodes are typically prone to failure. These malfunctioning nodes in the network increase transit traffic and latency by diffusing useless packets. Trust based approach eliminates the malicious nodes from the routing path and thereby reduce the network congestion. Although implementation of trust concept in WSNs is new idea, some research papers are available for trust estimation of sensor nodes and thus computing the most trusted routing path. In ATSR algorithm [TZH10], routing decisions are taken based on geographical information as well as Total Trust (TT) value of the nodes. TT of a node is calculated by combining Direct Trust (DT) and Indirect Trust (IT) information of that particular node. In Geometric Mean based Trust Management system (GMTMS), trust values of individual nodes are estimated by geometric mean of all relevant trust metrics [SSB11]. This model has certain advantage over Momani's model [MOM08], in which trust is calculated as arithmetic sum of different parametric probabilities and may lead to serious false value. Suppose, all parameters give high trust values except only one parameter, say successful packet transmission or packet latency. As per Momani's model, a high trust value will be assigned even though packet transmission is zero or packet latency is infinite. This can be avoided in GMTMS [SSB11] by considering product or the geometric mean of the trust value of all parameters. In [SAM11], Direct Trust-dependent Link State Routing Protocol (DTLSRP) is proposed where Total Direct Trust (TDT) value is computed by GMTMS method [SSB11]. The nodes having TDT higher than a predefined application based threshold value can only participate in routing path. Link State Routing Protocol is used in DTLSRP to find out all available paths. The multiplicative route trust is computed for each path and highest route trust value is chosen as best routing path. DTLSRP performs better than other protocols like Ambient Trust Sensor Routing (ATSR) [TZH10] etc. However it does not include the Indirect Trust (IT) that is obtained from the feedback of neighboring nodes. Trust Integrated Link State Routing Protocol (TILSRP) uses both DT and IT for estimating the trust value of sensor nodes [ARM11]. It uses LSRP based on trust and finds the best trustworthy route. In FCC protocol [MZA09], Zarei et al.

propose a Fuzzy based trust estimation for congestion control in WSNs. The malfunctioning nodes are detected and isolated from routing path using fuzzy based trust estimation of the nodes. The resulting effects are some decrease in packet drop ratio and accordingly increase in packet delivery rate. FCCTF protocol [MZA10] is modified form of FCC [MZA09], in which the Threshold Trust Value (TTH) decision making is based on corresponding traffic ratio of the related region. TTH could change dynamically with increasing or decreasing form of traffic ration. By increasing TTH, more malicious nodes are detected and blocked and consequently useless packets are replaced by useful packets. Although FCCTF algorithm [MZA10] shows some improvement over FCC, still there is further scope of improvement in terms of congestion detection and control.

The fundamental theories and concepts of GA, as described in this chapter are taken from [DGB89], [UKC93], [UKC97] and [AMK05]. We also surveyed some existing GA based routing protocols for Wireless Sensor Networks. However, none of them were based to be found upon the trust values of the nodes. For example, a new GA based routing protocol is proposed in [ACS11], where network lifetime is increased from the previously existing schemes of LEACH [WRH00] and PEGASIS [SLC02]. In [GNW10], a GA based shortest path algorithm is described that replaces the famous Dijkstra's algorithm. The scheme proposed in [BRH12] describes another GA based congestion control algorithm depending upon the different traffic rates handled by the sensor nodes within the sensor network.

4.3 Proposed Work

An excellent method to control congestion in Wireless Sensor Networks is through the traffic balancing i.e. sharing of traffic among the different possible routing paths existing between the source and the sink. In this case, the word 'traffic' refers to the data or more specifically data sensed by the sensors of the Wireless Sensor Network.

The main difficulty in selecting the route for transmission of data from the source to the sink lies in the fact that the entire load gets concentrated on a single trusted link. This means that some nodes are active during the entire time duration whereas some other nodes remain completely idle. As a result both the lifetime and traffic handling capacity of that particular link diminishes significantly with time.

The selection of the path can be precisely modeled with the help of GA giving appropriate probabilities for selection of routing paths using the concept of Route Trusts [SAM11], [ARM11]. It must also be noted that in GA, one can work with a concise and effective population of individuals that allow us to select the more trusted paths even at the inception of the network and hence makes the entire network more reliable for communication.

For implementing GACCTR, Link State Routing Protocol (LSRP) is applied for finding out the routes between each pair of nodes and routing tables are formed. As a result, a particular sensor node has the knowledge of all the routes by which it can communicate data to the destination node. Usually, for a densely populated Wireless Sensor Network, a large number of paths exist between two nodes. However as a result of heterogeneous active durations of different sensor nodes due to some of them devoting relatively more time in sensing and communicating data compared to others, the residual energy (the remaining battery life) of the nodes are different.

Another important point that can be pointed out is that, no network is completely isolated from external intrusions or sensor node malfunctioning. Some of the routes deciphered at the inception of the network gradually become unreliable with time either due to improper working of the nodes or due to low packet transmission rate. Reduced battery life also decreases the overall lifetime of the sensor network. As a solution, different methods are employed for uniform consumption of energy at different points of the network in order to enhance the overall lifetime of the Wireless Sensor Network. The proposed GACCTR provides a set of routes having high fitness function values for distributing the load among the different paths, thus helping in the homogeneous power consumption throughout the network.

4.3.1 Overview of Genetic Algorithm

GA is basically probabilistic search algorithm and optimization technique based on the mechanisms of natural selection and evolution. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. Genetic Algorithm

maintains a population of coded forms for the possible solutions of the problem of interest. In the language of GA, these coded forms are known as chromosomes. In most cases binary coding is preferred for easy manipulation and operation. Each chromosome is evaluated by a function known as the fitness function which is usually called the cost function or the objective function of the corresponding search or optimization problem. At first, the initial population is randomly generated. New populations are created in subsequent generations through the four fundamental mechanisms: selection, crossover, reproduction and mutation operations. Selection is the process that determines which solutions are to be preserved and allowed to reproduce and which ones deserve to die out. The primary objective of the selection operator is to emphasize the good solutions and eliminate the bad solutions in a population while keeping the population size unchanged. Now, how to identify the good solution? A fitness function is assigned to each solution which quantifies the optimality of a solution. It is used to rank a particular solution against all the other solutions. There are different algorithms and techniques to implement Selection in GA. For example, Tournament Selection, Roulette Wheel Selection, Proportionate Selection, Rank Selection, Steady State Selection etc. In the proposed GACCTR scheme, we have considered the commonly used Roulette Wheel Selection mechanism which selects fitter individuals (parents) for crossover and mutation. The Crossover operator is used to create new solutions from the existing solutions available that causes the exchange of genetic materials between parents to produce offspring. The Mutation process incorporates new genetic traits in the offspring. The block diagram and flow chart of the Genetic Algorithm procedure are shown in Figs. 4.1 and 4.2 respectively.

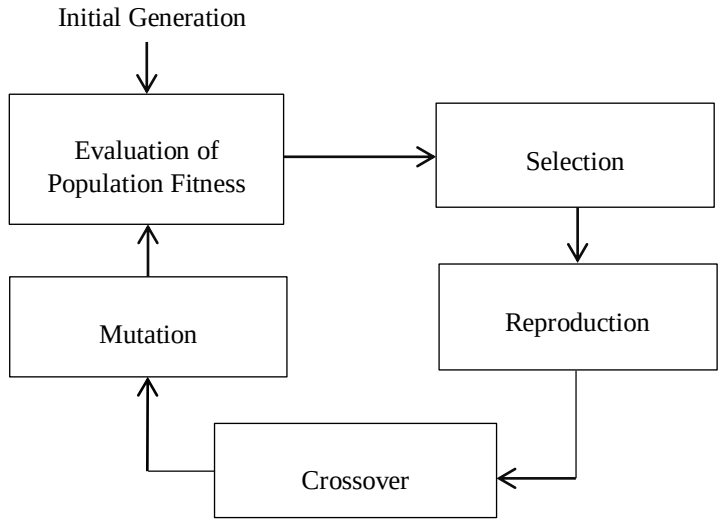


Figure 4.1: Block diagram of Genetic Algorithm procedure.

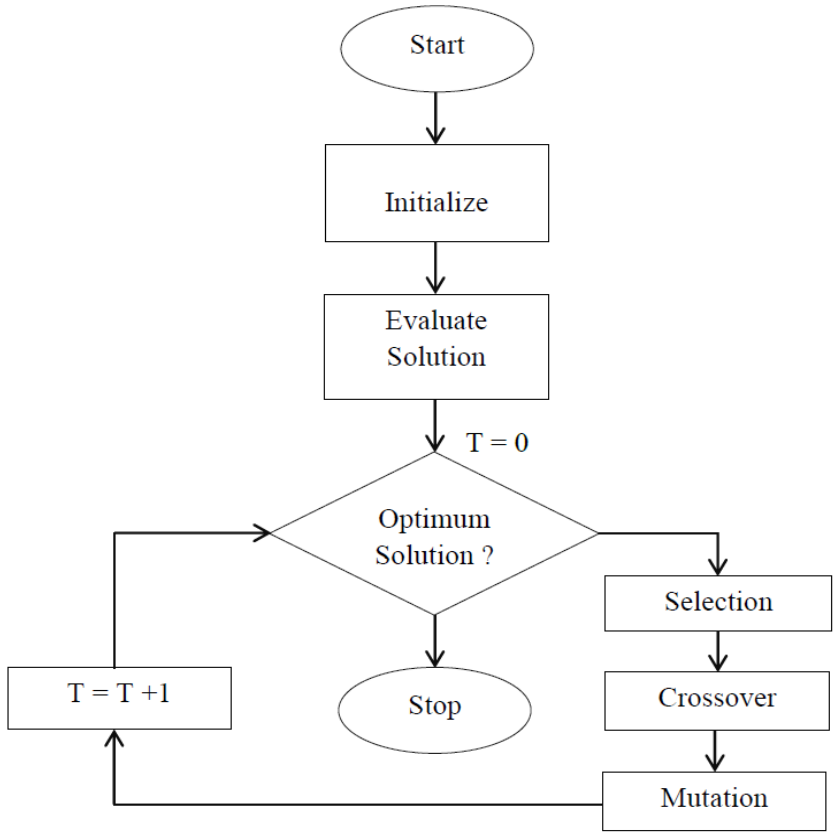


Figure 4.2: Flow chart of Genetic Algorithm.

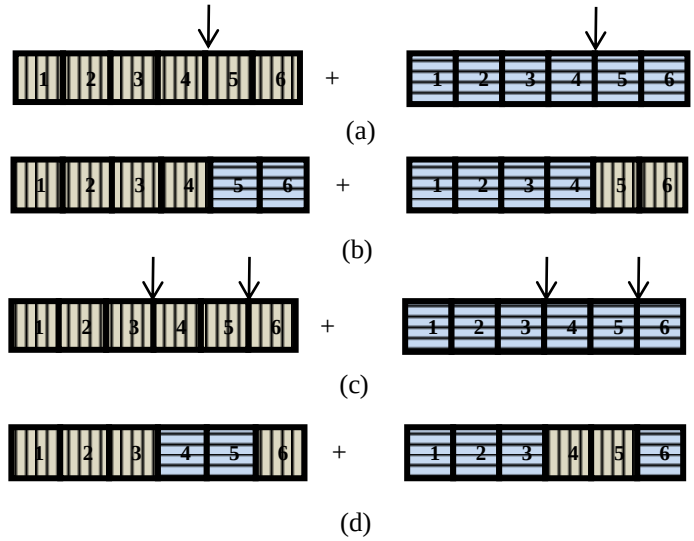


Figure 4.3: (a) Parents for one-point crossover; (b) Offspring after one-point crossover; (c) Parents for two-point crossover; (d) Offspring after two-point crossover.

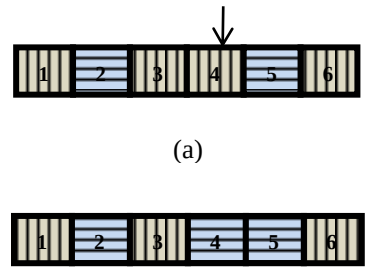


Figure 4.4: (a) Selected chromosomes for mutation; (b) Offspring after mutation.

The chromosomes representing the parents and the offspring are of length 6 are shown in Figs. 4.3 and 4.4 respectively. The points of crossover are shown by arrows. The parents for one-point crossover and the offspring after one-point crossover are shown in Figs. 4.3(a) and 4.3(b) respectively. Similarly, Figs. 4.3(c) and 4.3(d) represent the parents for two-point crossover and the offspring after two-point crossover respectively. The selected chromosome for mutation is represented in Fig. 4.4(a) with the help of an arrow and the offspring after mutation is depicted in Fig. 4.4(b).

4.3.2 Trust Based Routing Framework

In Wireless Sensor Networks, the technical meaning of the term trust is the opinion a particular sensor node possesses about another sensor node. It is actually a probability which splits the sensor nodes into two categories: benevolent and malicious. This demarcation is based upon the trust threshold (TTH) value usually taken to be 0.50 by default. Usually, trust of a sensor node can be defined in two ways: Direct Trust (DT) and Indirect Trust (IT) [MOM08], [ARM11]. In the proposed protocol, we are only interested in the Direct Trust of the sensor nodes that is, the opinion of the sensor nodes with respect to its direct one-hop neighbor. By the term one-hop neighbors of a sensor node, we mean those nodes which can communicate directly with the concerned sensor node. The Direct Trust of a node on another node is evaluated with the help of the procedure described in [SSB11]. As explored in [SSB11], the DT is calculated as the geometric mean of the different Trust Metrics as described in Chapter 2. The trust value of the node in the proposed GACCTR protocol is calculated by considering three numbers of Trust Metrics, namely, successful packet transmission, congestion status of the node and the remaining energy of the node. The congestion status of the node is obtained by estimating buffer queue length of the node. The Direct Trust of node j with respect to its one-hop neighbor i is defined as $T_{i,j}$ which is presented pictorially in Fig. 4.5 as shown below. Usually a routing path consists of a large number of sensor nodes starting from the source and terminating in the sink. For example, a routing path consisting of seven sensor nodes is shown in Fig. 4.6. The nodes are represented by the English alphabets a, b, c, d, e, f and g, where the nodes 'a' and 'g' are the source and sink node respectively. As per the definition, Direct Trust (DT) of 'b' with respect to 'a' is given as $T_{a,b}$. Similarly, DT of 'c' with respect to 'b' is given as $T_{b,c}$ and so on. An expression which utilizes DT values to depict the reliability of the entire routing path is known as the Route Trust [SAM11]. Here, it is represented as RT_{γ}, where γ is the path index.

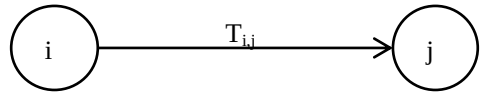


Figure 4.5: Direct Trust of a sensor node j with respect to i.

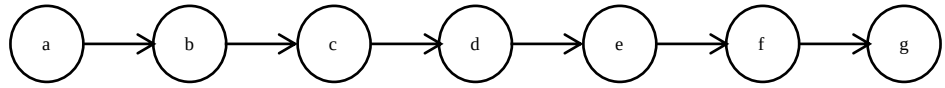


Figure 4.6: Routing path with sensor nodes ‘a’ and ‘g’ as source and sink respectively.

The value of RT for the path shown in Fig. 4.6 is given as a product of the individual Direct Trusts of the consecutive sensor nodes belonging to the path with respect to its immediate preceding node in the path. So, as shown in Fig. 4.6,

$$RT_{path} = T_{a,b} * T_{b,c} * T_{c,d} * T_{d,e} * T_{e,f} * T_{f,g} \quad \ldots . (4.1)$$

Suppose any two consecutive nodes of an arbitrary route κ are denoted by indices i and j. If the total number of sensor nodes in the path is N_S, then

$$RT_{\kappa} = \prod_{i=1}^{N_S - 1} T_{i,j=i+1} \quad \ldots . . (4.2)$$

The value of RT is an indication of the route’s reliability in terms of successful packet transmission, congestion status and remaining energy of the nodes in the route. The path is more reliable for greater value of RT. In case of equal RT values of different routes, which represent an ideal case – the shortest route is considered. This also renders the application of Dijkstra’s algorithm to be redundant.

4.3.3 Main Procedure

In the proposed GACCTR algorithm, the Eq. (4.2) has been adopted as the fitness function. This function incorporates the Direct Trust (DT) values of the sensor nodes with respect to the preceding node belonging to the different routing paths. The calculated numerical value of the fitness function determines the route through which the data is transmitted from the source to the sink. This route actually signifies the most reliable or trusted path. The selection of the route being a probabilistic process, the path of data transmission is not fixed. Moreover the DT of a node being evaluated dynamically and updated periodically itself helps in balancing the load. GA provides the most suitable way in this probabilistic modeling. The primary advantage of using GA is that, it helps to identify the best routing path in reduced number of steps and duration. In addition to this quick convergence to the most appropriate path, GA is also easy to implement and requires mediocre processing capability of the sensor nodes.

For practical implementation of the proposed GACCTR, it is considered that all the sensor nodes have prior knowledge of their corresponding routing tables, that is, the routes existing between the node itself and all other nodes within the Wireless Sensor Network. The basic routing protocol used here is the Link State Routing Protocol (LSRP). However, the requirement of Dijkstra's Algorithm for finding the shortest route from the source to the sink is redundant in the proposed protocol; because in this case, the main goal is to select the most trusted route instead of the shortest one. If the Direct Trusts of all the nodes are equal with respect to the one-hop neighbours, then the shortest possible route is chosen as the data routing path from the source to the sink, according to the proposed GACCTR protocol.

4.3.3.1 Chromosome and its Constituent Genes

The fundamental step in GA is the evaluation of the fitness function which is represented as a function of the representative genes of the chromosomes. In the proposed GACCTR, the chromosome denotes the routing path. The constituent genes depict the sensor nodes within the path and the gene indices G_0, G_1,……. G_n, represent the positions of the sensor nodes starting from the source node. Thus, the node ID stored in G_0 represents the source node or the node from where the data is generated, G_1 represents the next node after the source node to which the data is communicated directly. Similarly G_i represents the i^{th} node in the routing chain. The chromosome size (or the number of genes) L represent the maximum length of the route in terms of the number of existing nodes. Since, Wireless Sensor Networks are densely deployed and so, innumerable paths exist between the initiating node and the terminal one. Hence, it becomes imperative to impose an upper limit on the maximum length of the chromosome with the minimum length being a direct connection between the source and the sink. It is evident that the length of the routes would never be uniform. Hence to keep the length of the chromosome constant, the non-existing gene indices are padded with zeros.

Figure 4.7: L length chromosome.

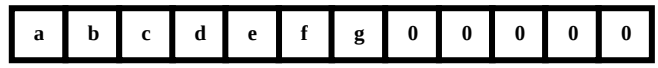


Figure 4.8: 12-length chromosome.

The pictorial view of L length chromosome is shown in Fig. 4.7, whereas Fig. 4.8 shows 12-length chromosome representation of the routing path as given in Fig. 4.6. As mentioned earlier, Eq. (4.2) represents the fitness function. It can be restated in Eq. (4.3),

$$f_i = RT_i = \prod_{i=0}^{L-2} T_{G\_i,G\_i+1} \quad \ldots . (4.3)$$

where, sensor nodes are presented as genes having a fixed source and sink. Application of the usual GA operators for initial population generation and subsequent generations through parent selection, crossover, and mutation are described in Algorithms 1, 2, 3, 4 and 5 respectively to produce fitter generations of chromosomes, which subsequently represent the more reliable routing paths. At some suitable generation, the fitness values are plotted for a Roulette Wheel Selection or Fitness Proportionate Selection [DGB89]. The fitness value is used to associate a probability of selection with each individual chromosome representing a routing path. If f_i is the fitness of individual i in the population, the probability of being selected is given by

$$p_i = \frac{f_i}{\sum_{j=1}^{Np} f_j} \quad \ldots . (4.4)$$

where, N_p is the number of individuals in the population. It is to be noted that closer the values of RTs, render equal probabilities of route selection. The justification of the roulette wheel analogy can be envisaged by imagining a roulette wheel in which a candidate solution represents a pocket on the wheel; the sizes of the pockets are proportionate to the probability of selection of the solution. Selecting M chromosomes from the population is equivalent to playing M games on the roulette wheel, as each candidate is drawn independently. So, it is evident from utilization of this rule that no route is fixed from transmission of packets from the source to the sink. The method being completely random helps to distribute the load among the different routing paths, with GA to find the most reliable ones. The pseudo codes of the algorithms in the proposed GACCTR scheme are given in algorithm 4.1, algorithm 4.2, algorithm 4.3, algorithm 4.4, algorithm 4.5 and algorithm 4.6; which corresponds to the initial population

generation algorithm, parent selection algorithm, one point crossover algorithm, repair algorithm, mutation algorithm and the main procedure respectively.

Algorithm 4.1: Initial Population Generation Algorithm

Inputs: g <= Initial population size that is the number of individuals in a generation;
N <= Number of nodes in the WSN; So <= Source Node; Si <= Sink Node;
D <= Routing tables along with Route Trust (RT) values available in each of the nodes present
in the WSN;
Output: Set of routing paths from a particular Source to a particular Sink;
Procedure generate_initial_population ( );
temp2 = So
Loop1: ***for*** j=1:g
Set trust values of nodes once crossed to zero in data table
Loop2: ***for*** i=1:N
m=1;
Loop3: ***for*** k=1: N
If (data1 (temp2, k) > 0
p2 (m) = k; % storing the node values that can be trusted in array p2
m=m+1; % incrementing the counter
end **Loop3**
ran_num=randi(length(p2)); % generating random number of length of array p2
temp1 = p2(ran_num); % storing element from p2 in temp1
If (temp2 is equal to Si)
Terminate **loop2**
Endif
If (Si not reached within N repetitions)
Set generation to zero and start over loop2
End if
End loop2
temp2=So; % setting temp node to source node value for next generation starting
data1=data; % reload trust value chart
End Loop 1
Display initially generated population
End Procedure

Algorithm 4.2: Parent Selection Algorithm

```
Inputs: population ,N( no. of nodes)
Outputs: selected_pop,trustofpopulation(w)
Procedure Selection();
L= no. of members in population
for  j=1:L
        for i=1:(N-1)
        temp1=population(j,i) ;        /*Store first node in temp1*/
   temp2=population(j,(i+1));      /*store second node in temp2*/
   if(temp2 < 0.01)        /* if destination node value is zero break*/
      break;
     endif
   w(j) = w(j)*data(temp1,temp2);  /* product of trust values of each path stored in
weight matrix*/
   endfor
for  i =1:L
        r(i) = w(i) + prev;     /* adding the previous value of weights to obtain a
continuous range from 0 to 1*/
      prev = prev + w(i);
end
        end
Display pie diagram     /*displays the equivalent roulette wheel*/
for  i=1:L
if ( r(i) < roll <r(i+1) )   /* if random number lies in a specified range select that node
      selected(j) = i;
      break;
     endif
endfor
for i=1:l
      selected_pop(i,:) = population(selected(i),:);  /*selected population*/
end
End for
End Procedure
```

Algorithm 4.3: One-point Crossover Algorithm

Inputs : Selectedpop; no. of nodes ***Outputs :*** Crossoverpop; ***Procedure*** Crossover();
L = no. of members in population; a= number of non-zero element in first parent;
b= number of non-zero element in second parent;
crossover point = ceil((a+b)/4) /* Round off the crossover point to nearest integer */
for j=1:crossover point
tempstore(j) = firstmate(1,j); /* store the elements of first parent from starting to crossover point in temp matix*/
endfor
for j=1:crossover point
firstmate(1,j) = secondmate(1,j); /* rewrite elements of first parent from start to crossover point with elements of second matrix*/
secondmate(1,j) = tempstore(j); /*rewrite elements of second matrix with that of temp matrix*/
endfor
Send paths for repairing to 'Repair' Program
newgen(i,:) = firstmate(1,:); newgen((i+1),:) = secondmate(1,:); newgenfit = fitness of crossover generation; parentfit = fitness of parent generation
for j=1:2:L
temp(1) = newgenfit(j);
temp(2) = newgenfit((j+1)); /*store the fitness value of new generation of two offsprings*/
temp(3) = parentfit(j,:); /* store the fitness value of parentgen*/
temp(4) = parentfit((j+1); tempgen(1,:) = newgen(j,:); /* store offspring node path*/
tempgen(2,:) = newgen((j+1),:); tempgen(3,:) = selected_pop(j,:);
tempgen(4,:) = selected_pop((j+1),:); /* store the parent node path*/
[~, i1] = max(temp); /* obtain index position of the path having greatest trust value */
temp(i1) = 0; /* set trust value of the path having highest fitness to zero*/
[~, i2] = max(temp); /* obtain index position of path having next best fitness*/
finalgen(j,:) = tempgen(i1,:); /* store the path having highest fitness*/
finalgen((j+1),:) = tempgen(i2,:); /*store the path having next best fitness*/
Endfor
End Procedure

Algorithm 4.4: Repair Algorithm

Inputs: row_mat; ***Outputs :*** repaired_mat;
Procedure Repair ();
N1 = index position of last non-zero element; N= number of nodes;
Data = matrix containing trust values;
for i=1:N1
Element = row_mat(:,j);
Comparison = find(row_mat==element);
If (length(comparion) > 1)
Firstindex = Comparison(1);
Lastindex = Comparison(2);
for i=1:(Firstindex-Lastindex)
Replace by portion of node non-overlapping portion
Endfor
Replace the portion of node after the destination node with zero
Endif
End
for i = 1: (N-1)
temp1 = row_mat(1,i); temp2 = row_mat(1,(i+1));
if (temp2>0);
if (data(temp1,temp2) == 0) /*Discontinuity*/
pos = find(row_mat==temp2);
initialrow_mat = row_mat(1,1:(pos-1));
/* store the part of the row which is continuous in a separate matrix*/
Send to 'Initialize program' the node just before discontinuity as source and the array initialrow_mat /*initialrow_mat is sent to prevent overlapping*/

Returns path modifiedrow
Row_mat = [initialrow_mat modifiedrow]
Endif
Endfor
End Procedure

Algorithm 4.5: Mutation Algorithm

Inputs: pop,N

Output: mutpop

Procedure Mutation():

G = number of generations in population;

N = number of nodes in grid;

Loop : R = randi(g) /*randomly select a population*/

a = number of nodes in selected generation;

pos = randi(a) /*randomly select the node position to be mutated*/

prenode = determine the node before the node to be mutated;

postnode = determine the node after the node to be mutated;

determine possible routes from prenode to postnode with single intermediate node

if (fitness of new path) < (fitness of old path)

Repeat loop

endif

store new path I mutpop

End Loop

End Procedure

Algorithm 4.6: Main Procedure

Inputs : Number of nodes, generations, sourcenode, sinknode

Outputs : fittest path route, plot of fittest path against iteration, Roulette wheel after each iteration

Set zeta, termination iteration(iter), convergence iteration(t)

Main Algorithm

Generate Initial population; Store the fittest population member;

Set initialpopulation to currentpopulation

For K=1:iter

Send current population to Selection program and obtain selectedpop

Send selectedpop to Crossover program and obtain newpop

If ((Mutation_probabilty)*k) >1)

Send currentpop to Mutation program and obtain mutatedpop

End if

Store fittest value of newpop

If(fittest value doesn't vary by small threshold(zeta) for last t iterations)

Break

End if

set newpop to current pop

End Repeat

Plot fittest value against iterations

4.4 Simulation Results

Performance evaluation of the proposed GA based GACCTR algorithm has been tested using extensive simulations. The algorithm is superior in terms of computing speed and efficiency (i.e., consumption of less energy) compared to the conventionally available trust based route detection techniques [TZH10], [SAM11], [ARM11] which involve checking of all possible routes from the source node to the sink node.

Simulations are conducted by assuming square grid networks as shown in Fig. 4.9, having n^2 sensor nodes where n is the number of nodes in a single row or column. If it is assumed that the total number of sensor nodes is N, then $N=n^2$. For simplicity, it is

assumed that a sensor node communicates only with its horizontal and vertical neighbors but not with the diagonal ones. This can be viewed from another perspective that the distance between the centres of horizontal or vertical adjacent nodes is the radius of communication. This prevents the diagonal nodes in communicating with each other. Fig. 4.9 represents the grid of sensor networks for n= 3 and N = 9, in which the source node and sink node are denoted by green and pink colour respectively. As mentioned earlier, only Direct Trusts of the sensor nodes are considered. The trust values are included in the routing table after trust evaluation. An example of this is given in Table 4.1, which is in accordance to the sensor network as shown in Fig. 4.9.

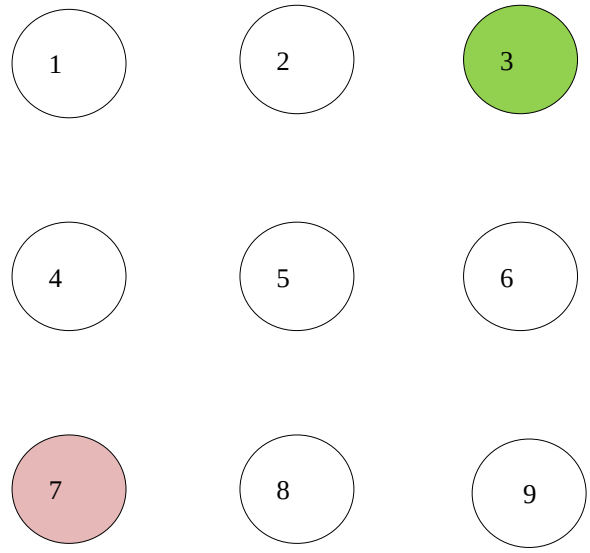


Figure 4.9: Grid of sensor networks for n=3 and N=9.

Table 4.1: List of variables

Node whose trust is calculated / Node on which trust is calculated	1	2	3	4	5	6	7	8	9
1		0.5		0.75					
2	0.85		0.65		0.75				
3		0.65				0.7			
4	0.6				0.45		0.55		
5		0.65		0.6		0.5		0.5	
6			0.4		0.5				0.6
7				0.45				0.5	
8					0.45		0.35		0.45
9						0.6		0.5	

In the simulation experiments, it is assumed that the trust values are incorporated within the routing table which is constructed on applying LSRP. The proposed algorithm considers the dynamically varying trust values. It can also deal with node failure by setting corresponding trust values associated with that node to zero.

A grid network [as in Fig. 4.9] is assumed for distribution of nodes and the algorithm is considered to be capable of determining the route with maximum trust value from any specified source to any specified sink. It is to be noted that the grid network is assumed for the sake of simplicity and the proposed protocol is valid as long as the formation of the routing table is possible on application of LSRP as shown in Table 4.2.

Simulation of the proposed algorithm include the following

1) The fitness value of the fittest chromosome occurring in the present population with progressive iterations by varying the total number of nodes and the number of generations.

2) The number of iterations required to attain the fittest chromosome by:
   2.1) Keeping N, Source and Sink fixed and varying number of generations.

2.2) Keeping number of generations, source and sink fixed and varying N.

2.3) N and number of generations fixed and the distance between source and sink is varied (in terms of the number of minimum hops)

3) Network lifetime of the entire network compared to some already existing algorithms by varying the percentage of malicious nodes.

4) Probabilities of transmission through different paths in the form a pie-chart simulating the Roulette wheel.

Simulation (1)

It represents the number of iterations required to obtain fittest fitness value of the chromosomes currently present within the population with varying number of generations and total number of nodes. The graph shown in Fig. 4.10 presents the results of simulation 1. It can be appropriately concluded from Fig. 4.10 that the number of iterations required for same population size are greater for larger number of sensor nodes barring one or two exceptions.

Simulation (2.1)

It represents the number of iterations required to attain the fittest chromosome by keeping N, source and sink fixed and varying number of generations. It determines how fast the algorithm converges to the most trusted route as the number of generations is varied. Varying the number of generations affects the number of possible routes between source and sink in the initial population and subsequent generations. The result of simulation (2.1) is shown in Fig. 4.11; as the number of generation is increased, there are improved chances of getting trusted routes by Crossover and also there is a trivial possibility of obtaining the best possible path in the initial generation itself. It has also been witnessed that decreasing the number of generation below a certain threshold causes the algorithm to converge on a local maxima of trusted values.

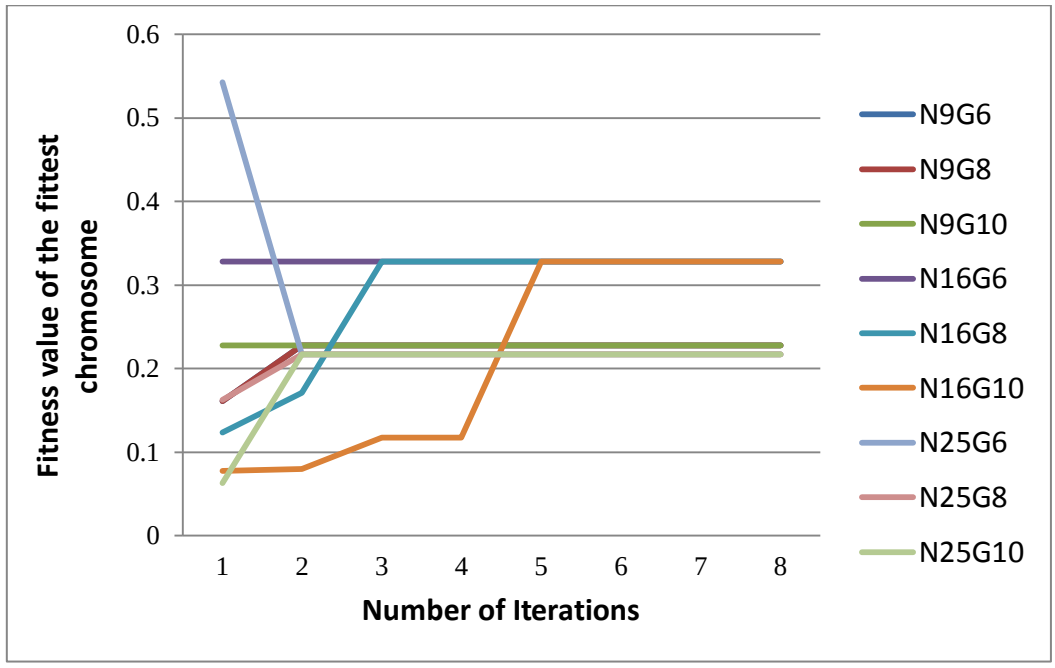


Figure 4.10: Number of iterations required to get the fittest member.

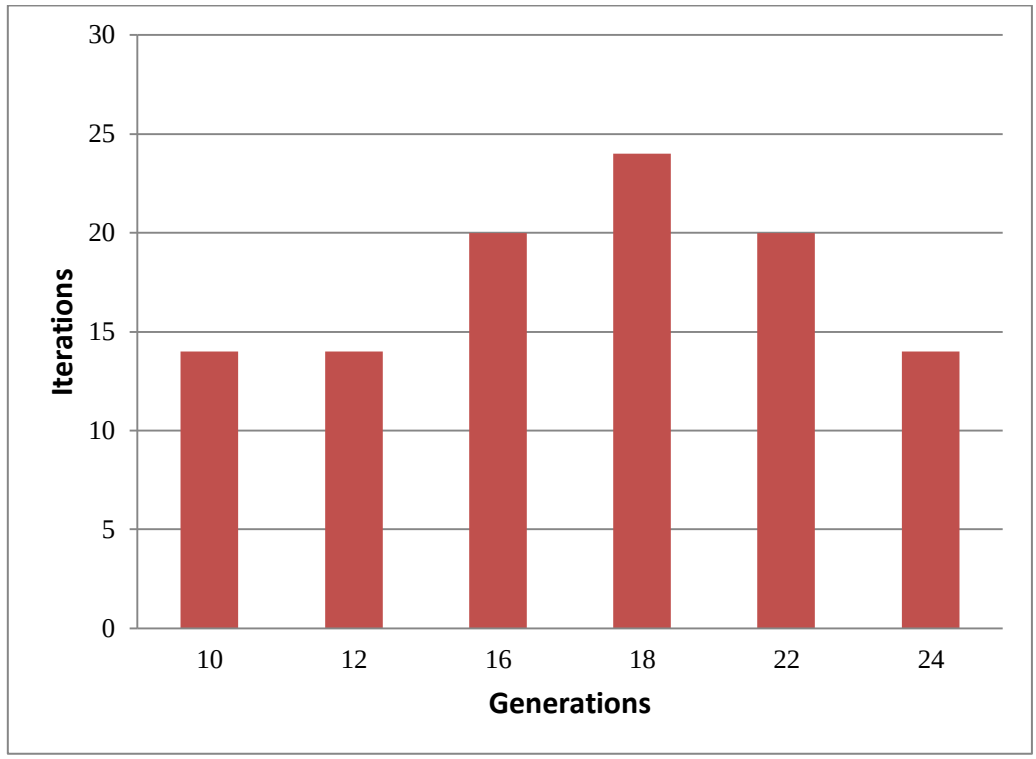


Figure 4.11: Result of simulation (2.1).

Simulation (2.2)

It determines how the complexities (time complexity in terms of iterations) vary as the number of nodes in grid changes. It is ensured that the number of hops between source and sink remain same even though number of nodes in grid is varied. The result of simulation (2.2) is presented by blue coloured graph shown in Fig. 4.12; the thin black and red lines are the approximation graphs of second and fourth order polynomial respectively, where the experimental graph can be fitted to get the prediction of the number of iterations for a certain number of nodes.

Simulation (2.3)

It provides the number of iterations that are needed to obtain the most trusted route, as the number of hops between source and sink is varied. The results of simulation (2.3) are given in Fig. 4.13.

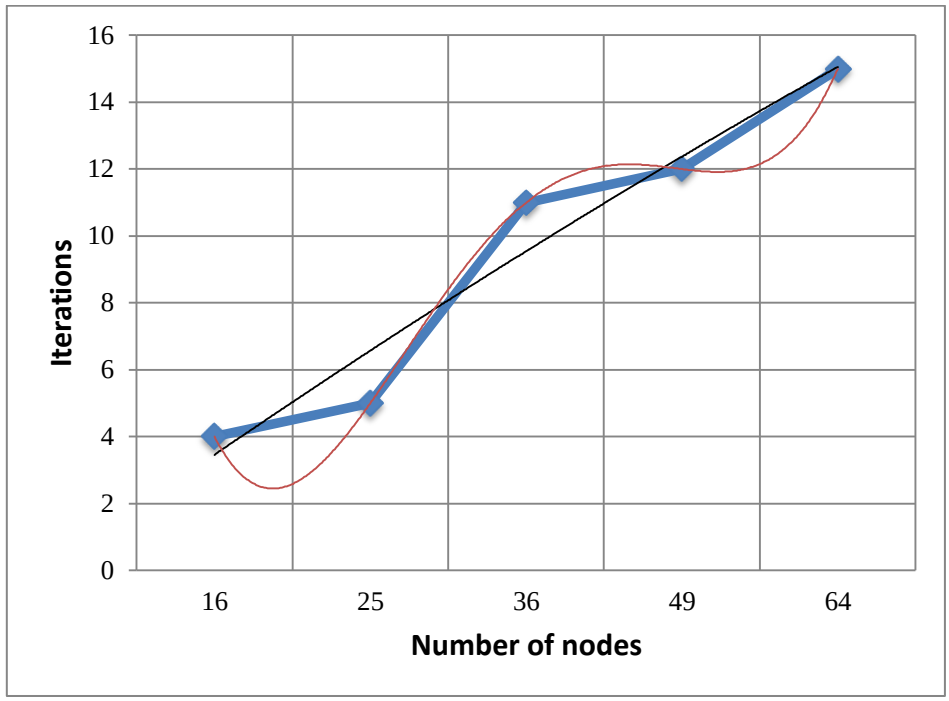


Figure 4.12: Result of Simulation (2.2).

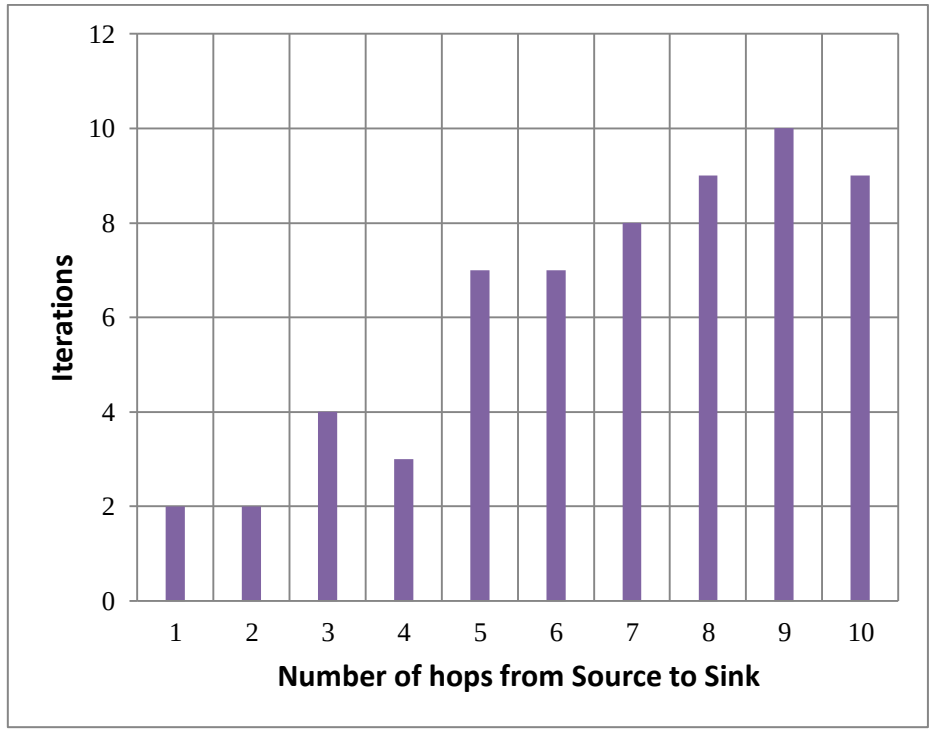


Figure 4.13: Result of Simulation (2.3).

Simulation (3)

Simulation (3) is actually implemented in IRIS motes on the TinyOS platform. A network of 100 homogeneous sensor nodes is deployed at the site of experiment. The source and sink are assumed to be fixed and the time duration is calculated till which at least a single path existed between the source and sink. This is termed as the network lifetime. In the proposed GACCTR algorithm, the network lifetime is defined as the time interval starting from the inception of the network until the absence of any path between a particular source and sink. In this case, the initial Direct Trusts of all the sensor nodes and trust threshold (TTH) value are both initialized to 0.5. A comparative study of the proposed GACCTR protocol with other three existing protocols viz. ATSR [TZH10], DTLSRP [SAM11] and TILSRP [ARM11] are shown in Fig. 4.14, where network lifetimes are plotted with the percentage of malicious nodes. Similarly, comparison of the proposed GACCTR scheme with the existing protocols is given in Fig. 4.15, in which the percentages of successful packet transmission versus percentages of malicious nodes are shown.

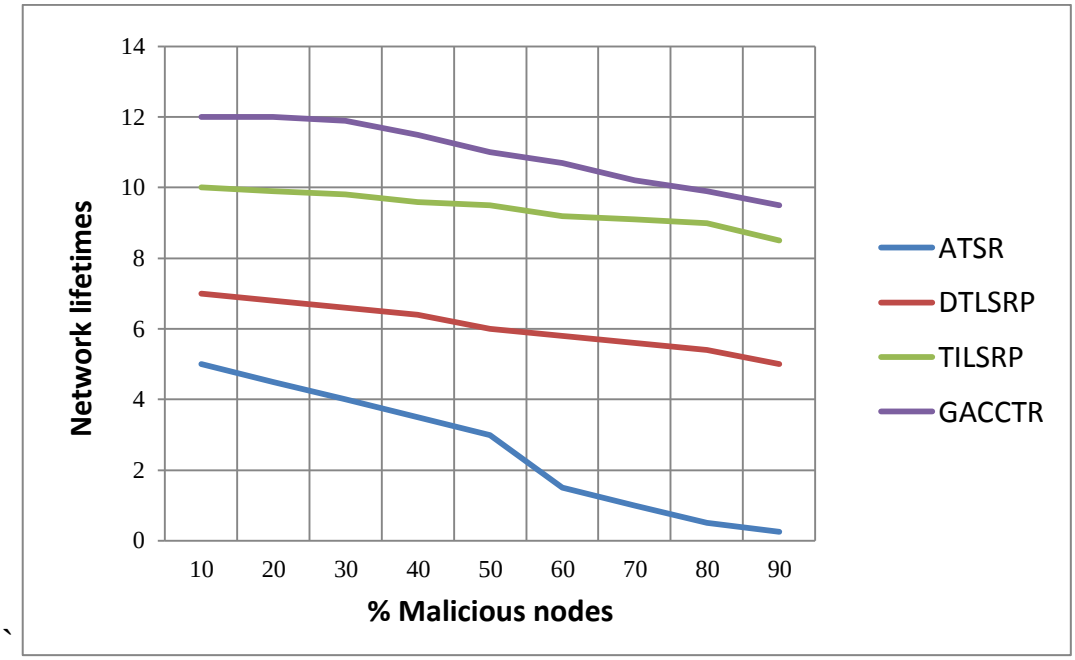


Figure 4.14: Comparison of network lifetime.

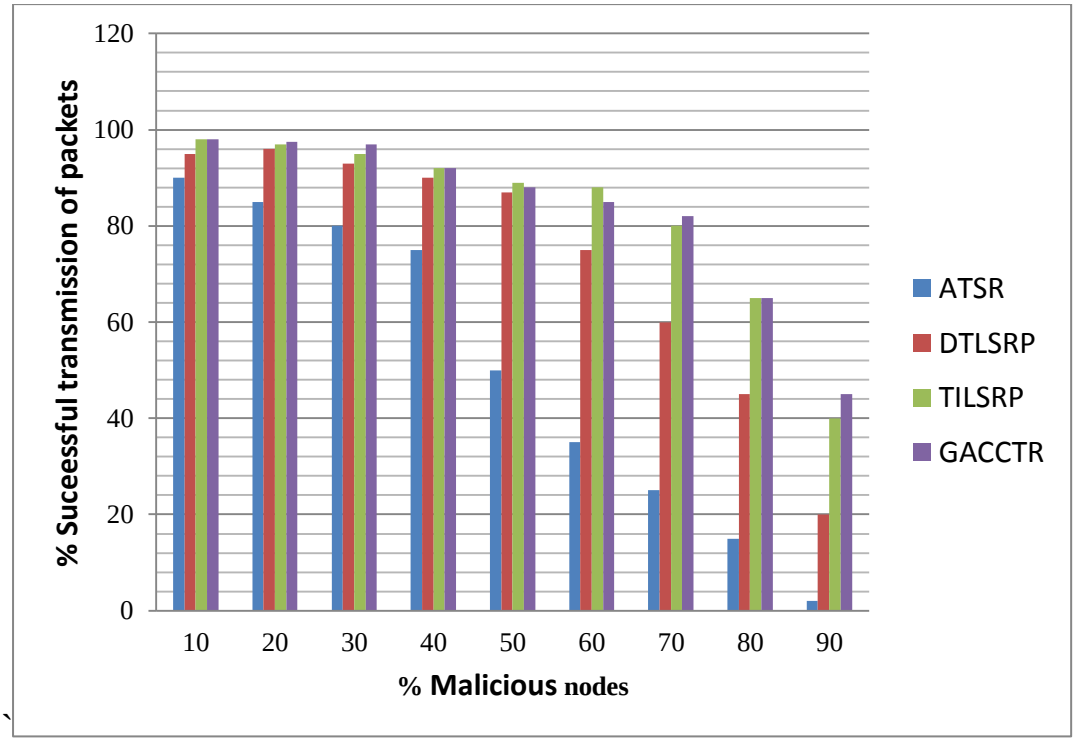


Figure 4.15: Comparison of successful packet transmission.

Simulation (4)

Simulation (4) shows the probabilities of transmission through 8 different paths (the population size is fixed at 8), in the form of pie-chart simulating the Roulette Wheel. Figs. 4.16, 4.17 and 4.18 represent different probabilities at three distinct time instants – at the start, at some intermediate instant and after the saturation, respectively. At saturation, all the paths have almost equal probabilities as shown in Fig. 4.18, this shows that the data can be transmitted through the eight paths with equal probability reducing congestion.

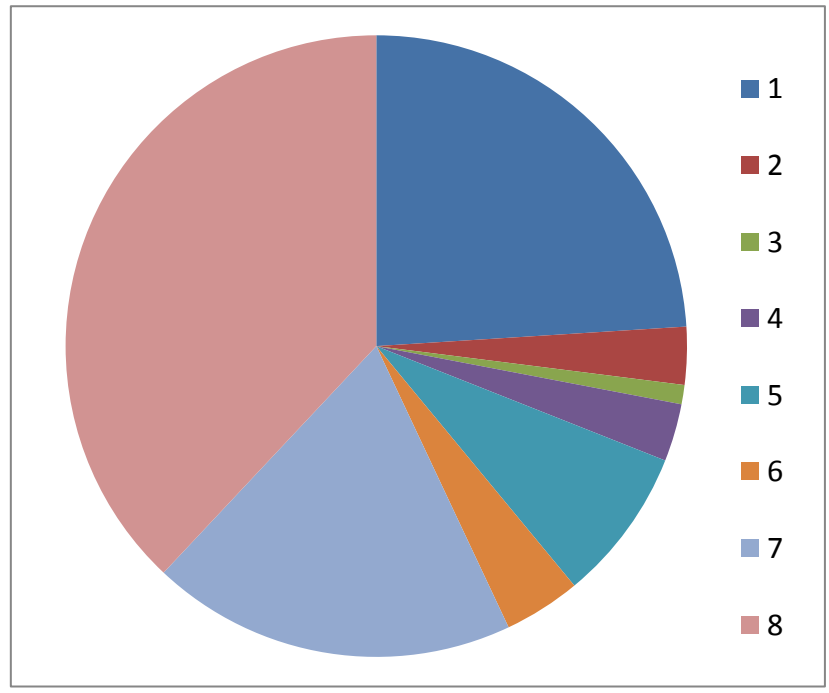


Figure 4.16: Path probabilities at the start.

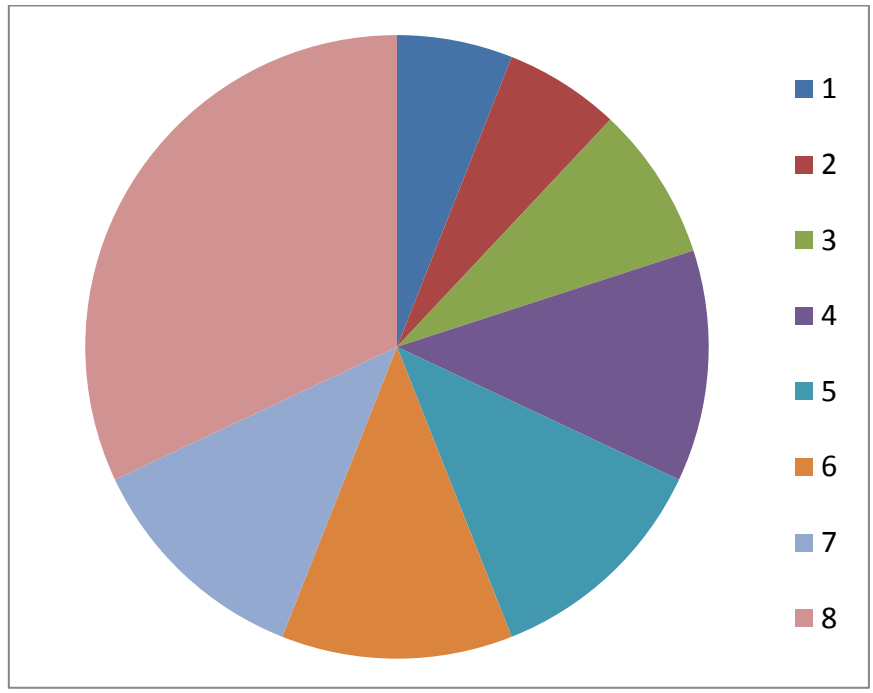


Figure 4.17: Path probabilities at some intermediate instant.

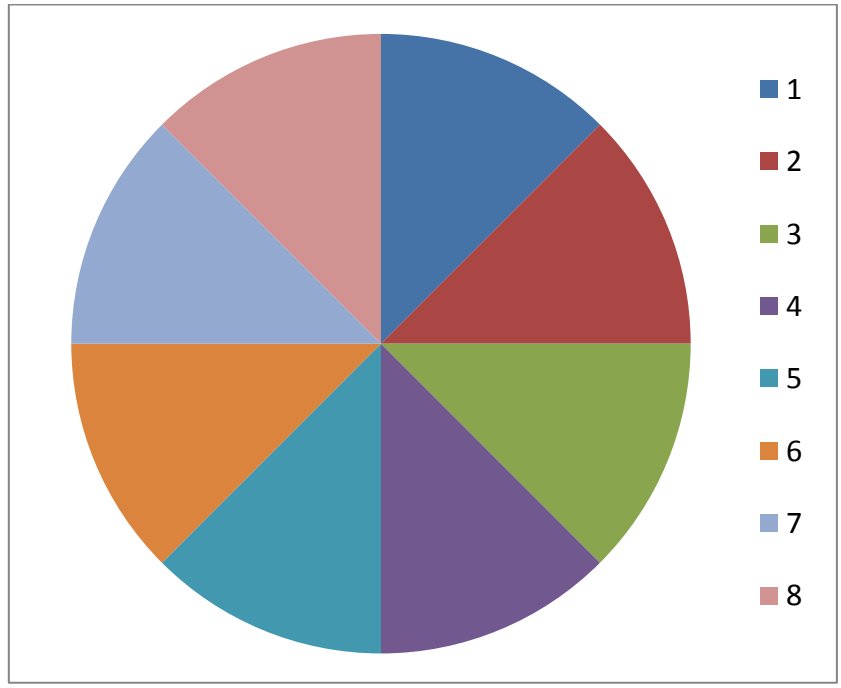


Figure 4.18: Path probabilities after saturation.

The interpretation of Figs. 4.16, 4.17 and 4.18 can be comprehended like this: let a random number be chosen between 0.000 and 1.000 and the random number helps to pick any of the colored regions, when expressed as an ordered fraction of the whole circular region (taken to be 1). Number of generations is fixed at 8. Here, different colours as denoted by integers from 1 to 8 represent eight alternate paths from source to sink.

4.5 Summary

In this chapter, GACCTR algorithm is proposed, where selection of path is modeled by GA inspired load balancing for WSNs. The Simulation results as given in Section 4.4 show the performance of the proposed GACCTR protocol, when made to run through a variety of experiments under different conditions. Almost all of them shows better performance with GACCTR protocol when compared to its peers. As shown in Fig. 4.14, the network lifetime of the proposed GACCTR is plotted with that of the other similar existing protocols by increasing the percentage of malicious nodes in the network from 10% to 90%. The mean value of the percentage increment in network lifetime for GACCTR with respect to the existing protocols TILSRP and DTLSRP are calculated and it shows almost 32% of enhancement in network lifetime in GACCTR. Similarly, comparison of successful packet transmission for different protocols is represented in Fig. 4.15. The mean value of the difference in packet transmission in GACCTR with respect to the protocols TILSRP and DTLSRP are calculated which show nearly about 17% of increment in successful packet transmission in case of GACCTR. Thus, it is concluded that the proposed GACCTR is a better alternative than the existing ones as mentioned in Section 4.2. However, selection of routing path may also be modeled by other optimization techniques, like Ant Colony Optimization or Particle Swarm Optimization, which can be taken as the future works.

CHAPTER 5

Fuzzy Algorithm for Trust Based Congestion Control in Wireless Multimedia Sensor Networks

5.1 Introduction

Wireless Multimedia Sensor Networks (WMSNs) is a new, emerging field of Wireless Sensor Networks (WSNs) which are being developed in recent years for multimedia applications like video surveillance, video traffic control systems, health monitoring and industrial process control. They contain sensor nodes having low cost CMOS cameras, microphones and other sensor devices for retrieving video and audio streams, still images and scalar sensor data from the physical environment [IFA07]. Similar to WSNs, WMSNs are resource constrained in terms of battery power, memory space, computational capability and communication bandwidth. The densely populated, randomly deployed sensor nodes in WMSNs generate large volume of high bit rate multimedia data which are either of snap shot type or of streaming data type. Snap shot type multimedia data is bursty in nature that is obtained through event triggered observation in a short time period whereas streaming multimedia content is generated over longer time period and requires sustained continuous delivery of information

[MHY08]. All these multimedia data may create network congestion in the upstream direction from the source node to the base station (BS), if the data processing and transmission speed lag behind the speed of the incoming traffic. Congestion creates buffer overflow, increased latency, packet drops, wastage of energy, deterioration of QoS and lowering the network throughput. Even in the worst case of severe congestion, the entire operation of the network may collapse. So, congestion detection and congestion control of a network is absolute necessity by any means. Since the traditional congestion control mechanisms are not suitable for resource constraint WSNs, the challenge is to design energy efficient and reliable congestion controlled transport mechanisms for WSNs as well as for WMSNs, to optimize network resources and QoS requirements. Although some congestion control algorithms for WSNs are available in the literature, most of them do not consider the impact of security attack and the role of malicious nodes on network congestion. Generally, low cost sensor nodes are prone to failure and sometimes behave as faulty nodes in due course of time. The faulty nodes are known as the malicious nodes, when they behave intelligently to lead to several security threats in the sensor networks without being detected easily. Some security attacks as described in [TKD10] and [CKD03] have direct impact on the network congestion. For example, HELLO flood attacks, Jamming attacks, Sybil attacks and Node replication attacks aggravate congestion by flooding the network with fake messages, jamming intermittently, retransmitting same message several times and creating false node identification respectively. The resulting effect is additional computation and communication overhead and an increase in energy consumption which effectively reduces network lifetime**.**

Unfortunately, traditional congestion control protocols do not consider the impact of the malicious nodes in network congestion. In this chapter, two new Fuzzy based congestion control algorithms, TFCC (Trust-based Fuzzy algorithm for Congestion Control) and TCEER (Trust-integrated Congestion-aware Energy Efficient Routing) have been proposed, where the malicious nodes are detected and isolated from the data routing path, using the concept of trust. The basics of Fuzzy Logic (FL) and Fuzzy Logic Controller (FLC) are briefly described in Chapter 3 (Section 3.4.1.1). In the first algorithm TFCC [ACA12], Trust Metric of the sensor nodes are derived by using two stage Fuzzy Logic Controller schemes. The traffic flow from the source node to the base station is optimized

by implementing Link State Routing Protocol. In the second algorithm TCEER [ACS15], the sensor nodes forward data packets to its one hop neighbouring node in radio communication range, on the basis of the trust value, congestion status, distance of the node from the base station and the remaining energy of the node.

The rest of the chapter is organized as follows: The related works are stated in Section 5.2. The proposed TFCC algorithm and its simulation results are described in Section 5.3. Section 5.4 presents the proposed TCEER algorithm and its simulation results. Finally, the chapter is summarized in Section 5.5.

5.2 Related Works

Congestion and security attacks are common phenomena in resource constrained WSNs, especially for WMSNs, where a large volume of high bit rate multimedia data needs to be managed by the network. Many novel algorithms have been proposed in the literature for energy efficient routing in WSNs. But most of them do not consider the practical problems that arise due to the presence of malicious nodes and congestion in the network. Trust based congestion aware routing in WSNs is a new research topic and has not been addressed in the literature to a great extent. T-LEACH [JHJ09] is the improved version of the popularly known data gathering algorithm LEACH [WRH00], which minimizes the number of cluster head selection and thus extends the lifetime of the network, compared to that of other similar protocols. But, scope of further improvement in network lifetime is there, since T-LEACH [JHJ09] does not consider the existence of malicious nodes and network congestion. Routing protocols equipped with trust management have been described in [SSB11], [STP14], [ARM11], [ACA12] and [TZH10] respectively. However, the problem of network congestion has not been addressed here. CODA [CYW03] has proposed energy efficient congestion detection and avoidance scheme for sensor networks that are comprised of receiver based congestion detection, open loop hop-by-hop back pressure and closed loop multisource regulation. In ESRT [YSO03], a transport solution is developed to achieve reliable event detection with minimum energy expenditure and congestion resolution functionality. In PSFQ [CYW02], data is distributed from a source node by pacing data at a relatively slow speed called “pump slowly” and allowing the nodes that experience data loss to recover missing segments from their local immediate neighbors aggressively called “fetch

quickly". PCCP [CWK06] is a hop by hop congestion control algorithm in the upstream direction, in which node priority index is considered and node congestion is measured by using packet inter arrival time and service time. A fuzzy based congestion control for WMSNs are proposed in [CSZ14], where some packets of the frames are dropped and as a result, frames are being transmitted to the BS with lower but acceptable quality. Shahin Mahdizabeh Aghdam et al. [SMA13] have proposed a congestion control algorithm for WMSNs, in which congestion is avoided at the source by adjusting the sending rate and by distributing the departing packets from the source. In addition, the intermediate nodes monitor the queue length to detect congestion. All the above mentioned protocols have discussed congestion control and have tried to improve network performances. However, aggravation of network congestion by faulty behavior of the malicious nodes has not been discussed in these protocols by any means. The congestion and trust are both discussed in [MZA09] and [MZA10]. Zarei *et al.* [MZA09] have proposed a fuzzy logic based trust estimation scheme for congestion control in WSNs. [MZA10] is basically a modification of the protocol discussed in [MZA09], in which Trust Threshold value is used for decision making. In the existing work, our objective is to develop new trust integrated congestion aware data routing algorithm for WMSNs which will be energy efficient and capable to exhibit promising improvement in network performance compared to existing similar data routing algorithms.

5.3 Proposed Algorithm 1: TFCC

In the proposed model, the misbehavior of the sensor nodes is identified using the concept of trust. The malicious nodes are thus isolated and blocked from the data communication. The trust of an individual sensor node with respect to all of its one-hop neighbors is computed from TMs using a Fuzzy Inferencing Scheme (FIS). If the trust value of a node of a particular one hop neighboring node exceeds a predefined Trust Threshold (T_{TH}), then the node is referred to benevolent node or trusted node with respect to that particular node [ARM11]. The link between the two nodes is called trusted link. Data packet transmissions are possible only through the trusted link. The same node may not be act as trusted node with respect to the other one hop neighboring node. In this case, trust value of the node on that particular neighboring node is below T_{TH} and the link between these two nodes is untrusted link. Data packets could not be transmitted through the untrusted link. The nodes without any trusted link are referred to as malicious or

faulty nodes. All the malicious nodes are blocked from the network so that they could not take part in data packet routing. In next step, the congestion level of each trusted node is computed from the buffer queue length of the node. The congestion-trust metric is formed using the proposed inference engine. The available routes from source to sink are obtained by applying the Link State Routing Protocol (LSRP) [LLP07], [AST89]. A Fuzzy control mechanism is proposed to prevent congestion of the sensor nodes by regulating the rate of traffic flow on the basis of the priority of the traffic source.

5.3.1 TFCC: Step One

In this step, the misbehavior of the sensor nodes is detected using the concept of trust. The trust value of each individual sensor node is estimated by using a Fuzzy algorithm. The Trust Threshold (T_{TH}) value is assigned, which is application dependent. As T_{TH} value is increased, the network security is also increased. The malicious nodes are isolated and blocked for further data packet transmission and finally we apply Link State Routing Protocol for getting the available routes. The pictorial depiction of the algorithm (step one) is shown in Fig. 5.1.

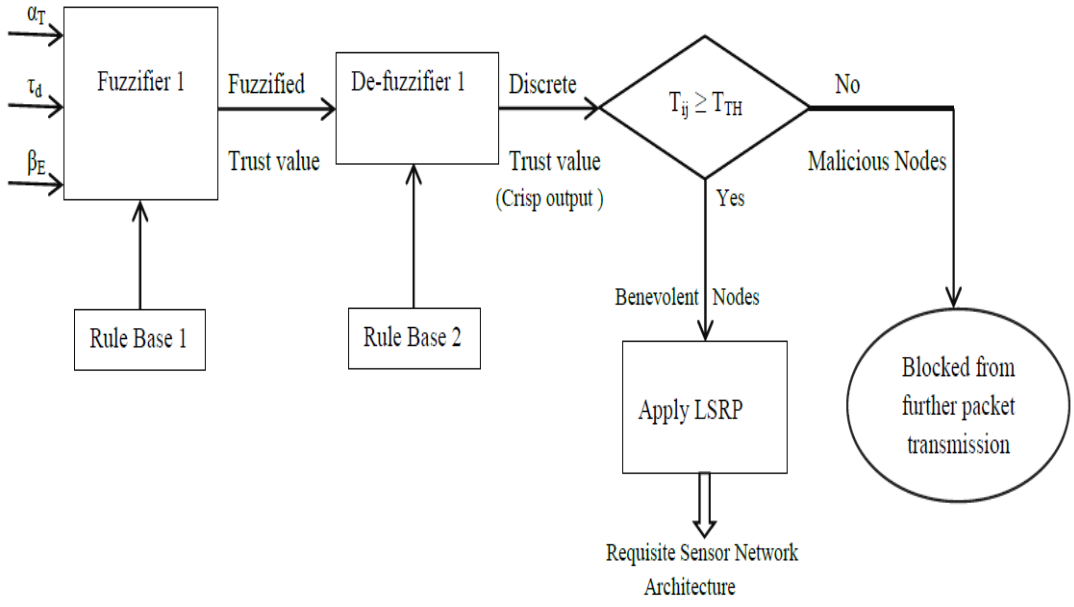


Figure 5.1: Flow diagram of TFCC (step one).

5.3.1.1 Computation of Trust Value

Any number of Trust Metrics (TM) can be taken for the computation of trust in the sensor nodes. In this model, three important TMs, namely Packet Transmission Ratio (α_T), Packet Latency Ratio (τ_d) and Remaining Energy Ratio (β_E) have been taken as the crisp input variable at Fuzzifier 1. The Fuzzified trust value of the sensor nodes are obtained at the output using Rule Base 1 (consisting of 64 rules).

The **Packet Transmission Ratio** is defined as the ratio of the number of acknowledgements received (α_r) from the suspected node to the total number of packets (α_{tx}) sent to the suspected node. Mathematically,

$$\alpha_T = \alpha_r / \alpha_{tx} \quad \ldots . (5.1)$$

where, $\alpha_r \leq \alpha_{tx}$ and $0<\alpha_T<1$. The case $\alpha_r > \alpha_{tx}$ implies the misbehavior of the node in which the suspected node transmits useless packets or retransmits the same packets several times.

The **Packet Latency Ratio** is the latency of the suspected node (τ_s) to the mean latency of the nodes other than the suspected node (τ_{av}). Mathematically,

$$\tau_d = \tau_s / \tau_{av}. \quad \ldots . (5.2)$$

The case, $\tau_s > \tau_{av}$ implies that the latency of the suspected node is more that the average value and it is considered as the misbehavior of the node. Again, if the input packets are sent very fast without data aggregation, the latency in the suspected node would be very low, which implies $\tau_s << \tau_{av}$ and it is also considered as the misbehavior of the node.

The **Remaining Energy Ratio** (β_E) is defined as the ratio of the present battery voltage (β_p) to the full battery voltage (β_F) of the sensor node. So,

$$\beta_E = \beta_P / \beta_F \quad \ldots . (5.3)$$

where $0< \beta_E<1$.

The uzzy variable ranges of the parameter Packet Transmission Ratio is divided into four classes, given as VL (Very Low), L (Low), M (Medium) and H (High). The parameter Packet Latency Ratio **i**s also divided into four classed termed as VLD (Very Low Delay), LD (Low Delay), AD (Average Delay) and HD (High Delay).

Similarly, the four classes of the parameters Remaining Energy are termed as VLE (Very Low Energy), LE (Low Energy), ME (Medium Energy and HE (High Energy) respectively. The crisp input ranges of the above mentioned Fuzzy variables at the Fuzzifier 1 are shown in Table 5.1. The Rule Base 1 (comprising of 64 rules), which is implemented in Fuzzifier 1, is presented in Table 5.2.

Table 5.1: Crisp Input Range and Fuzzy Variable at Fuzzifier 1

Trust metric	Crisp input range	Fuzzy variable range
Packet Transmission Ratio, α_T	0 - 0.45	VL (Very Low)
	0.4 – 0.6	L (Low)
	0.55 – 0.75	M (Medium)
	0.7 - 1	H (High)
Packet Latency Ratio, τ_d	0 – 0.45	VLD (Very Low Delay)
	0.4 – 0.6	LD (Low Delay)
	0.55 – 1	AD (Average Delay)
	Greater than 1	HD (High Delay)
Remaining Energy Ratio, β_E	0 – 0.45	VLE (Very Low Energy)
	0.4 – 0.6	LE (Low Energy)
	0.55 – 0.75	ME(Medium Energy)
	0.7 – 1	HE (High Energy)

Table 5.2: Rule Base 1 for TFCC algorithm

α_T	τ_d	β_E	Fuzzy Trust Value
VL/L/M/H	VLD/LD/AD/HD	VLE/LE	VLT
VL/L	VLD/HD	ME	VLT
VL/L	VLD/HD	HE	VLT
VL/L	AD/LD	ME	MT
VL/L	AD/LD	HE	MT
M/H	AD/LD	ME	HT
M/H	AD/LD	HE	HT
M/H	VLD/HD	ME	LT
M/H	VLD/HD	HE	LT

Table 5.3: Rule Base 2 for TFCC algorithm

Fuzzy variable range for Trust	Crisp Trust Value, T_{ij}
VLT	0 – 0.45
LT	0.4 – 0.6
MT	0.55 – 0.75
HT	0.7 – 1

The Fuzzy trust value of the nodes are categorically divided into four classes such as Very Low Trust (VLT), Low Trust (LT), Medium Trust (MT) and High Trust (HT) respectively. The crisp trust value of each node is obtained from Defuzzifier 1, which is controlled by Rule Base 2 as shown in Table 5.3.

5.3.1.2 Segregation of Malicious Nodes

Let us consider T_{ij}, T_{ik}, and T_{ip} be the trust value of node with respect to node j, node k and node p respectively. All the nodes denoted by j, k and p be the one hop neighbors of i^{th} node. If $T_{ij} \geq T_{TH}$, the i^{th} node is called the benevolent or trusted node with respect to node j and the link between node i and j is known as the trusted link. Again, if $T_{ik} < T_{TH}$, the node i would not be considered as the benevolent node with respect to node k, although it behaves as benevolent with respect to node j. The link between node i and k is considered as the untrusted link. Similarly, if $T_{ip} \geq T_{TH}$, the link between node i and node p is trusted link. For $T_{ip} < T_{TH}$, it is untrusted link. Data packet routing is possible only through the trusted link. If a node does not have any trusted link with its one hop neighbor, it is called the malicious node. Hence, the malicious nodes are identified from the misbehavior of the nodes and the value of $T_{TH.}$ In the present model, $T_{TH} = 0.5$ (this value is application specific [SSB11] and may vary as per the requirement). The malicious nodes of the sensor network are blocked and would not be able to take part in data packet transmission and data routing.

5.3.1.3 Link State Routing Protocol (LSRP)

In the proposed model, LSRP is applied on the trusted links to get the available routes in the upstream direction from source to sink. The route search algorithm deployed in the proposed model is described in Chapter 3 (Section 3.4) [ACA12].

5.3.2 TFCC: Step Two

Here, the congestion of each trusted node is estimated by a Fuzzy algorithm considering the current buffer queue size of the corresponding sensor node. The parameter Complementary Congestion Index (CCI) is calculated to represent the congestion status of the nodes. Two fixed threshold C_{TH}(Min) and C_{TH}(Max) (values defined in the range of the queue length). If the queue length of a node is less than C_{TH}(Min), congestion is low; when it is in-between C_{TH}(Min) and C_{TH}(Max), the congestion is medium and related linearly to the queue length; The congestion is high when queue length is more than C_{TH} (Max) [MHY08]. Next, a congestion-trust metric is generated and finally, a Fuzzy trust-based congestion control scheme is proposed in which the rate of incoming traffic flow of the sensor node is adjusted dynamically on the basis of the priority of the traffic. The flow diagram of step two of the proposed model is given in Fig. 5.2.

5.3.2.1 Computation of Complementary Congestion Index

The congestion status of a trusted node is obtained from Complementary Congestion Index (CCI) which is a function of buffer queue length. It is assumed that each sensor node has more than one buffer and separate queue to store input packets from each child nodes and from its own local source. So, if i^{th} node has N_i child nodes, then the total number of queues in i^{th} node is given by N_i +1 [2]. For k^{th} queue of i^{th} node, the queue size is denoted by $Q_s(k)$ and we define a parameter $I_K'(i) = f(Q_s(k))$ where $I_K'(i)$ is the CCI of i^{th} node for k^{th} queue. The total CCI for i^{th} node is given by:

$$I'(i) = \left(\prod_{k=1}^{N_i+1} I_k'(i) \right)^{\frac{1}{(N_i+1)}} \quad \ldots . (5.4)$$

Mathematically,

$$I_k'(i) = 1 - I_k(i) \quad \ldots . (5.5)$$

Where, $I_k(i) = \in$ for $Q_s(k) \leq C_{TH}(\text{Min})$,

$I_k(i) = 1$ for $Q_s(k) > C_{TH}(\text{Max})$, and

$$I_k(i) = (1 - \in)\left(\frac{Q_s(k) - C_{TH}(\min)}{C_{TH}(\max) - C_{TH}(\min)} \right) + \in \quad \ldots . (5.6)$$

for $C_{TH}(Min) \leq Q_s(k) < C_{TH}(M_{AX})$ & $\in$ is a small quantity less than one. I′(i) for i^{th} node is taken as the crisp input variable of the Fuzzifier 2. The Fuzzified CCI value of the trusted sensor nodes are obtained at the output of Fuzzifier 2 using Rule Base 3 as shown in Table 5.4.

Table 5.4: Rule Base 3 for TFCC algorithm

Crisp Input range of I′(i)	Fuzzy variable range of I′(i) at the output of Fuzzifier 1
0 – 0.3	VLC (Very Low Congestion)
0.25 – 0.55	LC (Low Congestion)
0.5 – 0.75	MC (Medium Congestion)
0.7 – 1	HC (High Congestion)

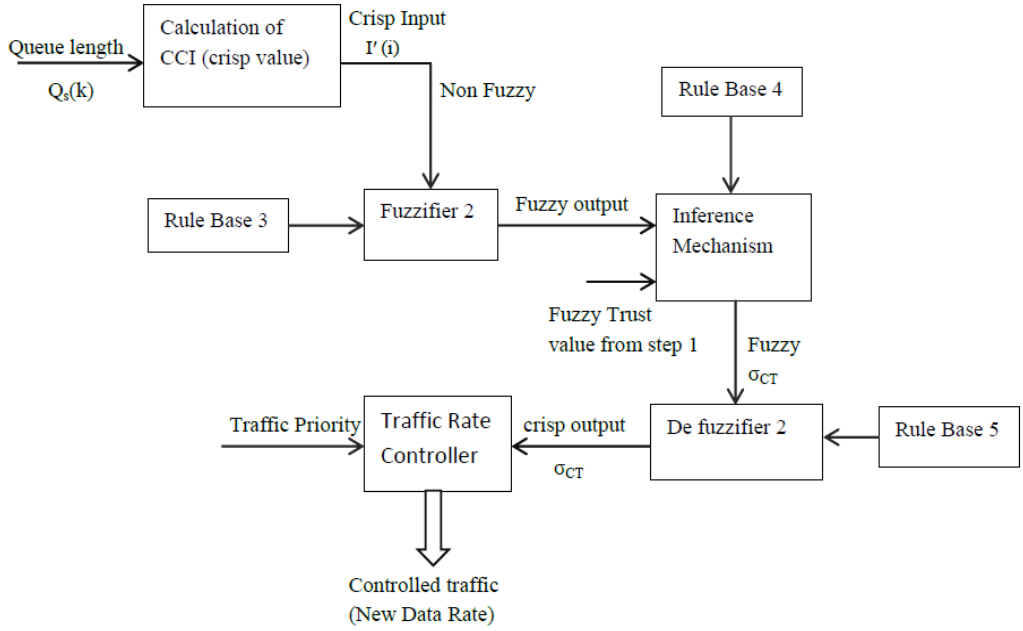


Figure 5.2: Flow diagram of TFCC (step two).

5.3.2.2 Computation of CCI Trust Metric, σ_{CT}

The Fuzzy trust value and Fuzzy I′(i) of the sensor nodes are mapped to CCI-Trust metric, Fuzzy σ_{CT} at the inference mechanism using Rule Base 4, as shown in Table 5.5. The Fuzzy trust value of VLT and LT implies the misbehavior of the malicious nodes and hence they are not considered in Rule Base 4. The crisp value of σ_{CT} is obtained at the output of Defuzzifier 2 which is controlled by Rule Base 5, as depicted in Table 5.6.

Table 5.5: Rule Base 4 for TFCC algorithm

Fuzzy I′(i)	Fuzzy Trust	Fuzzy σ_{CT}
VLC	MT/HT	VL (Very Low)
LC	MT /HT	L (Low)
MC	MT	M (Medium)
HC	MT	H (High)
MC	HT	M (Medium)
HC	HT	H (High)

Table 5.6: Rule Base 5 for TFCC algorithm

Fuzzy variable σ_{CT}	Crisp output range of σ_{CT}
VL	0-0.2
L	0.15 – 0.5
M	0.45 – 0.8
H	0.75 – 1.0

The parameter σ_{CT} of each benevolent node, thus obtained from Defuzzifier 2, reflects the condition of the node in terms of its trust value and congestion status. The high value of σ_{CT} implies highly congested node.

5.3.2.3. Congestion Control and Rate Adjustment

The congestion control refers to the mechanism and techniques for controlling congestion in order to keep the offered load below the capacity of the network. The existing trust based routing protocols mostly select nodes with high trust value. Hence, in most of the cases, trusted nodes are highly congested. In our protocol, the congestion of the trusted nodes are controlled at the Traffic Rate Controller (TRC) which adjust the rate of the traffic flow on the basis of σ_{CT} value and the priority of the corresponding traffic. The local traffic source transmit packet with highest priority. The priority of the other traffic from child nodes is assigned as per the trust value of the node with respect to its parent node. For example, let us consider k1, k2 and k3 are the child nodes of the parent i^{th} node as shown in Fig. 5.3. All the child nodes would like to transmit data packets towards its parent node. Let the trust value of k1, k2 and k3 nodes with respect to

i^{th} node be T_{k1i}, T_{k2i} and T_{k3i} respectively. In TFCC, the local traffic generated from i^{th} node is transmitted with highest priority. The priority of the remaining traffic from k1, k2 and k3 nodes are assigned on the basis of the trust value of the nodes. Hence, if $T_{k1i} > T_{k2i} > T_{k3i}$, then, the priority of the traffic generated from K_1 node is higher than that from K_2 node. Similarly, the priority of the traffic from K_2 node is higher than that from K_3 node and so on.

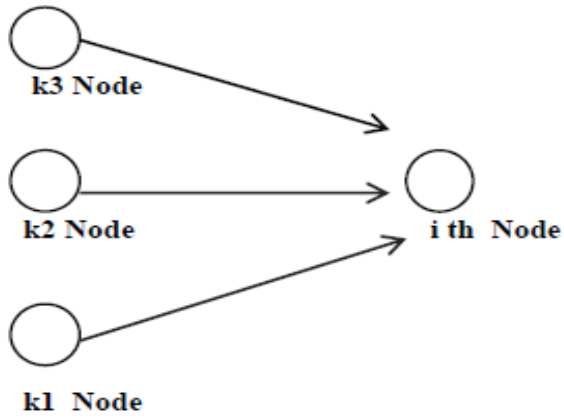


Figure 5.3: Data packet transmission from child to parent node.

The rate of data packet transmission is adjusted at TRC on the basis of the traffic priority and σ_{CT} as shown in Fig. 5.3. The high priority traffic is transmitted at higher rate than the low priority traffic until the condition $C_{TH}(Min) \leq Q_s(k) < C_{TH}(M_{AX})$ is satisfied at the node. When $Q_s(k) > C_{TH}(M_{AX})$, the rate of the data packet transmission is reduced accordingly to maintain the optimum condition. Thus controlled traffic flow is obtained at the output of TRC. So, in TFCC model the congestion conditions of the trusted nodes are adjusted which in turn improves the network congestion as well as the throughput of the network.

5.3.3 Simulation Results of TFCC Algorithm

In this section, the merits of the present protocol have been investigated through some simulation experiments. Here, an arbitrary single tier heterogeneous Wireless Multimedia Sensor Network comprising of 100 nodes (video, audio and scalar sensor nodes) has been considered that are deployed randomly into a field of dimension 50 m X 50 m. The set up under consideration has approximately 50% benevolent nodes while the rest are deemed malicious. It is considered that the nodes deployed in the field are connected with each other by multi hop wireless radio links. The malicious nodes are

identified and blocked against data packet routing by using the concept of trust. All the links considered are duplex in nature. The different paths connecting each individual benevolent node deployed into the field and the source to the sink are obtained via implementation of the Link State Routing Protocol (LSRP).

The proposed algorithm is compared with the existing protocols with colour code as QCCP-PS [MHY08] plotted in blue, and FCCTF [MZA10] plotted in black, while the proposed one is plotted in red, as shown in Fig. 5.4. The normalized throughput is taken as a metric for comparing the protocols. Time is plotted along x-axis while normalized throughput is plotted along the y-axis. The throughput of the sensor network is defined as the number of packets passing through the network in unit time. From the simulation experiment it is observed that the proposed TFCC algorithm gives better result compared to the other two similar existing protocols.

In QCCP-PS algorithm [MHY08], congestion control is proposed by using Rate Adjustment Unit (RAU), but it does not consider the practical conditions pertaining to a sensor network which deteriorate with time, such as node energy, insurgence of malicious packets, nodes turning malicious in nature and trustworthiness of nodes. As TFCC takes all these into consideration, its performance is far better than QCCP-PS [MHY08] in a holistic sense. The other algorithm, such as FCCTF [MZA10] considers both trust and congestion in data packet routing, but trust computation in FCCTF [MZA10] is less scientific, as it does not consider the node energy. The performance of the sensor node highly depends on the remaining battery power and hence it appears as very important trust metric, which is not taken care in FCCTF algorithm [MZA10]. In the proposed TFCC algorithm, remaining energy of the node is taken as one of the trust metric and the better result of the proposed TFCC algorithm compared to FCCTF [MZA10] is justified. Furthermore, in the proposed TFCC algorithm, the most trusted and less congested route is selected dynamically, instead of the shortest route. Thus, it is argued that the proposed trust based congestion control in TFCC protocol performs superior than the conventional congestion control in Wireless Sensor Networks.

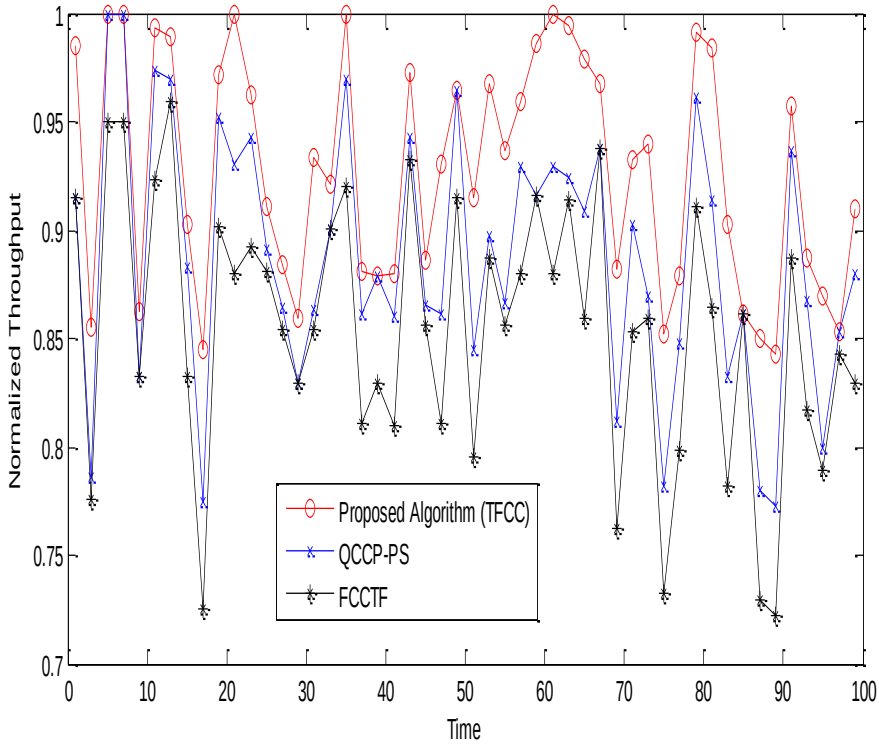


Figure 5.4: Time versus normalized throughput graph.

5.4 Proposed Algorithm 2: TCEER

The proposed TCEER algorithm presents trust integrated congestion aware energy efficient data routing scheme for Wireless Multimedia Sensor Networks, in which the trusted nodes forward data packets to its one hop neighboring node in radio communication range, on the basis of the parameter called Node Potential (NP), which is a function of trust value, congestion status, distance of the node from the BS and the remaining energy of the node. The trust based congestion control in WMSNs and in other application specific large WSNs generate new research area that shows improvement compared to the conventional congestion control mechanisms, in terms of network throughput and QoS. In this section, a new trust based congestion aware data routing algorithm TCEER for WMSNs have been proposed, which is also applicable for WSNs. In this work, it is considered that the WMSNs consisting of N number of multimedia sensor nodes randomly deployed over the sensor field under the condition of free space propagation. Here, it is assumed that all sensor nodes have equal initial energy and trust values. They are able to communicate with each other in their one hop radio range. Each node maintains a database having the above information of its one hop neighboring

nodes, which is updated dynamically on regular intervals. The proposed algorithm consists of two phases. Phase I is the initialization phase whereas Phase II is the routing phase. The details of each phase are described below.

5.4.1 Phase I: Initialization Phase

The phase I computes four parameters, namely trust, complementary congestion index (CCI), distance metric and energy metric. The malicious nodes are also segregated in this phase on the basis of their trust values. In the proposed algorithm, both DT and IT are considered for calculation of overall trust of the nodes. A predefined Trust Threshold (T_{TH}) value is set, depending upon the application of the sensor network. A high value of T_{TH} corresponds to a high level of security of the network. The nodes having trust value greater than T_{TH} are called trusted nodes otherwise they are termed as malicious nodes.

5.4.1.1 Trust Calculation and Segregation of Malicious Nodes

Different methods are available in literature for computing trust value of the sensor nodes. Some of them are described in [SSB11], [ARM11] and [MOM08]. In the proposed TCEER scheme, trust value of a node on its neighboring nodes within the radio range is calculated by GMTMS algorithm [SSB11]. This has certain advantages over the other models. In Momani's model [MOM08], if one of the TM values for data packet transmission is zero and the rest of the TMs have high values, the overall trust value of the node may be above the trust threshold. In this case the node appears to be trustworthy which is not correct. The above difficulties can be avoided with geometric mean based calculations. So, in the proposed model, it is preferred to use GMTMS algorithm [SSB11]. Direct Trust of node N_1 on node N_2 (DT_{N_1,N_2}) is calculated from the geometric mean of the various Trust Metrics for different events occurring between N_1 and N_2. Any number of TM can be considered for trust computation.

For *k* TMs it is represented by the formula given as,

$$DT_{N_1,N_2} = \left(\prod_{i=1 \text{ to } k} (TM_i) \right)^{\frac{1}{k}} \quad \ldots . (5.7)$$

Indirect Trust of node N1 on node N2 (IT_{N_1,N_2}) is computed by the geometric mean of various Direct Trusts (DTs), obtained from different neighboring nodes of N_1. This is represented as,

$$IT_{N_1,N_2} = \left(\prod_{j=1 \text{ to } l} (DT_j) \right)^{\frac{1}{l}} \quad \ldots . (5.8)$$

where, $DT_1, DT_2, DT_3, \ldots\ldots, DT_l$ are the DTs from l number of neighboring nodes of N_1.

The overall trust of node N1 on node N2 ($T_{N1, N2}$) is the weighted sum of DT and IT which is represented by the Eq. (5.9) as shown below.

$$T_{N1, N2} = W_D * DT_{N1, N2} + W_I * IT_{N1, N2} \quad \ldots . (5.9)$$

where, W_D and W_I are the weights to DT and IT respectively and $W_D + W_I = 1$. In some applications, DT has been given more importance than IT and accordingly the value of W_D is chosen higher than that of W_I. In the proposed scheme, equal values of W_D and W_I are considered, which implies equal importance towards DT and IT respectively. For trusted node $T_{N1, N2} > T_{TH}$, whereas, for malicious nodes $T_{N1, N2} < T_{TH}$. Thus, in TCEER scheme, all the malicious nodes are identified by their trust values and then blocked, so that they cannot take part in the data packet routing algorithm.

5.4.1.2 Congestion Evaluation

In the proposed work, congestion of the sensor nodes is estimated from the buffer queue size of the corresponding nodes. We have introduced a new congestion metric known as the Complementary Congestion Index (CCI) which measures the congestion status of the nodes. CCI is defined as the function of the buffer queue length and is quantified as described below.

5.4.1.3 Computation of CCI

Computation of CCI for trusted nodes, having trust values greater than T_{TH} level are considered. Since computation of CCI for malicious nodes are excluded, the energy overhead is reduced. Two fixed threshold values, C_{Th}(Min) and C_{Th}(Max) are defined in the range of the buffer queue length. If buffer queue length is less than C_{Th}(Min), congestion is low, if it is in between C_{Th}(Min) and C_{Th}(Max), congestion is medium and

if greater than C_{Th}(Max), congestion is high [MHY08]. It is assumed that every sensor node has only one buffer where it stores all the packets that are obtained from its own local source as well as the packets accepted from its one-hop neighbors. Let the buffer queue length of the k^{th} node be denoted by $Q_s(k)$. The Complementary Congestion Index (CCI) is calculated which is the function of the buffer queue length. Let CCI for the k^{th} node be represented by I_K' and $I_K' = f(Q_s(k))$. Then I_K' can be computed mathematically from **Congestion Index** I_K as given by $I_K' = 1 - I_K$ which is represented in Eq. (5.5). The mathematical formulae for computation of CCI are shown below in Table 5.7.

Table 5.7: Formulae for Computation of CCI

$Q_s(k)$	**I_k**
$Q_s(k) \leq C_{TH}(Min)$	$\in$ (where $\in$ is a small quantity)
$C_{TH}(Min) \leq Q_s(k) \leq C_{TH}(Max)$	$(1-\in)\left(\frac{Q_s(k)-C_{TH}(\min)}{C_{TH}(\max)-C_{TH}(\min)}\right)+\in$
$Q_s(k) > C_{TH}(Max)$	1

5.4.1.4 Evaluation of Residual Energy

The residual energy of the node is one of the most important parameter in hop by hop routing protocol. In the proposed work, let us consider that initial energy of all nodes be same and is denoted by $E_{initial}$. The effective residual energy (E_{er}) of the node is normalized as:

$$E_{er} = \omega * E_{cn} + (1-\omega) * E_{pnn} \quad \ldots . (5.10)$$

Here E_{cn} denotes the energy of the present source node and E_{pnn} represents the energy of the potential next node in one hop neighbor within the radio communication range. The parameter ω is the weighing factor, usually set to a value less than 0.5 so as to give higher priority to the remaining energy of the potential next node. The potential next node is the nearest trusted node from the present source node, within its one hop neighbor radio communication range.

5.4.1.5 Evaluation of Distance Metric

In order to ensure the direction of data transmission from source node towards BS, we have introduced a new parameter known as the Dist_Metric. The meaning of the parameter Dist_Metric is explained with the help of the Fig. 5.5. As shown in Fig. 5.5, node A, B and C represent the present source node, the potential next node and the BS respectively. The node B is lying within one hop neighbor radio communication range of node A as shown by the dotted line. Let d_1 be defined as the ratio of the distance between the present source node and the potential next node to the radio communication range of the sensor node. Similarly, d_2 is the ratio of the distance between the potential next node and BS to the distance between the present source node and BS. Thus, from Fig. 5.7, d_1 = AB/r and d_2 = BC/AC, where radio communication range of the sensor node is specified as 'r' unit. The parameters d_1^C and d_2^C are called the complementary distance from the present source node and the complementary distance from the BS respectively, which are represented by the Eq. 5.11 as,

$$d_1^C = 1 - d_1,\ d_2^C = 1 - d_2 \qquad \ldots . (5.11)$$

It is obvious that lower values of d_1 and d_2 or higher values of d_1^C and d_2^C imply more desirable condition. The Dist_Metric of the potential next node is related to d_1^C and d_2^C which is represented by Eq. 5.12 as,

$$\text{Dist\_Metric} = \frac{k_1 * d_1^C + k_2 * d_2^C}{k_1 + k_2} \qquad \ldots . (5.12)$$

where, k_1 and k_2 are the weights of d_1^C and d_2^C respectively. In the proposed TCEER algorithm, more importance are given to the distance of the potential next node from the BS and hence the value of k_2 is chosen higher, compared to the value of k_1. In order to make the routing distance minimum, the potential next node should always be closer to the BS, compared to the present source node. Hence, in Fig. 5.5, the distance BC is less than the distance AC which implies that the ratio d_2 is always less than one.

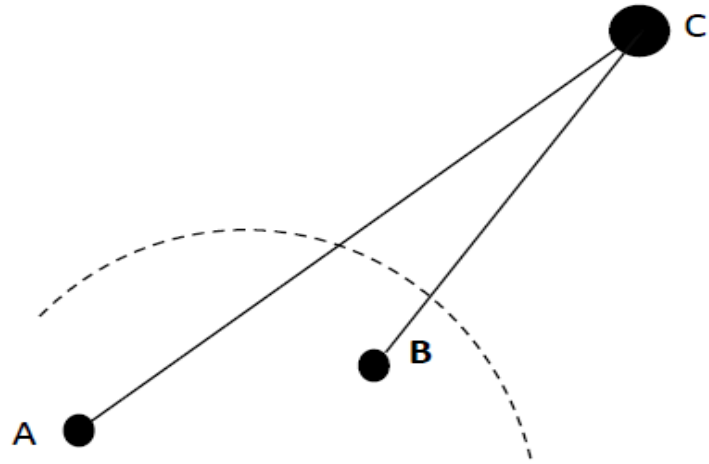


Figure 5.5: Computation of Dist_Metric.

5.4.2 Phase II: Routing Phase

In this phase, a Fuzzy Logic Controller (FLC) is used for the computation of the parameters known as the Trust Congestion Metric (TCM) and the Energy Distance Metric (EDM) of the trusted nodes. Fuzzy Logic Controller (FLC) is considered for the quantitative analysis of the parameters from the qualitative or imprecise information. The characteristics of the sensor nodes are described in terms of trust and congestion, with the help of the parameter called Trust Congestion Metric (TCM). The residual energy and the distance of the node from the BS are quantified by the parameter called Energy Distance Metric (EDM).

5.4.2.1 Computation of TCM and EDM

The configuration of the FLC used in TCEER algorithm is shown in Fig. 5.6. It consists of a Fuzzifier-1/ Defuzzifier-1/ Rule Base-1/ Inference Mechanism-1 and a Fuzzifier-2/ Defuzzifier-2/ Rule Base-2/ Inference Mechanism-2, respectively. The four parameters (Trust, CCI, Effective residual energy E_{er} and Dist_Metric) obtained from phase I, are taken as the inputs to the FLC. The TCM and EDM are the two outputs of the FLC that are inferred through the corresponding rule bases and inference mechanisms. The Fuzzy trust values of the nodes are categorically divided into five classes namely Very Low Trust (VLT), Low Trust (LT), Medium Trust (MT), High Trust (HT) and Very High Trust (VHT). Similarly, Fuzzy CCI values of the nodes are classified as Very Low CCI (VLCC), Low CCI (LCC), Medium CCI (MCC), High CCI (HCC) and Very High CCI

(VHCC). The crisp input range and Fuzzy input variable for trust and CCI are shown in Table 5.8 and 5.9 respectively. Similarly, crisp input ranges and the corresponding Fuzzy input variables for Dist_Metric and the residual energy are shown in Table 5.10 and Table 5.11 respectively. The Dist_Metric is classified into five categories, such as Very Far Distance (VFD), Far Distance (FD), Medium Distance (MD), Close Distance (CD) and Very Close Distance (VCD). Similarly, residual energy is Fuzzified into five classes, like Very Low Energy (VLE), Low Energy (LE), Medium Energy (ME), High Energy (HE) and Very High Energy (VHE).

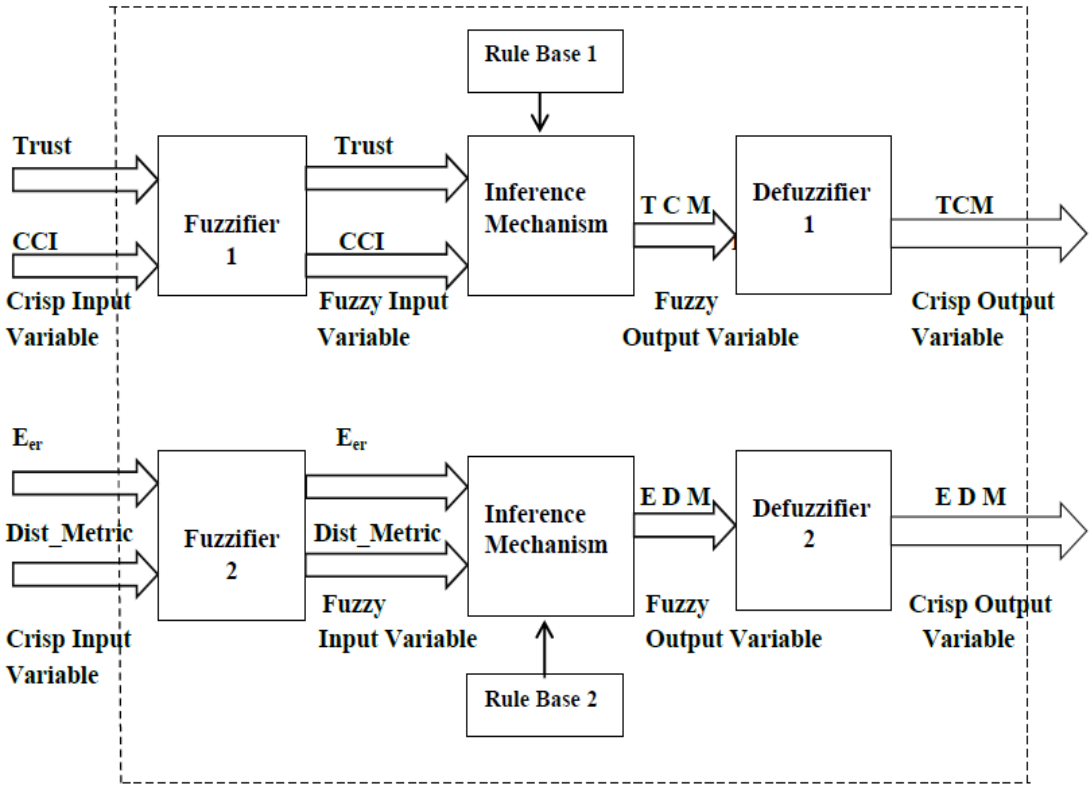


Figure 5.6: Schematic diagram of FLC used in TCEER

Table 5.8: Crisp Input Range and Fuzzy Trust Value

Crisp Input Range	Fuzzy Trust Value
0 – 0.4	VLT
0.2 – 0.6	LT
0.5 – 0.8	MT
0.75 – 0.95	HT
0.85 – 1	VHT

Table 5.9: Crisp Input Range and Fuzzy CCI Value

Crisp Input Range	Fuzzy CCI Value
0 – 0.3	VLCC
0.25 – 0.5	LCC
0.45 – 0.75	MCC
0.7 – 0.9	HCC
0.8 – 1	VHCC

Table 5.10: Crisp Input Range and Fuzzy Dist_Metric

Crisp Input Range	Fuzzy Dist_Metric Value
0 – 0.3	VFD
0.2 – 0.4	FD
0.35 – 0.65	MD
0.6 – 0.85	CD
0.8 – 1.0	VCD

Table 5.11: Crisp Input Range and Fuzzy E_{er}

Crisp Input Range	Fuzzy E_{er} Value
0 – 0.3	VLE
0.2 – 0.5	LE
0.4 – 0.7	ME
0.6 – 0.9	HE
0.8 – 1	VHE

Table 5.12: Fuzzy TCM Value and Crisp Output Range

Fuzzy Output Variable, TCM	Crisp Value of the Output Variable, TCM
VL	0 – 0.2
L	0.1 – 0.4
M	0.3 – 0.7
H	0.65 – 0.95
VH	0.8 – 1.0

Table 5.13: Rule Base 1 for TCEER algorithm

Sl no.	Fuzzy CCI Value	Fuzzy Trust Value	Fuzzy Output Variable TCM
1.	VLCC /LCC/MCC	VLT	VL
2.	HCC/ VHCC	VLT	L
3.	VLCC	LT	VL
4.	LCC/MCC	LT	L
5.	HCC/VHCC	LT	M
6.	VLCC/LCC	MT	L
7.	MCC/HCC	MT	M
8.	VHCC	MT	H
9.	VLCC	HT	L
10.	LCC	HT	M
11.	MCC	HT	H
12.	HCC/VHCC	HT	VH
13.	VLCC	VHT	L
14.	LCC	VHT	M
15.	MCC	VHT	H
16.	HCC/VHCC	VHT	VH

The TCM Fuzzy values are classified as Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH) respectively and the crisp value of TCM with respect to the Fuzzy value is depicted in Table 5.12. The inference mechanism and Rule Base 1 for generation of Fuzzy output variable TCM are shown in Table 5.13. The Residual Energy

E_{er} obtained from Eq. (5.10) and the Dist_Metric obtained from Eq. (5.12), are taken as the two input variables in Fuzzifier 2. The Dist_Metric is fuzzified into five classes known as Very Far Distance (VFD), Far Distance (FD), Medium Distance (MD), Close Distance (CD) and Very Close Distance (VCD). Similarly, the residual energy is also classified as Very Low Energy (VLE), Low Energy (LE), Medium Energy (ME), High Energy (HE) and Very High Energy (VHE).

The Fuzzy value of EDM is divided into five classes namely Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH). The crisp value of EDM with respect to the corresponding Fuzzy value is given in Table 5.14 and the Rule Base 2 is represented in Table 5.15 respectively. The Fuzzy EDM is obtained at the output, as per the inference mechanism which is controlled by the Rule Base 2, having 25 rules shown in Table 5.15. The Fuzzy output EDM is classified into four types, given as Very Low (VL), Low (L), Medium (M) and High (H) respectively.

Table 5.14: Fuzzy EDM and Crisp Output Range

Fuzzy Output EDM	Crisp Value of EDM
VL	0 – 0.25
L	0.15 – 0.35
M	0.3 – 0.8
H	0.75 – 0.95
VH	0.85 – 1.0

5.4.2.2 Computation of Node Potential and Data Packet Routing

Data packet routing in TCEER algorithm is done on the basis of the parameter known as the Node Potential (NP) which is a function of trust value, congestion status, distance of the node from the BS and the remaining energy of the node. NP of the trusted node is calculated by the relation given by,

$$\text{Node Potential} = \frac{\alpha * EDM + \beta * TCM}{\alpha + \beta} \quad \ldots . (5.13)$$

Here α and β are the weightage of EDM and TCM respectively that are assigned on the basis of the application of the sensor network. For example, in some security related

applications, more importance is given to TCM compared to EDM and hence the value of β is kept higher than α. It is to be noted that the summation of α and β is always unity.

Table 5.15: Rule Base 2 for TCEER algorithm

Sl no.	Fuzzy E_{er}	Fuzzy Dist_Metric	Fuzzy Output EDM
1.	VLE/LE	VFD	VL
2.	ME/HE	VFD	L
3.	VHE	VFD	M
4.	VLE	FD	VL
5.	LE/ME	FD	L
6.	HE/VHE	FD	M
7.	VLE/LE	MD	L
8.	ME	MD	M
9.	HE/VHE	MD	H
10.	VLE/LE	CD	L
11.	ME	CD	M
12.	HE	CD	H
13.	VHE	CD	VH
14.	VLE	VCD	L
15.	LE	VCD	M
16.	ME	VCD	H
17.	HE/VHE	VCD	VH

Nomenclatures of the various parameters used in the proposed TCEER algorithm are listed in Table 5.16; the hop by hop data packet routing mechanism and the route formation with TCEER protocol is explained in Fig. 5.7; the dotted circles represent the radio communication range of the nodes. The source node denoted by S selects destination node with highest NP from its one hop neighboring nodes in the radio communication range. The node with NP value less than some threshold value (NP_{TH}) is not considered for data packet routing. In the next hop, the above mentioned destination node acts as the present source node and selects the intermediate destination node with highest NP from its one hop neighboring nodes in the radio communication range.

Similarly, in the next hop, the previously mentioned intermediate destination node acts as present source node and selects another intermediate destination node with highest NP from its one hop neighboring nodes. In this way data packets are forwarded hop by hop from the source node S until the destination node is the BS.

Table 5.16: Nomenclature of the Parameters

Parameter	Description
DT	Direct Trust
IT	Indirect Trust
TM	Trust Metric
T_{TH}	Trust Threshold
CCI	Complementary Congestion Index
$E_{initial}$	Initial Energy of the Nodes
E_{er}	Effective Residual Energy of the Nodes
E_{cn}	Energy of the present Source Node
E_{pnn}	Energy of the potential Next Node
d_1^C	Complementary Distance from the present Source Node
d_2^C	Complementary Distance from the Base Station
Dist_ Metric	Distance Metric
TCM	Trust Congestion Metric
EDM	Energy Distance Metric
NP	Node Potential
PRR	Packet Reception Ratio
MRA	Maximum Retransmission Attempts

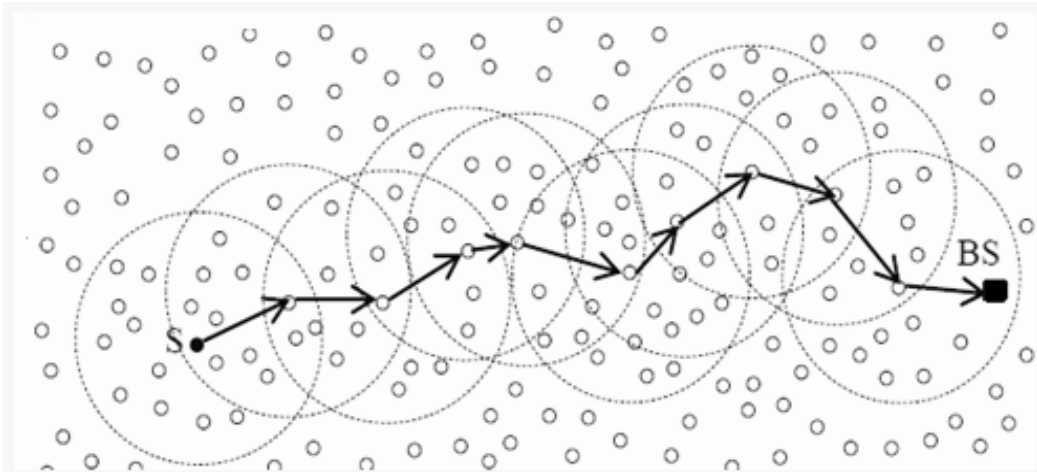


Figure 5.7: Route formation with TCEER algorithm.

5.4.3 Simulation Results of TCEER Algorithm

In this section, the merits of the proposed TCEER scheme have been investigated through extensive MATLAB simulations. An arbitrary network is considered, comprising of 50 multimedia sensor nodes deployed randomly into a field of dimensions 100 m X 100 m and 200 m X 200 m respectively. The distances of the nodes from the base station is taken stationary throughout the experiment. In the simulation experiments, any number of TMs can be considered. However, for the sake of simplicity, we have taken only three TMs, namely data packet forwarded, packet address modified and remaining energy of the nodes. Initially, it has been considered that the sensor nodes are all trusted nodes. Trust value of a node is updated periodically after time Δt equal to 5 seconds. A Trust Threshold (T_{Th}) value is taken as 0.5 whereas minimum and maximum trust values are 0 and 1 respectively. The values of the constant parameters that have been considered in the calculations of the proposed work are listed in Table 5.17; we have considered, W_D equal to W_I, which implies that DT and IT are given equal importance in the computation of the overall trust of the node, as represented in Eq. (5.9). Again, ω is kept less than 0.5, in order to put more importance on the remaining energy of the potential next node compared to that of the current source node, as given in Eq. (5.10); in the simulation experiment, ω has been arbitrarily set to 0.2. Similarly, k_2 is chosen higher than k_1 so that, the distance of the potential next node from the BS is less than the distance of the present source node to the BS, as given in Eq. (5.12).

Table 5.17: Constant parameters used in TCEER

Parameter	Weightage	Value
W_D	Direct Trust	0.5
W_I	Indirect Trust	0.5
Ω	for calculation of Energy Metric	0.2
k_1	d_1^C for calculation of Distance Metric	2
k_2	d_2^C for calculation of Distance Metric	3
A	EDM for calculation of Node Potential	0.3
B	TCM for calculation of Node Potential	0.7

Again, as shown in Table 5.17, $\beta > \alpha$ implies that the trust and congestion of the node are given more importance than the remaining energy and the distance of the node from the BS, during computation of NP of the corresponding node. Thus, the values of the constant parameters used in the proposed scheme are justified. The graphical view of the parameters, TCM and EDM of the sensor nodes, as obtained from the simulations of TCEER algorithm, are shown in Fig. 5.8 and 5.9 respectively.

The data packet routing in TCEER protocol for different numbers of packets are simulated in MATLAB. The route formation in TCEER with a single packet, 5 packets and 20 packets are shown in Figs. 5.10, 5.11 and 5.12 respectively. Since the nodes are deployed randomly in the sensor fields, the node position changes in each simulation experiment. It is found that the packets have taken different routes to reach the BS, at different time, depending on the values of the NP of the intermediate nodes, which is modified dynamically at regular time interval.

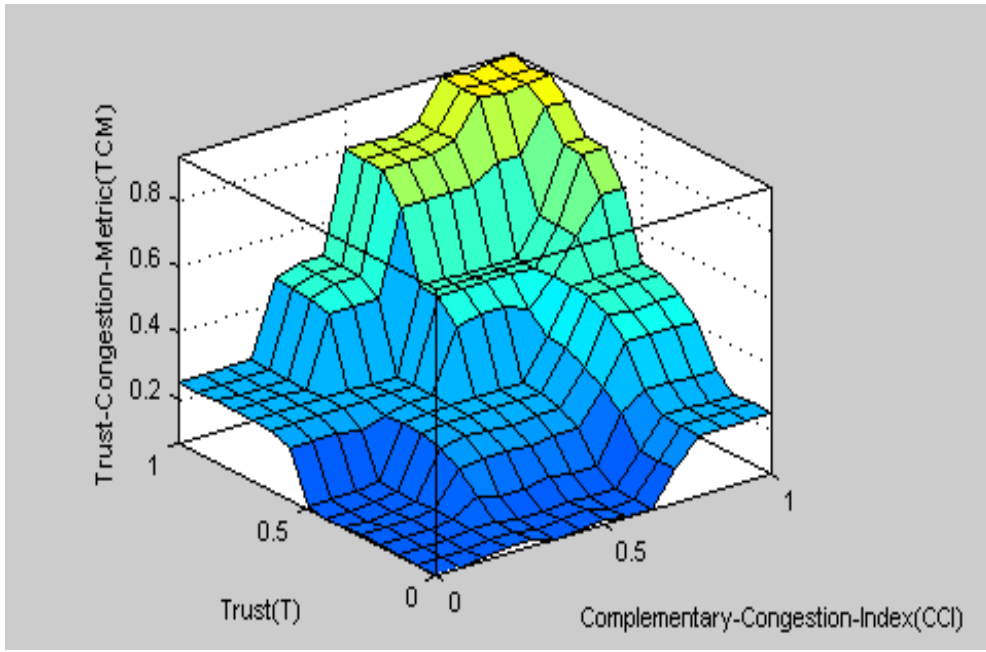


Figure 5.8: TCM for different values of Trust and CCI

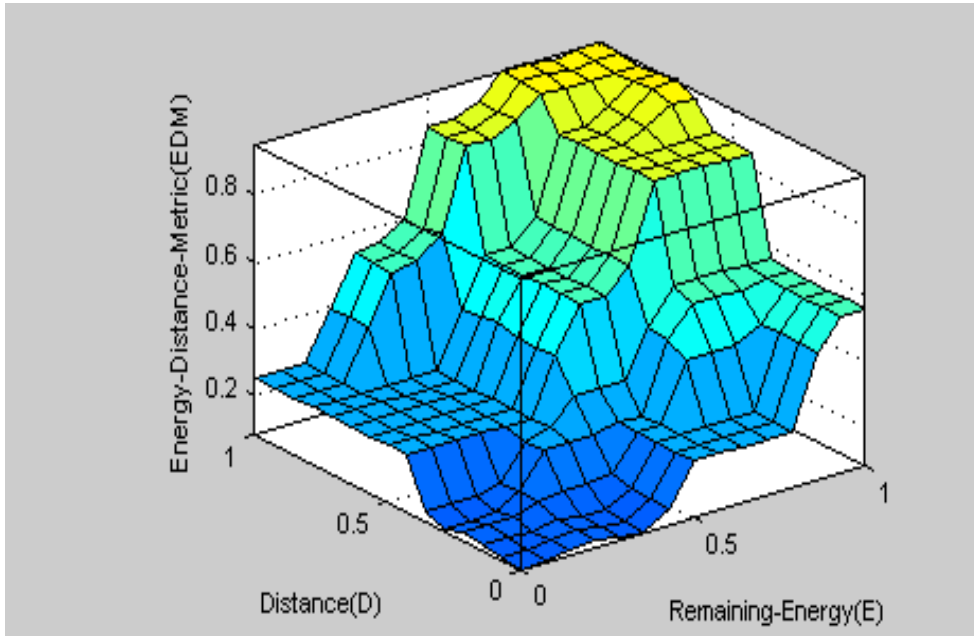


Figure 5.9: EDM for different values of Distance Metric and Residual Energy

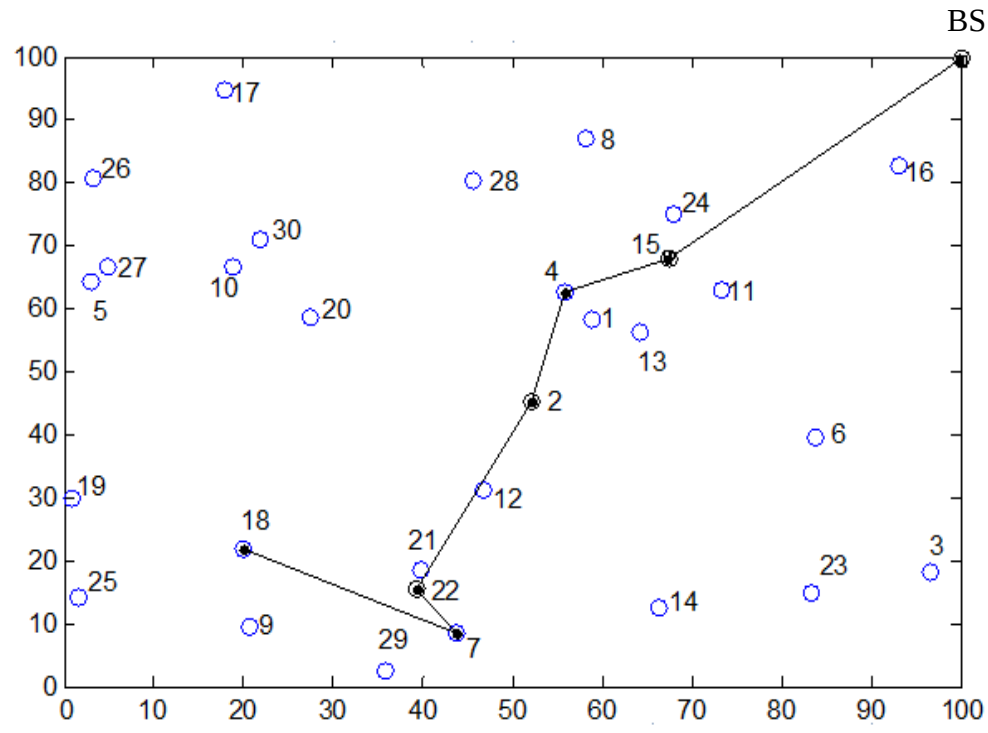


Figure 5.10: Routing of single packet from node 18 to the Base Station.

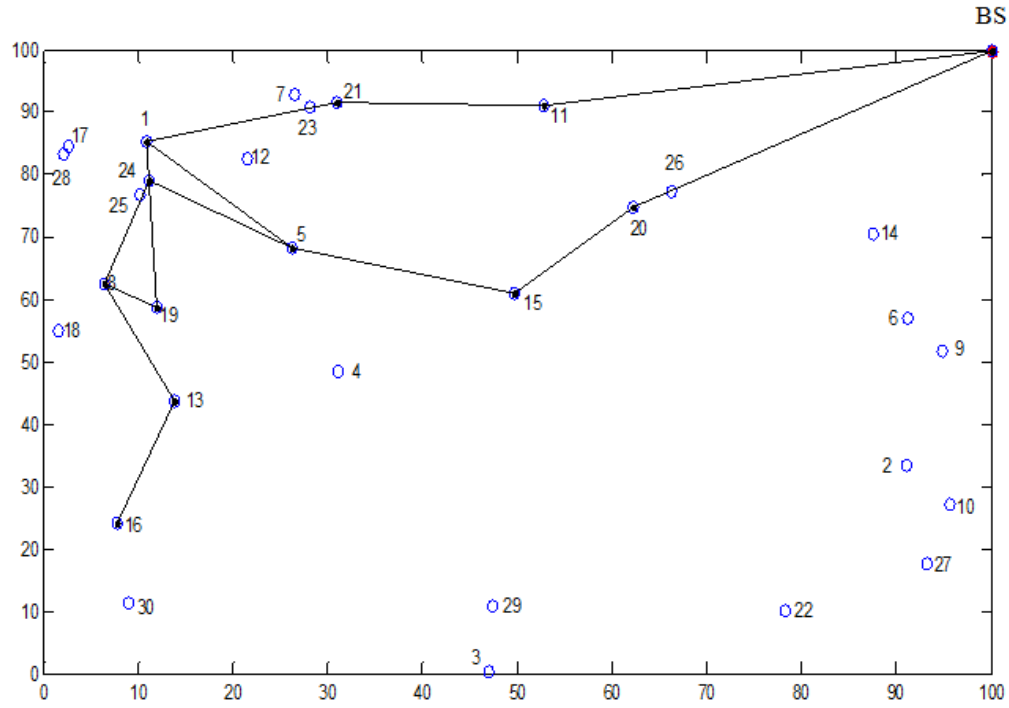


Figure 5.11: Routing of 5 packets from node 16 to the Base Station

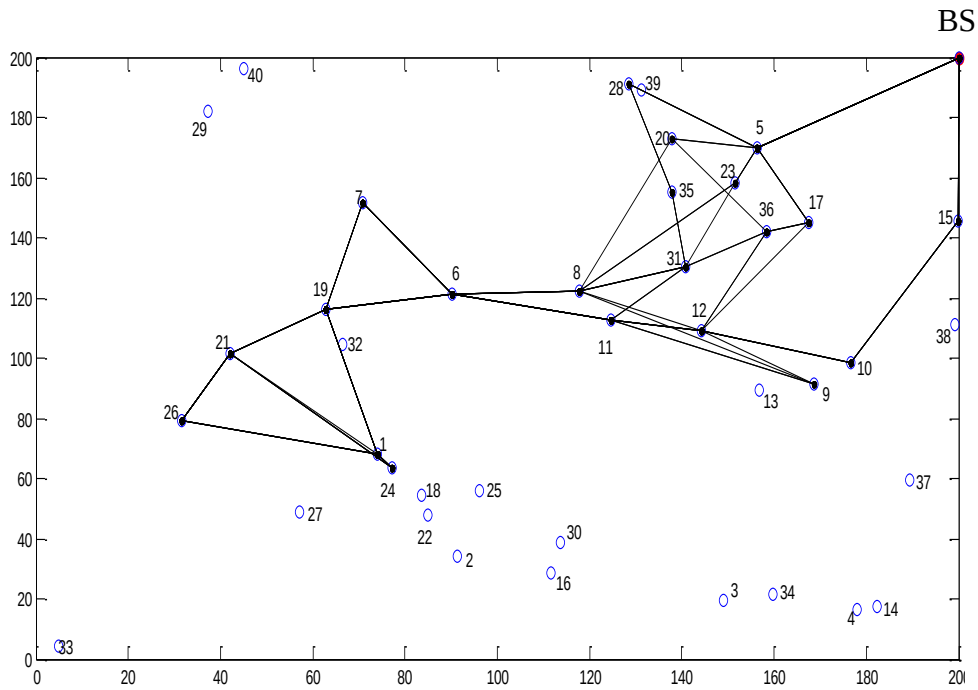


Figure 5.12: Routing of 20 packets from node 24 to the Base Station

The comparison of the proposed TCEER protocol is made with the existing algorithms, viz. T-LEACH [JHJ09], TRANS [STP14], TFCC [ACS13] and TC-ACO [ASA13] for different initial node energies on a 200 m X 200 m WSN field. The number of rounds versus percentage of dead nodes for the above mentioned protocols is given in Table 5.18, for various initial node energies. The simulation results are plotted with percentage of dead nodes as the abscissa and number of rounds as the ordinate, in Figs. 5.13, 5.14 and 5.15 for initial node energy of 0.25 Joules per node, 0.5 Joules per node and 1.0 Joule per node respectively. The graphs and figures indicate that the proposed TCEER protocol provides higher network lifetime for different node energies, compared to other similar protocols and thereby outperform its peers. It has been also observed that the proposed scheme provides better results for the initial node energy of 1.0 Joule per node in comparison to that of 0.25 Joules per node and 0.5 Joules per node respectively.

Table 5.18: Number of Rounds with Percentage of Dead Nodes for Various Algorithms

Initial Energy (J/node)		Protocol	Percentage of dead nodes						
			1%	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*
0.25	Number of Rounds	TRANS	682	792	836	845	912	967	985
		T-LEACH	750	818	864	908	945	983	1028
		TFCC	834	892	918	970	1005	1020	1047
		TC-ACO	855	910	993	1063	1089	1075	1157
		TCEER	891	952	1060	1122	1140	1188	1202
0.5		TRANS	1258	1334	1480	1512	1604	1640	1710
		T-LEACH	1312	1405	1512	1598	1663	1710	1802
		TFCC	1320	1440	1498	1580	1647	1701	1796
		TC-ACO	1352	1495	1523	1627	1689	1723	1821
		TCEER	1386	1599	1634	1688	1745	1788	1873
1		TRANS	1965	2132	2242	2496	2701	2910	3147
		T-LEACH	2087	2221	2378	2601	2895	3020	3304
		TFCC	2223	2365	2455	2673	2812	3108	3276
		TC-ACO	2235	2413	2559	2713	2888	3217	3345
		TCEER	2406	2677	2818	2997	3108	3285	3566

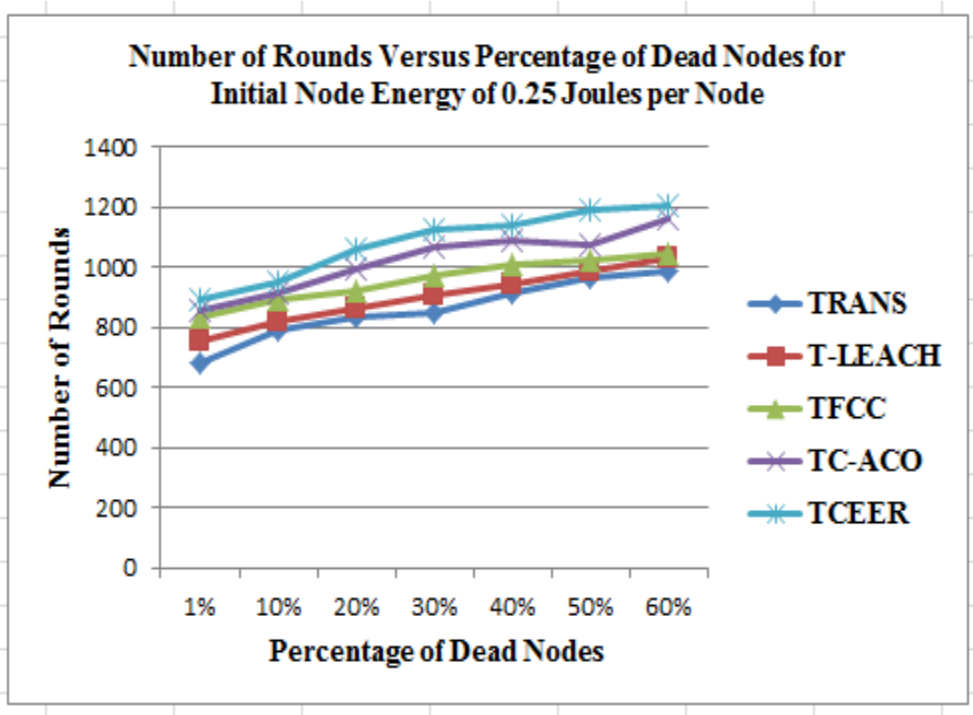


Figure 5.13: Performance analysis with initial energy of 0.25 Joules per node

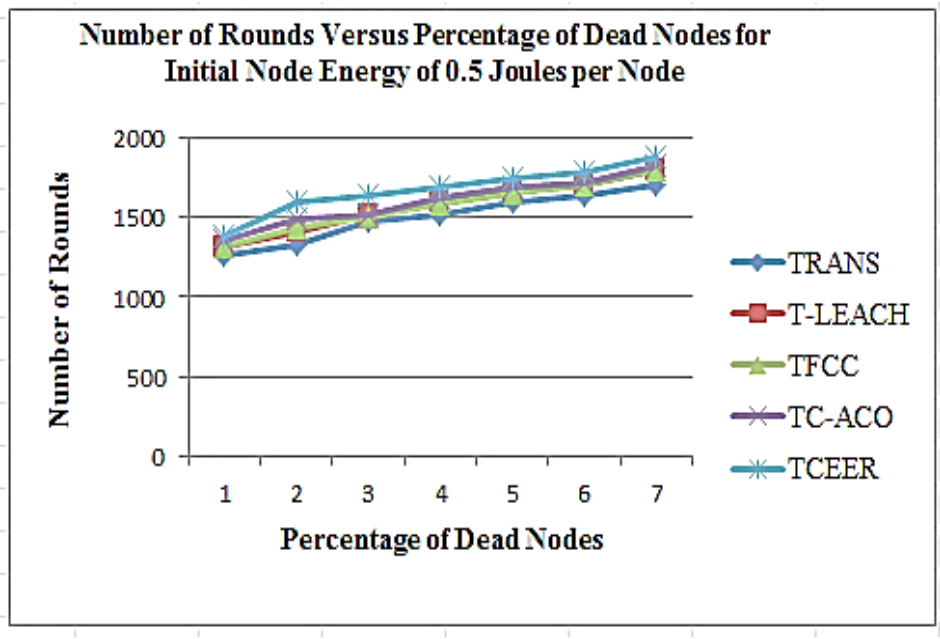


Figure 5.14: Performance Analysis with Initial Energy of 0.5 Joules per Node

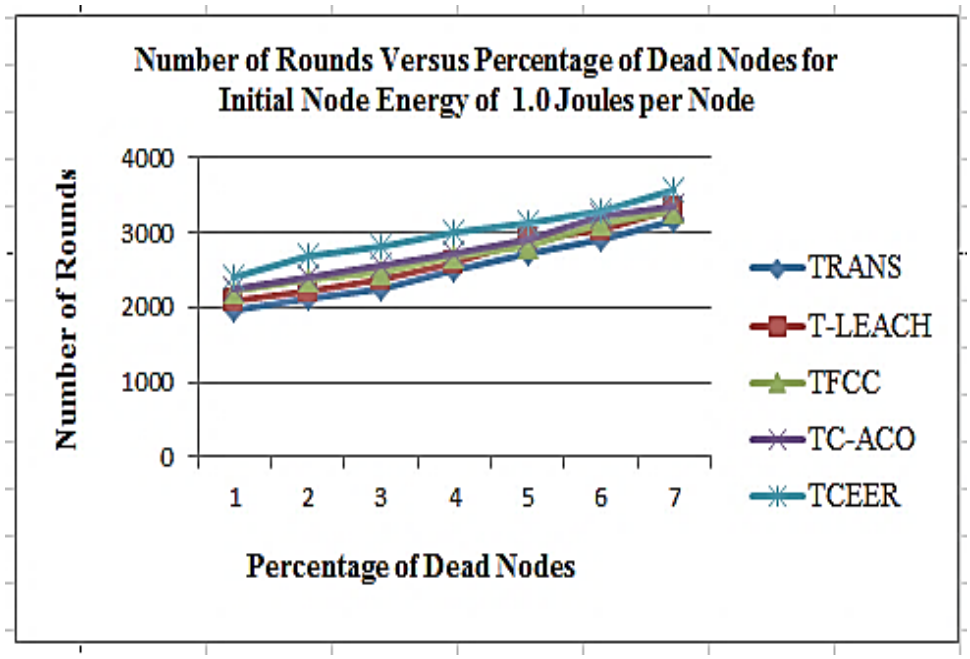


Figure 5.15: Performance Analysis with Initial Energy of 1.0 Joule per Node

Next, the proposed TCEER scheme has been studied to find out the impact on Packet Reception Ratio (PRR) and Maximum Retransmission Attempts (MRA) [ACK09], in comparison with the other existing protocols. The PRR is calculated as the ratio of the number of packets received successfully to the total number of packets transmitted. The packet retransmission is required in case of unsuccessful packet delivery. MRA means

the maximum number of retransmission needed for a particular packet to send it successfully. In our experiment, we calculate the fraction of the packets reaching the BS successfully, by varying the number of retransmission attempts. As the number of retransmission attempts increase, the PRR also increases. In case of successful packet delivery, PRR is equal to one. Table 5.19 represents the maximum and minimum number of packets delivered successfully to the BS for different values of MRA for T-LEACH [JHJ09], TRANS [STP14], TFCC [ACS13], TC-ACO [ASA13] and the proposed TCEER algorithms respectively. MIN and MAX refer to the maximum and minimum values of PRR obtained over 30 runs for transmitting 25 packets. The simulation results shown in Table 5.19 are based on the values of the parameters listed in Table 5.17. It shows that, less number of retransmission is required in TCEER for achieving PRR value equal to one, compared to the other similar algorithms.

Next, the proposed TCEER scheme is verified under different parameter settings, where each parameter is varied, one at a time, keeping the values of other parameters constant. Table 5.20 represents number of rounds recorded, when the values of α and β are varied, keeping all other parameters at the previous values (ω = 0.2, k_1= 2 and k_2= 3). The number of rounds at the critical point of the sensor network is recorded where 50% nodes are dead. The initial node energy is considered as 0.5 Joules/node. It has been observed that maximum number of rounds are obtained for the case when α and β is equal to 0.3 and 0.7 respectively. Since the parameter α and β represent the weightage on EDM and TCM respectively as represented in Eq. (5.13), it would imply from Table 5.20 that trust and congestion have greater impact on network lifetime compared to the remaining energy and the distance of the node from the BS. If β is zero, that is, congestion and trust factor are not considered at all, number of rounds goes to minimum.

Table 5.19: PRR and MRA comparison

Protocol	Packet Reception Ratio (PRR)	Maximum Number of Attempts (MRA)					
		1	2	3	4	5	6
TRANS	MIN	0.53	0.61	0.74	0.81	0.96	1
	MAX	0.64	0.68	0.82	0.97	1	1
T-LEACH	MIN	0.51	0.63	0.7	0.79	0.98	1
	MAX	0.68	0.7	0.85	0.95	1	1
TFCC	MIN	0.55	0.59	0.68	0.79	0.98	1
	MAX	0.71	0.75	0.9	0.98	1	1
TC-ACO	MIN	0.62	0.65	0.72	0.96	1	1
	MAX	0.74	0.84	0.96	0.98	1	1
TCEER	MIN	0.61	0.66	0.81	0.89	1	1
	MAX	0.73	0.85	1	1	1	1

Table 5.20: Number of rounds when α and β is varied.

A	B	Number of rounds
0	1	1577
0.1	0.9	1687
0.2	0.8	1722
0.3	0.7	1748
0.4	0.6	1734
0.5	0.5	1691
0.6	0.4	1675
0.7	0.3	1652
0.8	0.2	1623
0.9	0.1	1594
1	0	1564

Table 5.21: Number of rounds with variation of ω

Existing Protocols	No. of Rounds at 50% Dead Nodes	Variation of ω in Proposed TCEER	No. of Rounds at 50% Dead Nodes in Proposed TCEER
TRANS	1690	Ω	0
		0	1633
		0.1	1712
T-LEACH	1701	0.2	1748
		0.3	1732
		0.4	1721
TFCC	1710	0.5	1698
		0.6	1645
		0.7	1621
TC-ACO	1714	0.8	1497
		0.9	1278
		1	988

The case when ω is set to zero implies that the energy of the current source node is not considered in the calculation of the energy metric, as shown in Eq. (5.10). On the other hand, ω equals to one implies that the contribution of energy of the potential next node is set to zero in the calculation of the energy metric. This is not desirable. In this case, minimum number of rounds are obtained, which is quite expected because energy of the potential next node has higher impact in data packet routing protocol compared to the source node.

The number of rounds obtained in TCEER algorithm for 50% dead nodes are compared with that for the existing protocols. Better results are obtained in the proposed TCEER protocol for the setting $0 < \omega < 0.5$, as shown in Table 5.21.

5.5 Summary

In the proposed TFCC algorithm, a new trust based congestion control scheme for WSNs and WMSNs is presented, in which the rate of traffic flow is dynamically adjusted on the basis of the priority of the traffic. It is very much relevant, as a recent research topic since multimedia applications in WSNs is gaining popularity nowadays. The TFCC protocol considers the impact of the network congestion due to the misbehaviour of the faulty

nodes and thereby minimizes their effect during packet transmission. The protocol resolves congestion of the individual nodes and hence reduces network congestion significantly. The congestion control and quality of service (QoS) are two closely bounded issues and improving one implies improving the other. The simulation and experimental results indicate that TFCC provides higher throughput compared to other similar protocols and thereby outperform its peers. Here, an interval type I Fuzzy set has been adopted for computational purposes. In spite of its several advantages, Fuzzy sets are limited by the fact that the degree of membership of an entity in a particular class also has a measure of uncertainty associated with it. Hence, this leads to a certain degree of uncertainty which is associated with the resulting data packet transmission rate to the various nodes. This can be overcome by means of either type II Fuzzy computation or ordinary deterministic non-Fuzzy computation.

In the TCEER algorithm, the relationship between trust and congestion are discussed and a novel trust based congestion aware routing protocol using Fuzzy Logic Controller for WMSNs is proposed, which is also applicable for large scale WSNs. The proposed scheme protects sensor networks against various security attacks by efficient detection and avoidance of malicious nodes. The optimum route for the data packet transfer is dynamically selected on the basis of the parameter called Node Potential (NP) which is a function of the trust and congestion status of the sensor nodes. The simulation results show that the proposed TCEER algorithm provides a significant improvement of 25% in terms of number of rounds and network lifetime, compared to the protocols TRANS [STP04] and T-LEACH [JHJ09]. The results are verified with different sets of parameter values. Better results of the proposed TCEER scheme are quite justified because the additional energy consumption due to the congestion obtained from the misbehavior of the faulty nodes are not considered in TRANS [STP04] and T-LEACH [JHJ09]. Again, proposed TCEER algorithm shows better results of 10% and 7% compared to TFCC (described in Chapter 5, Section 5.3) [ACS13] and TC-ACO (described in Chapter 6) [ASA13] respectively. Although, trust and congestion both parameters are considered in TFCC [ACS13], TC-ACO [ASA13] and TCEER protocol, the data routing algorithms are different. In TFCC [ACS13], Link State Routing Protocol is implemented, in TC-ACO [ASA13], it is done as per the Ant Colony Optimization whereas in TCEER, hop by hop routing on the basis of the Node Potential of the trusted nodes is

implemented. In future, the nature of congestion obtained due to the various security attacks in WMSNs may be studied. Different trust based congestion control schemes can be compared to get the improved energy efficient solution. The proposed TFCC and TCEER algorithms are tested on the network architecture having limited number of sensor nodes. The scalability issues may be studied further.

CHAPTER 6

Trust Based Congestion Aware Data Routing Using Ant Colony Optimization

6.1 Introduction

Wireless Sensor Networks have immense potential for a variety of applications in diverse fields, where employment of human beings is not feasible. It consists of a large number of randomly deployed sensor nodes, which are battery powered and resource constrained in terms of limited energy, circumscribed computational and communication capability, bounded memory and processing speed [IFA02], [ABV02] and [DCD04]. Since the turn of the 21st century, researchers throughout the globe have proposed various energy efficient routing algorithms to maximize the network lifetime of WSNs. Yet, a scope for improvement does exist. Sensor networks are generally operated in an idle mode and then suddenly become active in response to the detected event [CYW03]. When a large number of sensor nodes become active and try to transport data to the base station (BS), packet collisions and network level congestion are quite probable. Buffer overflow occurs due to the limited buffer size of the nodes, causing packet drops which in turn decreases network performance and throughput. Moreover, inexpensive sensor nodes are prone to failure and the faulty nodes create various security threats which aggravate the problem of congestion by diffusing useless packets, flooding with fake messages, intermittent jamming and/or retransmitting the same message

several times. The resulting effect is redundant computation and communication, causing wastage of energy resources and a decrease in network lifetime. If the problematic faulty nodes are eliminated from the data routing path, congestion would be minimized and this will enhance energy efficiency and network lifetime. To address this challenge, a congestion aware data routing scheme, based on the Ant Colony Optimization (ACO) mechanism is proposed, in which faulty nodes, also called malicious nodes, are detected and isolated by utilizing a trust based framework. Trust management is a relatively new idea which is used these days for detecting faulty nodes in order to establish a trustworthy data routing path from the source node to the BS [MOM08]. The concept of trust is basically borrowed from the human society, in which the sensor nodes monitor the behavior of their neighbors during previous data transfer operations through these nodes, on the basis of some parameters known as the Trust Metrics [ARM11]. In the proposed TC-ACO algorithm, congestion of the trusted node is computed by estimating the free space in the buffer place of the node and then ACO is utilized which factors in distance along with trust and congestion for energy efficient, trustworthy, optimal routing in Wireless Sensor Networks.

The rest of the chapter is organized as follows**:** Section 6.2 introduces Ant Colony Optimization (ACO) mechanism, in Section 6.3 the proposed routing algorithm is presented, simulation results and comparisons with other existing protocols are discussed in Section 6.4 and finally Section 6.5 concludes the paper.

6.2 Ant Colony Optimization Mechanism

Ant Colony Optimization (ACO) is the swarm intelligence based heuristic optimization technique, initially proposed by Marco Dorigo [MDG99], [MDM06] and [MDO92]. It is based on the behavior of real ants, while they search for their food in short routes from their nest to the location of the food [SOD09]. While travelling, an ant deposit pheromone in its path and the intensity of the pheromone decreases over time due to evaporation. The ants following shorter paths are expected to return earlier through the same path, compared to the ants in longer paths. Hence, the amount of pheromone deposited in the shortest path is more than that of the other paths. The new ants are subjected to follow the shortest path having more pheromone. In this way, the pheromone deposition in the shortest path increases whereas the other paths are lost due

to lack of pheromone. Finally, all the ants follow the shortest path. In the proposed scheme data packets are considered as artificial ants which are launched from the source node and find the optimal route towards the destination node (base station) in each cycle.

6.3 Proposed TC-ACO Algorithm

In this section, a novel trust based congestion aware energy efficient data routing scheme is proposed for Wireless Sensor Networks, in which Ant Colony Optimization (ACO) is utilized to maximize the network lifetime. It is considered that the sensor nodes are randomly deployed in the sensor field under free space propagation. The proposed algorithm works in two stages. In stage 1, the trust values and the congestion status of the nodes are calculated and trust-congestion metric is formed. In stage 2, the ACO algorithm, which utilizes the Trust Congestion Metric (TCM) and the distance metric, is implemented for data packet routing from source node to base station. The detailed operation of each stage is described in the following sections.

6.3.1 TC-ACO: Stage 1

In the proposed algorithm, stage 1 detects the misbehavior of the sensor nodes using the concept of trust. The trusted nodes having trust value above some pre-defined threshold level are identified and congestion levels are computed accordingly. The Trust Congestion Metric (TCM) is generated for the trusted nodes (also called valid nodes), for the data packet routing algorithm which is implemented in the next stage. The malicious nodes having trust value below the threshold level are not considered for data packet routing and so, the congestion metric is not computed for such nodes. This causes a reduction in the computation overhead and thereby enhances battery life time.

6.3.1.1 Trust Computation

The trust value of node i upon node j is calculated on the basis of three commonly used Trust Metrics namely, remaining node energy (N_e'), packet transmission ratio (P_{TR}') and packet latency ratio (P_L'). All the parameters are normalized so that the values belong to the range: [0,1]. $P_{TR}^{'}$ is defined as the ratio of the number of acknowledgement received from node j to the total number of packets sent from node i to node j. $P_L^{'}$ is the ratio of the latency of node j to the mean latency of the other nodes except node j, when data packets

are transmitted from node i. $N_e^{'}$ is defined as the average energy of the node i and node j. If E_i and E_j are the existing energy value of node i and node j respectively, then $N_e^{'} = (E_i + E_j) / 2$. The energy of a sensor node should be greater than or equal to the threshold value of E_{th} for transmitting data packets to its one hop neighbor in the radio communication range. Mathematically, the net trust of node i upon node j is calculated by the formula represented as,

$$T_{ij} = \frac{A_1 * N_e^{'} + A_2 * P_{TR}^{'} + A_3 * P_L^{'}}{A_1 + A_2 + A_3} \quad \ldots . (6.1)$$

where, A_1, A_2 and A_3 are the corresponding weights used for $N_e^{'}$, $P_{TR}^{'}$ and $P_L^{'}$ respectively such that $A_1, A_2, A_3 \in [0,1]$. A predefined trust threshold value (T_{TH}) is set on the basis of the application of the sensor networks [MZA09]. If $T_{ij} > T_{TH}$, the link between the node i and j is called a trustworthy link. Similarly, if $T_{ij} < T_{TH}$, the link is termed as an untrusted link. The nodes having no trustworthy link are called malicious nodes and those with at least one trusted link are called trusted nodes (valid nodes) that can take part in data packet routing.

6.3.1.2 Estimation of Node Congestion

The congestion level of a valid node is estimated with the help of the parameter known as the Congestion Index. It is assumed that each node maintains a queue for storing data packets in its buffer. As packets are transmitted from a particular node serially towards the next node, buffer space is cleared and the packets waiting in the queue go to the empty buffer space of the node. When the packet received rate of the node is greater than the packet transmission rate, queue length increases, buffer overflows, congestion level of the node increases. If a node is not able to clear the data packet in its queue, then it waits for a certain number of pre-defined cycles (say, WC_{max}) and holds the packets in each cycle, until the packets are finally dropped (at the end of WC_{max} cycles). The Congestion Index of the k^{th} node is computed by the expression as represented in Eq. (6.2) as,

$$CI_k = \frac{\bar{r}_{in}^k + Q^k(c-1) - \bar{r}_{out}^k}{\bar{r}_{in}^k + Q^k(c-1)} \quad \ldots . (6.2)$$

where, $Q^k(c-1)$ is the empty space left in the queue of the kth node till (c-1)th cycle. The parameters $\bar{r}_{in}^k$ and $\bar{r}_{out}^k$ are defined in Eq. (6.3) and (6.4) respectively as,

$$\overline{r_{in}^k} = \frac{\sum_{i=1}^{c-1}(N_{i,k}^A)}{c-1} \quad \ldots . (6.3)$$

$$\overline{r_{out}^k} = \frac{\sum_{i=1}^{c-1}(N_{i,k}^B)}{c-1} \quad \ldots . (6.4)$$

Here, $N_{i,k}^A$ = Number of packets forwarded to kth node in ith cycle, and

$N_{i,k}^B$ = Number of packets forwarded by kth node to the other nodes in ith cycle.

The congestion index of the trusted nodes, calculated by using Eq. (6.2), presents the node level congestion of the Wireless Sensor Network. It is calculated dynamically at regular intervals, depending upon the application of the network.

6.3.1.3 Computation of Trust Congestion Metric (TC_{ij})

The Trust Congestion Metric (TCM) of the trusted nodes (valid nodes), is computed by the Eq. (6.5), given below**:**

$$TC_{ij} = \alpha * CI_j + (1-\alpha) * T_{ij} \quad \ldots . (6.5)$$

where, node i and node j are considered as source node and destination node respectively. CI_j is the Congestion Index of the destination node and T_{ij} is the trust value of source node i upon the destination node j. The constant α is denoted as Trust Congestion Coefficient, belongs to [0,1].

6.3.2 TC-ACO: Stage 2

The proposed TC –ACO algorithm stage 2 implements data routing protocol using Ant Colony Optimization. It is assumed that the sensor nodes are deployed randomly in the entire sensor field, denoted by level 0, 1, 2…level L, (L+1)…. level (r-1), level r respectively, as shown in Fig. 6.1. The source node is considered as level 0 node. All nodes within the one hop neighbor of the source node in the radio communication range are denoted as the level 1 nodes. Similarly, all nodes within the one hop neighbor of the level 1 in the radio communication range are called the level 2 nodes and so on.

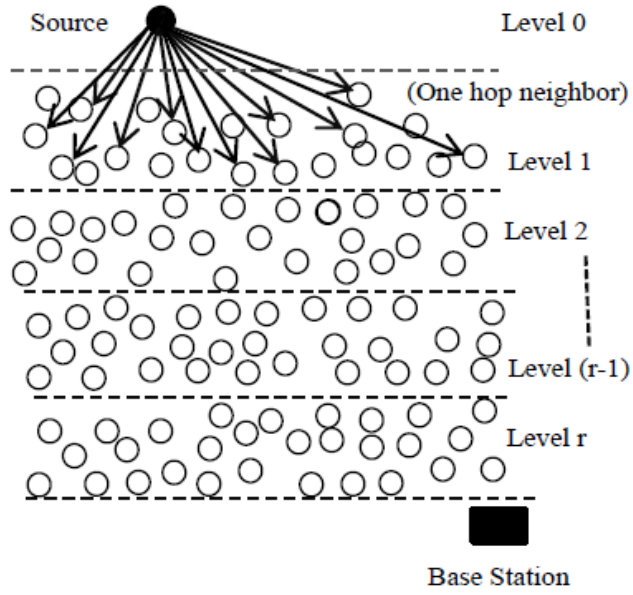


Figure 6.1: Random deployment of sensor nodes at various levels.

In the ACO based approach, each ant tries to find a path in the network at minimum cost. In the proposed scheme, ants (data packets) are launched from a source node, move hop by hop through the neighbor node in the next level and finally reach the destination node, called the base station. The pheromone values are stored in the node's memory. The choice of the next node is made according to the probabilistic decision rule that is biased by the pheromone associated with the node. The probability P_{ij} for transmission of data packets in optimal route from node i in level L to node j in level (L+1) is given by Eq. (6.6).

$$P_{ij} = \frac{\left(TC_{ij}\right)^{\beta_1} \cdot \left(\frac{1}{d_{ij}}\right)^{\beta_1} (\tau_{ij})^{\beta_2}}{\sum_{\forall k} (TC_{ik})^{\beta_1} \left(\frac{1}{d_{ik}}\right)^{\beta_1} (\tau_{ik})^{\beta_2}} \quad \ldots . (6.6)$$

where, ' k' represents the index number of the trusted nodes (valid nodes) in the level (L+1) and τ_{ij} is the pheromone value. The parameter $[(TC_{ij})^{\beta_1}.(1/d_{ij})^{\beta 1}]$ is the value of the heuristic, which is related to the trust congestion metric (TC_{ij}) and the distance (d_{ij}) between the node i and node j. This enables decision making according to the trust value, congestion status and the distance of the neighbor node. It means that if a node has high trust value, low congestion and is located near the source node, then it has the higher probability to be chosen. The coefficient β_1 and β_2 are the two parameters that control the relative weight of the pheromone trail and the heuristic value respectively. In ACO algorithm, each ant has deposited a quantity of pheromone $\Delta\tau^k(c)$ in its path given in Eq. (6.7), which is done by ant k at the end of the cycle c.

$$\Delta\tau^k(c) = \frac{N^k}{d^K}(c) \quad \ldots . (6.7)$$

N^k and d^k represents the total number of nodes and the total distance visited by the ant k respectively, at the end of the cycle c. The operation of pheromone evaporation is also accomplished in ACO algorithm. The evaporation constant ρ is defined to determine the weight of evaporation at the end of the cycle. The pheromone concentration τ_{ij} is updated as per the expression shown in Eq. (6.8),

$$\tau_{ij}(c) = (1-\rho).\ \tau_{ij}(c-1) + \Delta\tau_{ij}(c) \quad \ldots . (6.8)$$

Different variables of stage 2 of the proposed TC–ACO algorithm are listed in Table 6.1; the pseudo code of the data packet routing protocol is presented in Algorithm 6.1.

Table 6.1: List of Variables

Variable Name	Description
TC_{ij}	Trust Congestion Metric between node i and j
τ_{ij}	Pheromone concentration on the link connecting nodes i and j
d_{ij}	Distance between nodes i and j
P	Evaporation constant ($\rho \in [0,1]$)
N^k	Total number of nodes visited by the ant k at the end of the particular cycle.
d^k	Total distance visited by the ant k at the end of the particular cycle.
β_1 and β_2	Constant parameters and each $\in$ [0,1]

Algorithm 6.1: Data packet routing algorithm.

Input: Trust Threshold Level, Trust Congestion Metric

Output: Optimal Route

Begin

For each node i in L^{th} level,

do

Step 1: Find all valid node "j" in level (L+1). The nodes satisfying the condition Tij > Tth are valid nodes

Step 2: Compute probability of packet transmission from node "i" to node "j" defined as Pij

Step 3: Arrange valid nodes in descending order based on the value of Pij , store in matrix X.

Step 4: Initialize m =1

While ($Ei \geq Eth$ && $m \leq size(X)$)

// Forward packet from node i to X(m)

// Update energy of both nodes.

If (Queue ($X(m)^{th}$ node) is full OR ($E_{X(m)} < Eth$))

m = m+1;

end –if

end – while

end – do

End

)

6.4 Simulation Results

In this section, the merits of the proposed TC-ACO scheme have been investigated through MATLAB simulations. An arbitrary network comprising of 50 homogeneous nodes are deployed randomly into a field of dimension 200 m X 200 m. The distances of the nodes from the base station (BS) are taken constant throughout the experiment. It is assumed that the nodes are connected within the network in different levels as per the first order radio model [WRH00]. The graphical view of the packet routing, when 20 data packets are transmitted from the source node (marked as 27), towards the BS is presented in Fig. 6.2; the packets are routed in five different paths, of which 12 packets are transmitted through the optimal route, as obtained by the ACO algorithm, which is marked in red color in Fig. 6.2. During the next cycle of data transfer, some other optimal route would be selected dynamically, on the basis of the Trust Congestion Metric of the corresponding nodes.

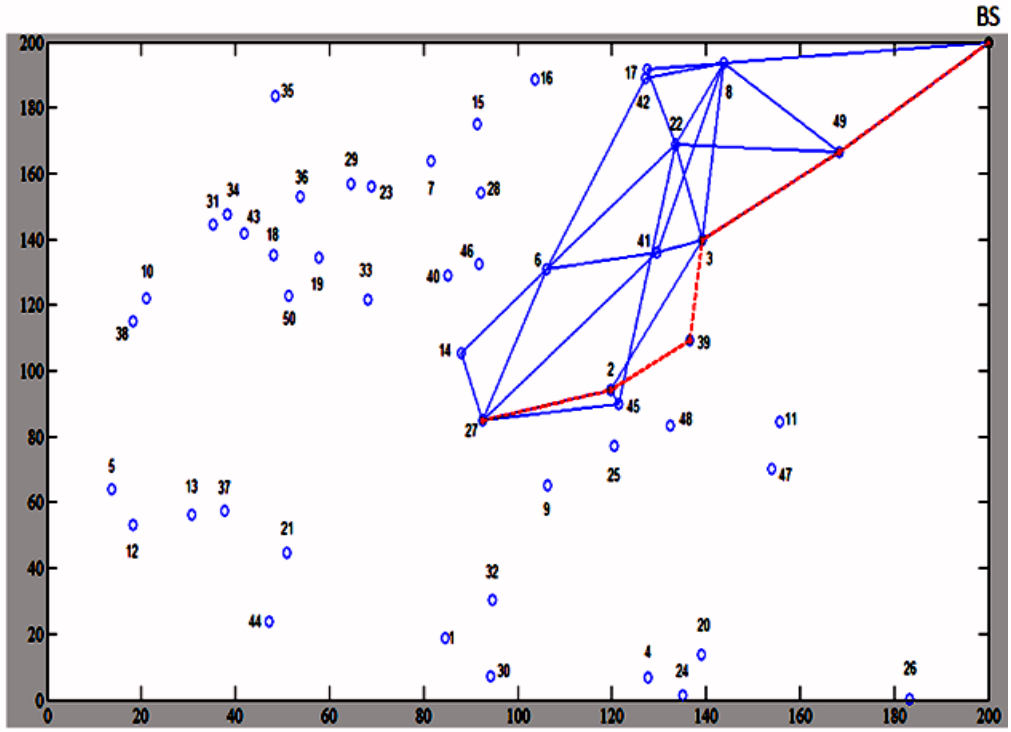


Figure 6.2: Routing of 20 packets from source node 27 to the BS.

The performance of the proposed TC-ACO algorithm has been tested through rigorous MATLAB simulations, by varying node energy and trust threshold level. The results obtained for the initial node energy of 1.0 Joule per node and trust threshold level equal to 0.5 have been presented. The proposed protocol is compared with the existing algorithms such as TRANS [STP04], MRP [JYM10] and TFCC [ACS13] respectively. The Table 6.2 represents the number of rounds verses percentage of dead nodes that are observed for the initial energy of 1.0 Joule/node. The graphical representations are shown in Figs. 6.3 and 6.4 respectively, where the percentage of dead nodes is plotted along the x axis and the number of rounds is plotted along the y axis of the graphs. The simulations results indicate that the proposed TC-ACO scheme provides higher network lifetime compared to the other similar protocols and thereby outperform its peers. It is quite justified since TRANS [STP04] and MRP [JYM10] do not consider the additional energy consumptions due to the congestion that are obtained from the misbehavior of the faulty malicious nodes. Although, trust and congestion both are considered in TFCC [ACS13] and TC-ACO, the data routing methods of the two schemes are different. In the previous one, the data routing is based on the Link Stare Routing Protocol (LSRP), whereas in the proposed TC-ACO scheme data routing is done by utilizing Ant Colony Optimization (ACO).

Table 6.2: Performance analysis at initial energy 1.0 Joule / node

Number of Rounds	Protocols	Percentage of Dead Nodes						
		1%	10%	20%	30%	40%	50%	60%
	TRANS	1965	2132	2342	2596	2701	2910	3197
	MRP	2087	2221	2378	2601	2895	3020	3304
	TFCC	2223	2365	2455	2673	2812	3108	3276
	TC-ACO	2406	2677	2818	2997	3108	3285	3566

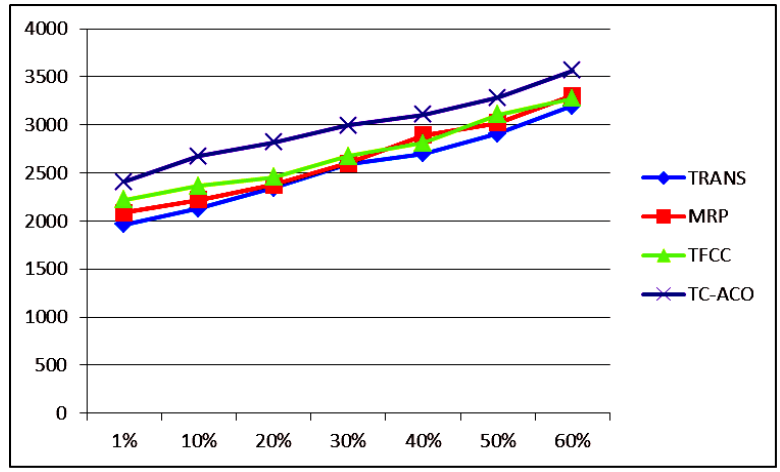


Figure 6.3: Performance analysis for initial energy 1.0 J/node.

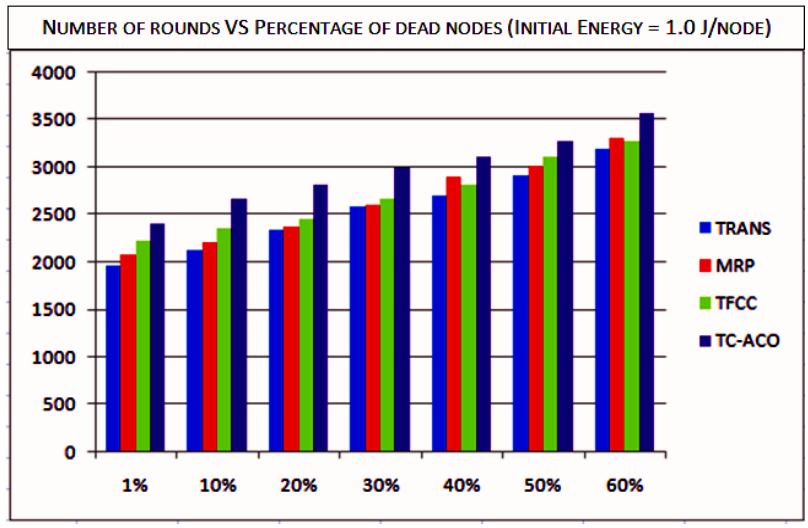


Figure 6.4: Bar graph analysis for initial energy 1.0 Joule/node.

Next, the proposed TC-ACO algorithm is compared with the previous work, GACCTR (Chapter 4), where the routing path is selected dynamically by using Genetic Algorithm. The simulation experiments in MATLAB are performed with TC-ACO and GACCTR respectively under same network configurations. The throughput in Kbps is observed by varying the number of sensor nodes (5, 10, 15, 20 and 25 nodes) deployed in the sensor field of dimension 200 m X 200 m. The parameter throughput is defined as the number of data packets transmitted from the source node to the base station during a particular time frame. Fig. 6.5 represents the variation of throughput with the number of nodes deployed for TC-ACO and GACCTR algorithms. In the simulation with the package of 256 bytes, both of them have shown almost similar performance, when the network is composed of

5, 10 and 15 nodes. However, when the size of the network is increased to 20 and 25 nodes, the throughput of TC-ACO shows better results compared to GACCTR.

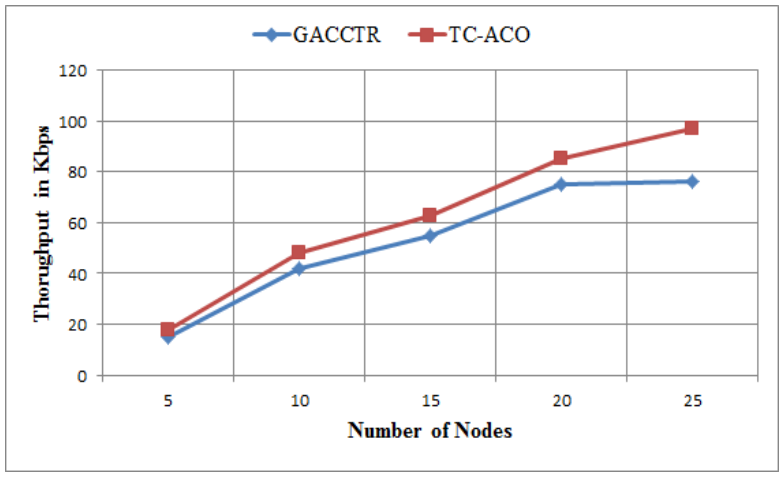


Figure 6.5: Variation of throughput with the number of nodes deployed.

6.5 Summary

In the proposed TC-ACO algorithm a novel trust based congestion aware data routing scheme for Wireless Sensor Networks is presented, in which the optimum route for the data packet transfer is dynamically selected on the basis of the trust value, congestion level and the inter-nodal distance of the sensor nodes. The proposed protocol considers the impact of the misbehavior of the faulty malicious nodes on the network congestion and thereby, tries to minimize its effect on the data packet routing. The Ant Colony Optimization (ACO) technique is utilized to select the dynamic routing path, which shows significant improvement in network lifetime compared to the similar existing algorithms. It is very relevant research topic, because in large Wireless Sensor Networks, especially for Wireless Multimedia Sensor Networks, a large volume of data is needed to transfer hop by hop from one node to another, where congestion is practically unavoidable.

CHAPTER 7

Congestion Aware Protocol Suite Using Trust Based Framework

7.1 Introduction

Wireless Sensor Networks have immense potential to revolutionize many segments of our life and economy in diverse fields, from military applications to environmental monitoring and conservation, health care, security system, industrial automation, transport system management and many more. Unlike centralized systems, WSNs are subject to a unique set of resource constraint characteristics such as finite battery power, limited communication bandwidth, computation capabilities and processing speed [IFA02], [DCD04]. In order to preserve battery power, randomly deployed sensor nodes in WSNs are generally operated in sleep mode and then suddenly become active in response to the detected event [CYW03]. When large number of active sensor nodes try to transport data simultaneously to the base station, packet collisions and network congestion are very common. Again, new generation of Wireless Sensor Networks consist of multimedia sensor nodes having capability to retrieve multimedia data like still images, voice, audio and video streams as well as scalar sensor data from the physical environment. The heavy traffic load of the multimedia data from source to sink is often greater than the available capability of the network which creates network

congestion. Moreover, inexpensive, fault prone sensor nodes sometimes behave like malicious nodes and create various types of security threats that aggravate congestion problems many folds by diffusing useless packets, flooding with fake messages, intermittent jamming and/or retransmitting same message several times. This generates redundant computations and communications which in turn decrease network lifetime through wastage of battery power. Thus, network congestion is one of the most serious problems in Wireless Sensor Networks that need to be controlled for optimization of network lifetime and throughput. Most of the existing data routing algorithms for Wireless Sensor Networks do not consider the congestion factor and the adverse effect of the malicious nodes in the sensor field. Trust management is a relatively new idea which is used these days for detecting misbehavior of the malicious nodes to establish trustworthy data routing path from the source node to the base station. In this paper, we have integrated three of our already existing data routing algorithms into a single congestion aware protocol suite with trust based framework, for energy efficient routing in Wireless Sensor Networks, where malicious nodes are identified and blocked so that they could not participate in further data routing activities. The additional energy consumption due to the malfunctioning activities of the malicious nodes are significantly reduced which in turn enhances energy efficiency and lifetime of the network. The adaptive selection of the routing schemes in each round depends upon the estimation of the efficiency of the protocols, known as TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15] respectively. The proposed Congestion-Energy-Trust Protocol Suite (CET-PS) algorithm is simulated in MATLAB considering arbitrary sensor networks which shows up to 18% improvement in network lifetime compared to the protocols, if implemented in standalone mode.

The rest of the chapter is organized as follows**:** In Section 7.2, the brief discussions of some related existing works are included. The proposed algorithm is presented in Section 7.3, the simulation results are discussed in Section 7.4 and finally the chapter is summarized in Section 7.5.

7.2 Related Works

Trust based congestion aware routing in WSNs is a relatively new research topic and has not been addressed in literature to a great extent. Although a lot of energy efficient routing protocols are available, most of them do not consider network security, role of the faulty nodes and the problem of congestion in their ambit. Congestion control protocols for WSNs are discussed in CODA [CYW03], ESRT [YSO03], PSFQ [CYW05], PCCP [CWK06], QCCP [MHY08], SUIT [CSZ14], WCCP [SMA13] and HTAP [CSV13] but the effects of malicious nodes in network congestion are not addressed there. In the FCC protocol [MZA09], Zarei et al. propose a Fuzzy based trust estimation for congestion control in WSNs. FCCTF protocol [MZA10] is basically a modification of FCC, in which the Threshold Trust Value is used for decision making. In TFCC [ACS13], traffic flow from the source to base station is optimized by implementing Link State Routing Protocol which provides improvement in network throughput. TC-ACO [ASA13] is a trust based congestion aware routing protocol for WSNs, where Ant Colony Optimization is utilized for data packet routing.

7.3 Proposed CET-PS Algorithm

In this section, a new trust worthy congestion aware data routing scheme is proposed which implements an adaptive synergy of the TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15] algorithms to achieve energy-efficient routing of data packets to the base station. The data routing period is subdivided into a set of rounds, with one of the three afore-mentioned protocols operating in each round. Initially, all the three schemes take charge in a round each, after which the data routing algorithm is chosen adaptively, based on a certain decision making algorithm. A brief description of the said decision making protocol along with succinct descriptions of TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15] schemes are given below.

7.3.1 TFCC Algorithm

TFCC algorithm is discussed in details in Chapter 5. In TFCC [ACS13], Trust Metrics of all nodes are derived by using a two-stage Fuzzy Inferencing Scheme. The traffic flow from source to base station is optimized by implementing the Link State Routing

Protocol (LSRP). The congestion of the sensor nodes is controlled by regulating the rate of traffic flow on the basis of the priority of the traffic.

7.3.2 TC-ACO Algorithm

The detailed discussions of TC-ACO are included in Chapter 6. In TC- ACO [ASA13], trust value and congestion status of the nodes are calculated and Trust Congestion Metric is formed. The Ant Colony Optimization mechanism is utilized for the implementation of the data packet routing from the source node to the base station.

7.3.3 TCEER Algorithm

TCEER protocol is discussed in details in Chapter 5. In TCEER [ACS15], the parameter Node Potential are computed on the basis of the trust value, congestion status, residual energy and the distance of the node from the base station using Fuzzy Logic Controller. The source node selects node with highest potential in its one hop radio range for data packet transmission.

7.3.4 CET-PS Algorithm

In the proposed Congestion-Energy-Trust Protocol Suit (CET-PS), the initial run of the data routing is described as phase I. Three random numbers r1, r2 and r3 are assigned for the three algorithms TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15] respectively. Next, run TFCC [ACS13] or TC-ACO [ASA13] or TCEER [ACS15], one time each, based on the increased value of r1, r2 and r3. It undergoes the following algorithm as given below:

```
do
    r1 = rand (1,1);
    r2 = rand (1,1);
    r3 = rand (1,1);
   while  (r1 = = r2 || r2 = = r3 || r1 = = r3)
end
```

For example, let the values of the random numbers arbitrarily set as r1 = 0.35, r2 = 0.44 and r3 = 0.2. It implies that the first round of the data routing would be on the basis of TC-ACO [ASA13] and the next two rounds are as per TFCC [ACS13] and TCEER [ACS15] respectively. For each round some performance metrics are computed that are denoted as Energy Efficiency (EE), Composite Energy Efficiency (CEE), Packet Transmission Efficiency (PTE), Composite Packet Transmission Efficiency (CPTE) and Composite Protocol Efficiency (CPE) respectively. The description of each parameter is discussed in the following sections. The step by step calculations of the parameter finally compute the value of CPE at the end of each round. The proposed CET-PS protocol implements the routing algorithm with highest CPE for the next round. The steps for calculation of CPE are given below:

- Calculate Energy Efficiency (EE)
- Calculate Composite Energy Efficiency (CEE)
- Determine the Fuzzified value of CEE
- Calculate Packet Transmission Efficiency (PTE)
- Calculate Composite Packet Transmission Efficiency (CPTE)
- Determine the Fuzzified value of CPTE
- Compute Fuzzy value of Composite Protocol Efficiency (CPE) from Fuzzy CEE and Fuzzy CPTE through rule base.

The block schematic diagram for the calculation of CPE using Fuzzifier and Defuzzifier is shown in Fig. 7.1. The crisp values of CEE and CPTE are the inputs to the Fuzzifier 1 and Fuzzifier 2 respectively. The corresponding Fuzzy values of CEE and CPTE are obtained at the outputs of the respective Fuzzifiers having suitable Rule Bases. The CPE Fuzzy value is obtained through the inference mechanism which is finally Defuzzified to get the CPE crisp value.

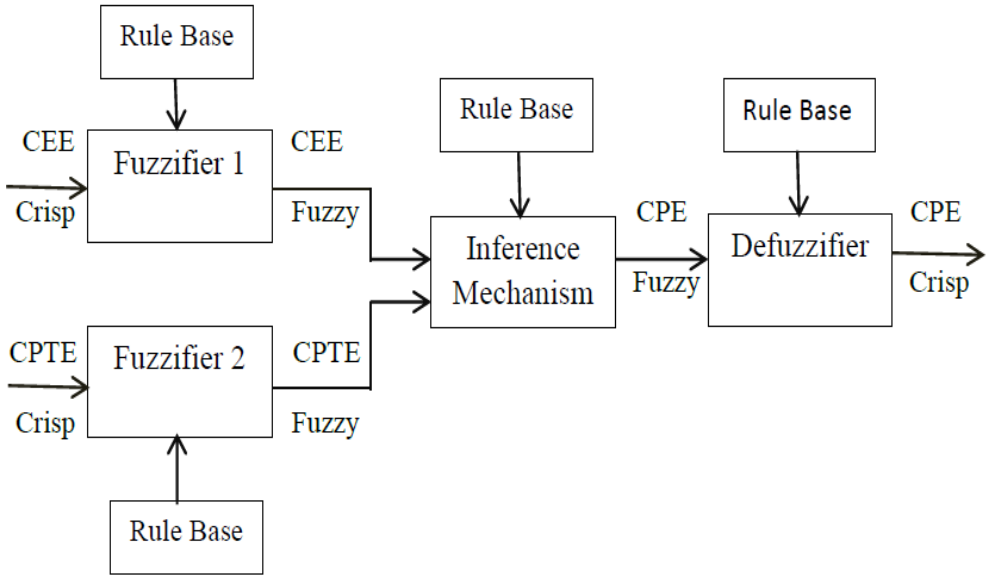


Figure 7.1: Block schematic diagram for the calculation of CPE.

7.3.4.1 Calculation of Energy Efficiency (EE)

The Energy Efficiency, EE_R is computed as per the Eq. (7.1) shown below.

$$EE_R \equiv \frac{E_1^T}{E_2^T} \quad \ldots . (7.1)$$

E_1^T and E_2^T are the energy of all trusted nodes before the round ensures and at the end of the round respectively. The value of a particular round for a protocol is denoted by R.

7.3.4.2 Computation of Composite Energy Efficiency (CEE)

Suppose, one of the three algorithms, namely TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15], have been adopted for "X" times, out of the last "N" rounds, The Composite Energy Efficiency is the parameter which is used to measure the performance of an algorithm as defined in Eq. (7.2)

$$\text{CEE} = \frac{EE_x + rEE_{x-1} + r^2 EE_{x-2} \cdots\cdots + r^{x-1} EE_1}{1 + r + r^2 + \cdots\cdots + r^{x-1}} \quad \ldots . (7.2)$$

$r \in (0,1)$ which represents the weightage of different rounds. The maximum weightage is given to the most current round denoted by Energy Efficiency EE_x.

7.3.4.3 Fuzzification of CEE

The Mamdani model of Fuzzification is used to get Fuzzified CEE. The value obtained from Eq. (7.2) is given to the input of the Fuzzifier. The five Fuzzified output states are identified as VLCEE (Very Low CEE), LCEE (Low CEE), MCEE (Medium CEE), HCEE (High CEE) and VHCEE (Very High CEE) respectively, which is obtained by using the rule base shown in Fig. 7.1.

7.3.4.4 Computation of Packet Transmission Efficiency (PTE)

The Packet Transmission Efficiency (PTE_R) for a particular round R is obtained by computing the ratio between the numbers of successful packet transaction to the total numbers of packet transaction.

7.3.4.5 Calculation of Composite Packet Transmission Efficiency

Let any particular algorithm such as TFCC, TC-ACO or TCEER, be run "y" times out of last "w" rounds. The PTE values of that particular algorithm are denoted as PTE_y, PTE_{y-1}, PTE_{y-2}…….PTE_2, PTE_1, where PTE_y is the most recent value. The Composite Packet Transmission Efficiency (CPTE) is represented by the formula as shown in Eq. (7.3).

$$\text{CPTE} = \frac{PTE_y + k.PTE_{y-1} + k^2.PTE_{y-2} + \cdots\cdots + k^{y-1}.PTE_1}{1 + k + k^2 + \cdots\cdots\cdots + k^{y-1}} \quad \ldots.(7.3)$$

Here, k represents the weightage of different rounds and k € (0,1). It is obvious that the maximum weightage (k = 1), is given to the most recent round having Packet Transmission Efficiency of PTE_y.

7.3.4.6 Fuzzification of CPTE

Fuzzification of CPTE has been done in similar way of CEE as described in Section 7.3.4.3. The input of the Fuzzifier is the CPTE which is obtained from Eq. (7.3). The five Fuzzified states are denoted as VLCPTE (Very Low CPTE), LCPTE (Low CPTE), MCPTE (Medium CPTE), HCPTE (High CPTE) and VHCPTE (Very High CPTE) respectively. The Fuzzified CPTE output is derived using the Rule Base as shown in Fig. 7.1.

7.3.4.7 Computation of Composite Protocol Efficiency (CPE)

The Fuzzy value of Composite Protocol Efficiency (CPE) is extracted from the Fuzzy CEE and Fuzzy CPTE using the Rule Base. The block diagram of the Fuzzy Inference Engine is depicted in Fig. 7.1, in which the Fuzzy CEE and Fuzzy CPTE are the two inputs and Fuzzy CPE is the output. The five types of Fuzzy CPE are denoted as VLCPE (Very Low Composite Protocol Efficiency), LCPE (Low Composite Protocol Efficiency), MCPE (Medium Composite Protocol Efficiency), HCPE (High Composite Protocol Efficiency) and VHCPE (Very High Composite Protocol Efficiency) respectively. The crisp value of CPE is obtained from the Defuzzifier by using the Rule Base. For each round, the value of CPE for the routing scheme TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15] are evaluated. The algorithm with highest crisp value of CPE is employed for the next round. In case of tie, that is, if the values of CPE for any two schemes are same, it is resolved as per the order established in initial run of CET-PS protocol.

7.4 Simulation Results

The proposed CET-PS protocol is tested in MATLAB, considering arbitrary network comprises of 50 homogeneous sensor nodes deployed randomly in a field of dimensions 100 m X 100 m. The distance of the nodes from the base station is taken constant. The variation of the number of rounds with time is compared with TFCC [ACS13], TC-ACO [ASA13] and TCEER [ACS15] respectively. More number of rounds is obtained in CET-PS which shows maximum 18% enhancement of network lifetime compared to the other protocols. The comparison graph showing the variation of number of rounds with time for different algorithms is presented in Fig. 7.2.

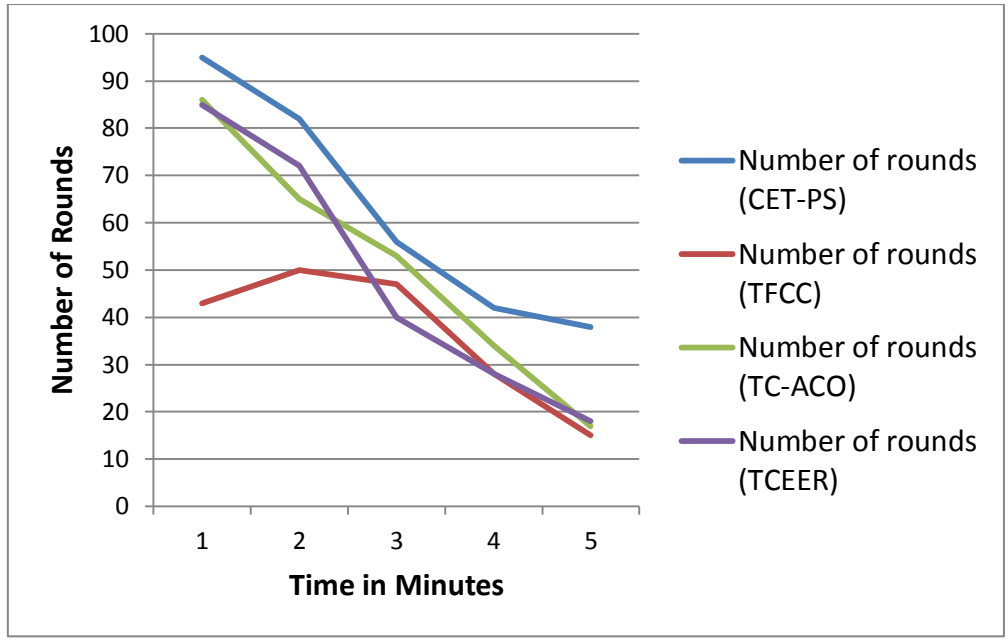


Figure 7.2: Comparison graph of various algorithms

7.5 Summary

In this chapter, a trust based congestion aware data routing protocol suite CET-PS is presented, in which three algorithms are deployed adaptively, one at a time, depending on the value of the parameter CPE, calculated at the end of each round. The proposed scheme is trustworthy since the malicious nodes are detected and eliminated from the data routing path. The congestion status of the node is considered during data routing followed by dynamic congestion control. The proposed scheme is evaluated using simulation and it is compared with the other similar algorithms. It has been found that CET-PS scheme exhibits higher network lifetime compared to its peers. It reduces energy consumption by decreasing additional communication obtained from the malicious nodes. In the previous proposed works described in this thesis, it is observed that the conventional data routing algorithms for WSNs are inferior than the trust based congestion aware data routing schemes in terms of energy efficiency and throughput. The simulation of the proposed CET-PS scheme shows satisfactory improvement over the previous proposed trust based congestion control algorithms when deployed in standalone mode.

CHAPTER 8

FPGA Implementation of a Fuzzy Coprocessor for Trust Based Congestion Aware Data Routing

8.1 Introduction

Wireless Sensor Network is one of the most discussed research topic throughout the globe since the last decade, due to its immense potential in different field of applications, which have significant benefits to the society. However, it has certain limitations in terms of battery power, memory space, communication bandwidth and computational capabilities [1DCD04], [IFA02], [JYB08]. Different aspects of WSNs have been studied and various new domains are still emerging with the advancement of technology. trust based congestion control in WSNs is one such emerging field of research. In WSNs, wireless channels are shared by various sensor nodes that have employed Carrier Sense Multiple Access (CSMA) protocols for getting access to the wireless medium. Typically, sensor nodes are remained inactive or in sleep mode for energy saving and then suddenly become active when something is detected in the surrounded areas [CYW03]. The resulting effect is generation of large and correlated impulses of data in the upstream direction from the source node to the sink. When large number of sensor nodes become active and try to transmit data to the sink simultaneously by sharing the available resources, there is a possibility of packet collision followed by

network congestion. This type of congestion in WSNs is called Link Level Congestion (LLC) or the Channel Congestion [ALG15], [AMH14]. On the other hand, if the data packet arrival rate exceeds the packet service time, buffer overflow may occur due to the limited buffer size of the sensor nodes, causing packet drops and retransmission of data packets, which in turn decreases network performances and throughput. It creates Node Level Congestion (NLC), alternatively termed as the Buffer Congestion [ALG15], [AMH14]. LLC and NLC are very common in resource limited WSNs, characterized by drastic drop in network throughput, packet loss, unacceptable packet delays, Quality of Service (QoS) deterioration, even in worst case, complete collapse in network performance. The present work, focused only on the NLC which is quantified by estimating buffer queue space of the sensor nodes. Moreover, WSNs are prone to security attacks due to its broadcasting nature and infrastructure less configuration. The inexpensive sensor nodes always have certain probability to behave as malicious nodes by sending fake messages, retransmitting the same message several times, jamming the network intermittently or flooding with HELLO messages [APJ04], [CKD03], [HCA03] and [TKD10]. It would aggravate network congestion many folds by additional computations, causing more energy consumptions and thereby decreases network lifetime. The congestion obtained because of the faulty behavior of the malicious nodes could be minimized, if the malicious nodes are detected and isolated from the normal activities of the network. But, the conventional security and cryptographic protocols are computation intensive and not suitable for resource constraint WSNs [SSB14]. The trust is the alternative concept for the detection of malicious nodes which is light weight as well as easy to implement in WSNs [MOM08]. It is defined as the degree of reliability or the level of confidence of one node on the other node in performing network related activities. It is a mathematical tool in which sensor nodes monitor the behavior of their one hop neighboring nodes to establish the degree of trustworthiness in forwarding packets, on the basis of some node characteristics known as the Trust Metrics (TM) [MOM08]. The examples of TM, commonly used in trust calculations are the number of data packet forwarded, number of control packet forwarded, latency in data transmission, remaining energy of the sensor nodes, packet address modified etc.

In WSNs, Fuzzy based calculations are often used because the data generated by the sensor nodes are imprecise in nature. But, the hardware implementation of Fuzzy Logic Controller in resource limited wireless sensor node is not easy and was not discussed much in literature. In the present work, a new trust based congestion control algorithm is proposed where dedicated **Fuzzy Coprocessor (FCP)** is implemented for Fuzzy calculations. The VLSI architecture of the FCP is proposed which is modeled in **Verilog Hardware Description Language (HDL)** and the results are verified in **Field Programmable Gate Array (FPGA)** using Xilinx Spartan 3 board. Typically, FCP is the special type of sensor node with integrated additional hardware for execution of fuzzy operations. The sensor nodes evaluate trust by considering Geometrical Mean based Trust calculation method [SSB11]. Next, the malicious nodes are identified by setting a predefined Trust Threshold (TTH) value. The sensor nodes other than the malicious nodes are termed as the trusted nodes. The proposed algorithm blocks the malicious nodes so that they could not participate in further network activities. It is assumed that a certain numbers of the trusted nodes are assigned as FCP, which are deployed in the sensor field in such a way that every trusted node can get access to at least one FCP in its neighborhood. The Link State Routing Protocol (LSRP) [AST89], [LLP07] is applied to the trusted nodes, to obtain all possible routes from the source node to the sink node. The LSRP is not applied to FCPs and they do not take part in data routing due to conservation of energy. The FCPs are only responsible for Fuzzy computations of the neighbor nodes. The trusted nodes of the routes compute their own congestion status dynamically by monitoring the local buffer queue space. The parameter called Complementary Congestion Index (CCI) is defined to quantify the congestion, which is calculated by the method described in our previous works [ACS13], [ACS15]. The Fuzzy logic based calculations are estimated in FCPs to get the parameter Trust Congestion Metric (TCM) of the nodes, which represent the condition of the nodes in terms of their trust value and the congestion status. The parameter called Route Trust Congestion Metric (RTCM) of each route is calculated and the route with the highest RTCM is selected as the best route from the source to sink considering trust and congestion. The performance of the proposed FCP architecture is tested and verified through MATLAB Fuzzy Toolbox simulations. Then, the comparison analysis of the proposed Trust-based Congestion-aware Routing with Fuzzy Coprocessor scheme

(TCR-FC) with the previous similar works are studied, which show up to 36.4% enhancement in network lifetime. Thus, the contributions of the present research work are listed as,

- Architecture design of the dedicated state-of-the-art Fuzzy Coprocessor (FCP).
- Proposal of a new trust based congestion aware data routing algorithm (TCR-FC) where FCP is implemented for Fuzzy computations.
- Comparison of the proposed algorithm with the existing similar works.

The rest of the paper is organized as follows. Section 8.2 provides a brief overview of the related works. Section 8.3 represents the proposed TCR-FC algorithm. The design and architecture of the Fuzzy Coprocessor is illustrated in Section 8.4. The simulation results and comparison with other protocols are presented in Section 8.5. Finally, the paper is concluded in Section 8.6.

8.2 Related Works

The traditional congestion control algorithms for WSNs, as discussed in [MAK14] and [JZL10], mainly consider buffer overflow and /or data packet collisions. Carrier Sense Multiple Access (CSMA) and Time Division Multiple Access (TDMA) are the Medium Access Control (MAC) approaches, primarily used to avoid data packet collisions in MAC layer. The protocol proposed in [BHK04], considers three mechanisms to mitigate congestion, such as hop-by-hop flow control, limit the source rate and prioritize MAC layer to drain out the congestion at the local nodes. It requires continuous monitoring of the parent node's activities which requires consumption of more energy and network resources. Some algorithms use cross layer approaches to control the buffer overflow. For example, CODA [CYW03] is an energy efficient Congestion Detection and Avoidance algorithm consisting of two mechanisms - open loop hop-by-hop back pressure and closed loop multisource regulation. Event to Sink Reliable Transport protocol, ESRT [YSO03] is a transport solution, developed to achieve reliable event detection with minimum energy consumption and congestion resolution at the sink. This protocol does not have any congestion control mechanism at the intermediate nodes. In PSFQ [CYW02], data is distributed from the source node at a relatively slow speed called "pump slowly" and allow the nodes to recover the missing segments from their

local immediate neighbors aggressively called “fetch quickly”. In PCCP [CWK06], the priority index of the node is considered and hop by hop congestion control in upstream direction is implemented. QCCP-PS [MHY08] represents a queue based congestion control protocol with priority support where queue length is used as an indication of the congestion degree. A Fuzzy based congestion control scheme is proposed in SUIT [CSZ14], where some packets of the frames are dropped to reduce congestion, causing lower but acceptable quality of the frame. In WCCP [SMA13], intermediate nodes monitor the queue length to detect congestion that is avoided by adjusting the sending rate at the source and by distributing the departing packets from the source. GMCAR algorithm [OBS12] represents a congestion avoidance routing protocol suitable for grided sensor networks. In HTAP algorithm [CSV13], a resource control algorithm is presented which consider alternative paths for the avoidance of congestion areas of the sensor networks.

The trust based congestion control is the new research trend that has been appeared in [ACS13], [ACS15], [MZA09], [MZA10] and [ASA13], respectively, where the misbehaviors of the malicious nodes are detected and isolated by using the concept of trust, for reducing network congestion. The Fuzzy based trust calculations are implemented in TFCC [ACS13], in which congestion status of the nodes is estimated dynamically by observing the buffer queue length. The adaptive data rate controller is used to optimize the traffic flow from the source node to the sink. In TCEER algorithm [ACS15], the parameter called **Node Potential** is evaluated, which is a function of the trust value, congestion status, node energy and the distance of the node from the sink. On the basis of the node potential value of the node, an adaptive hop-by-hop data routing algorithm is implemented in TCEER [ACS15]. In FCC [MZA09] and FCCTF [MZA10], Zarei et al. have proposed the Fuzzy Logic algorithms for trust based congestion control in WSNs. FCCTF [MZA10] is basically a modification of FCC [MZA09] protocol, where Trust Threshold value is used for decision making. TC-ACO [ASA13] is a trust based congestion aware routing protocol for WSNs, where Ant Colony Optimization is utilized for data packet routing. The trust based congestion control algorithms show better energy efficiency and throughput compared to the conventional congestion control protocols for WSNs.

The concept of Fuzzy Logic Controller is widely adopted in WSNs due to its simple approach based on the linguistic information and the ability to manage imprecise data obtained from the sensor nodes. However, hardware implementation of fuzzy logic controller in WSNs is not discussed much in literature.

8.3 Proposed Work

The objective of the proposed TCR-FC algorithm is to design the state-of-the art architecture of the Fuzzy Coprocessor unit and the selection of the best energy efficient data routing path, in terms of trust and congestion status of the sensor nodes. It is assumed that homogeneous sensor nodes are deployed in such a way that they can access the nearby FCP for Fuzzy calculations in one hop radio frequency range.

The deployment of FCPs are shown in Fig. 8.1, where nodes N1 – N8 would access the Fuzzy Coprocessor FCP1 for Fuzzy related calculations. Similarly, FCP2, FCP3 and FCP4 would perform the Fuzzy calculations of the nodes N9 - N16, N17- N24 and N25 –N32, respectively.

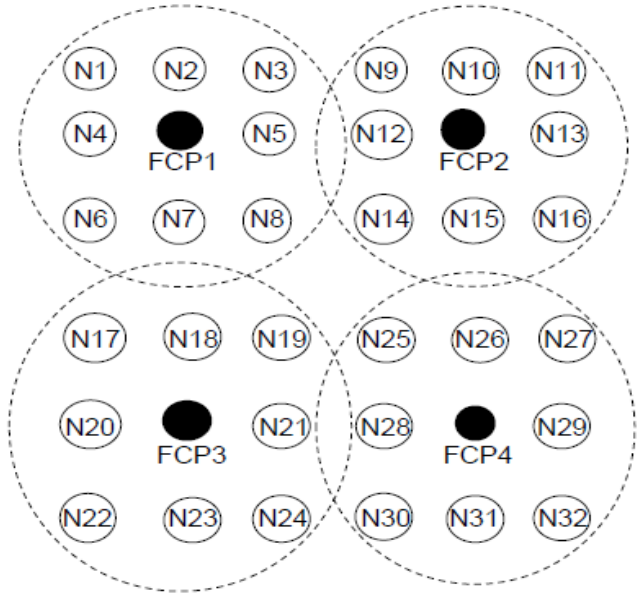


Figure 8.1: Deployment of Fuzzy Coprocessor in WSN.

The proposed work is divided into 5 steps as discussed below:

Step 1: Detection and isolation of malicious nodes using the concept of Trust.
Step 2: Implementation of LSRP to get all routes from the source node to the sink.
Step 3: Determination of congestion status of the nodes comprising the routes.
Step 4: Calculation of Trust Congestion Metric (TCM) of the node with the help of the Fuzzy Coprocessor.
Step 5: Calculation of Route Trust Congestion Metric (RTCM) of each route. Finally, select the route of highest Route Trust Congestion Metric (RTCM) as the best data routing path from the source node to the sink.

The detailed discussions of the above mentioned steps are given in the following sections.

8.3.1 Step 1

The two main activities of this step are evaluation of Trust values of the nodes and detection of malicious nodes on the basis of their trust values.

8.3.1.1 Calculation of Trust

In trust based congestion control, computation of trust plays a major role where detection of malicious node is done on the basis of the trust value of the node. Trust is broadly categorized as Direct Trust (DT) and Indirect Trust (IT) respectively. DT of a node depends on the direct observations made by the node on the behavior of its one hop neighbor during the previous data transfer through this node. The behavior of a node is dynamic in nature and is characterized by the Trust Metric (TM) values of the node. Hence, DT of a node with respect to its one hop neighbor is a function of Trust Metrics obtained during previous data transfer between these two nodes. On the other hand, IT of a node is calculated on the basis of the recommendations received from other trusted nodes in its neighborhood. Different methods for Trust calculation are available in the literature. However, in the present work, sensor nodes compute Trust value of its one hop neighbor by utilizing Geometric Mean based Trust calculation method [SSB11], which is used in TCEER protocol, described in chapter 5. The process of Trust calculation is dynamic in nature and the sensor nodes keep record of the dynamic Trust values of its

one hop neighbor nodes in its memory. For ease of understanding, the Geometric Mean based Trust calculation method is explained once again as given below.

Let the Direct Trust of node N_1 on node N_2 be denoted as DT_{N_1,N_2} which is calculated from the geometric mean of the various Trust Metrics for different events occurring between the nodes N_1 and N_2. The Direct Trust of node N_1 on node N_2 for k TMs is presented in Eq. (8.1) as,

$$DT_{N_1,N_2} = \left(\prod_{i=1 \; to \; k} (TM_i) \right)^{\frac{1}{k}} \quad \ldots . \; (8.1)$$

Let the Indirect Trust of node N_1 on node N_2 be denoted as IT_{N_1,N_2} and it is computed by the geometric mean of various Direct Trusts obtained from one hop neighbors of N_1. The Indirect Trust of node N_1 on node N_2, considering l number of neighbors of N_1 is presented in Eq. (8.2) as,

$$IT_{N_1,N_2} = \left(\prod_{j=1 \; to \; l} (DT_j) \right)^{\frac{1}{l}} \quad \ldots . (8.2)$$

Here, $DT_1, DT_2, DT_3, \ldots\ldots\ldots, DT_l$ are the Direct Trusts of N_2 with l number of neighbors of N_1. Let the overall trust of node N_1 on node N_2 be denoted as $T_{N1, N2}$ and it is represented by the equation (8.3) as shown below.

$$T_{N1, N2} = W_D * DT_{N1, N2} + W_I * IT_{N1, N2} \quad \ldots . (8.3)$$

Here, W_D and W_I are the weights to DT and IT respectively and $W_D + W_I = 1$. In some applications, DT has been given more importance than IT and accordingly the value of W_D is chosen higher than that of W_I. In the proposed scheme, we have considered equal values of W_D and W_I. This implies equal importance towards DT and IT respectively.

8.3.1.2 Detection of Malicious Nodes

A predefined Trust Threshold (T_{TH}) value is set which determines the earmark of malicious nodes. Higher the value of T_{TH}, higher is the security of the network. If the trust value of a node on a particular one hop neighboring node exceeds T_{TH}, then the node is referred to as the trusted node with respect to the former node. The link between these two nodes is called the trusted link. The same node may not be the trusted node with

respect to the other one hop neighbor. In this case, trust value of the node on that particular neighbor is below T_{TH} and the link between these two nodes is termed as the untrusted link. Transmission of data packets are allowed only through trusted links. The nodes without any trusted link are termed as malicious nodes and they cannot transmit data to the others.

As shown in Fig. 8.2, the nodes are connected with its one hop neighbors either by trusted link (represented by solid line) or by the untrusted link (represented by the dashed line). The nodes transmit data packets only through the trusted link.

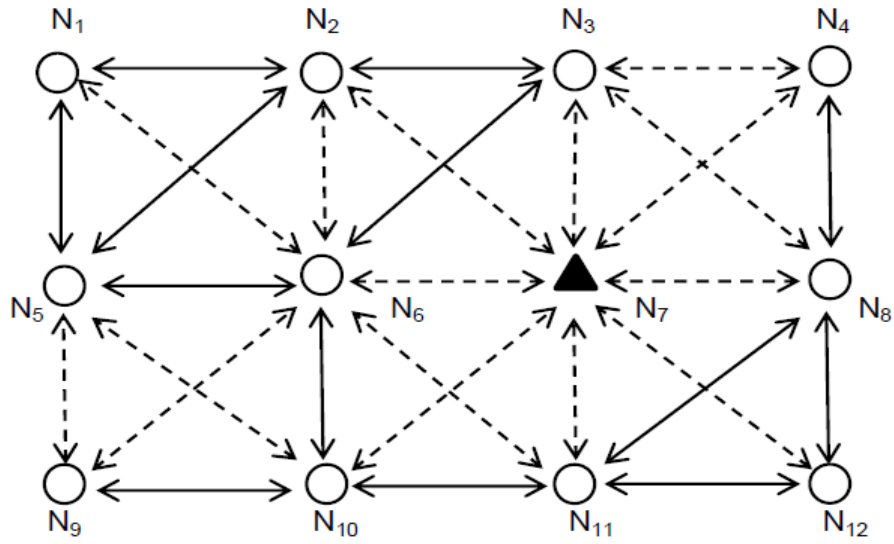


Figure 8.2: Pictorial representation of trusted and untrusted links.

It is seen that all nodes, except node N_7 (triangular shaped, marked in black), have at least one trusted link and they can transmit data to its neighbor through the trusted links. But, the node N_7 cannot transmit data to the other nodes since it does not have any trusted link with any other node. Thus, the node N_7 is isolated from the data routing path and is termed as the malicious node.

8.3.2 Step 2: Implementation of Link State Routing Protocol (LSRP)

In the proposed model, Link State Routing Protocol (LSRP) is applied to the trusted nodes as a basic routing protocol, to get all possible routes in the upstream direction from the source node to the sink. The LSRP is not applied to the malicious nodes, since the

communication is blocked by the untrusted link. Moreover, LSRP is also inactive to the Fuzzy Coprocessors that are only responsible for Fuzzy calculations of the nodes. At the first stage of LSRP, the trusted nodes send HELLO message to all other trusted nodes in its neighborhood through the trusted links. The sender node detects directly connected one hop neighbor by measuring the Received Signal Strength (RSS) obtained from the other nodes. Every node knows how to reach its directly connected neighbours and measure the delay or cost to each of its neighbors [AST89], [LLP07]. The node keeps record of the one hop neighbors in its memory. Next, the Link State Packet (LSP) is formed having the ID of the node who creates the LSP and the list of directly connected neighbours of that node, with the cost of the link to each one. The LSP is distributed by reliable flooding method where a node sends LSP to all of its directly connected nodes. On receiving LSP, the nodes on the other side send it out to all of their direct links. The process continues until the information has reached all the nodes in the network [LLP07].

8.3.3 Step 3: congestion status of the nodes comprising the routes

The proposed TCR-FC model concentrate only on the Node Level Congestion of the sensor nodes that can be measured by monitoring the buffer queue length of the node. It is assumed that each node maintains a buffer space for storing its own sensed data as well as the data obtained from its directly connected one hop neighboring nodes. The parameter called Complementary Congestion Index (CCI) is defined (refer to section 8.3.3.1), which is a function of the buffer queue length of the sensor nodes. Since malicious nodes are isolated and blocked from the data routing activities, CCI calculation for malicious nodes are not required. This reduces energy consumption and the computation overhead of the system.

8.3.3.1 Calculation of CCI

The Complementary Congestion Index (CCI) of the sensor node indicates congestion status of the node under consideration. Two predefined threshold values C_{Th}(Min) and C_{Th}(Max) are set in the buffer queue length of the node. Let, at a particular time instant, $Q_s(k)$ be the buffer queue length of the k^{th} node. Hence, at that time instant, any one of the following three conditions may happen.

Condition 1: If $Q_s(k)$ exceeds the limit $C_{Th}(Max)$, then the k^{th} node is termed as the congested node.

Condition 2: If $Q_s(k)$ is below the threshold value of $C_{Th}(Min)$, then it is implied that the congestion is not reported yet in the k^{th} node.

Condition 3: For $C_{TH}(Min) \leq Q_s(k) \leq C_{TH}(Max)$, medium level of congestion is reported in the k^{th} node.

The CCI value is quantified from the conditions of the buffer queue length as per the formula given in Table 1, which is explained in our previous work TCEER [ACS15]. Here, I_K and I_K' represent Congestion Index (CI) and Complementary Congestion Index (CCI) of the node respectively and $I_K' = 1 - I_K$.

Table 8.1: Formulae for computation of CCI

$Q_s(k)$	I_k
$Q_s(k) \leq C_{TH}(Min)$	$\in$ (where $\in$ is a small quantity)
$C_{TH}(Min) \leq Q_s(k) \leq C_{TH}(Max)$	$(1-\in)\left(\frac{Q_s(k)-C_{TH}(\min)}{C_{TH}(\max)-C_{TH}(\min)}\right)+\in$
$Q_s(k) > C_{TH}(Max)$	1

From Table 8.1, it is clear that high congestion in the node implies low CCI value and vice versa. The CCI calculations are updated in the sensor node after Δt seconds and it is recorded dynamically in the memory space of the corresponding sensor nodes.

8.3.4 Step 4: Calculation of TCM at the Fuzzy Coprocessor

The data obtained for the trust values and the congestion status of the sensor nodes are imprecise in nature. So, Fuzzy Logic is the best suitable method for the estimation of the parameter called Trust Congestion Metric (TCM) of the node which provides the overall node condition in terms of its Trust and congestion status. But, implementation of Fuzzy Logic Controller in individual wireless sensor node is very difficult due to the limitations of the resources. Considering that, the TCR-FC algorithm has proposed the concept of dedicated Fuzzy Coprocessor (FCP) that is used for the Fuzzy based calculations of the nodes. The FCP monitors the Trust and CCI values of the sensor nodes in its one hop

radio communication range at regular time interval of Δt, computes the TCM of the nodes in its neighborhood by using Fuzzy logic and then returns the TCM crisp value to the corresponding node for further actions.

8.3.5 Step 5: Calculation of Route Trust Congestion Metric (RTCM)

The Route Trust Congestion Metric (RTCM) of each route from source to sink is calculated by multiplying the TCM of all nodes comprising the route. It is given in the formula shown in Eq. (8.4).

$$RTCM(r) = \prod_{q=1}^{N} TCM(q) \quad \ldots . (8.4)$$

where, RTCM (r) is the Route Trust Congestion Metric of the r^{th} route, TCM (q) is the Trust Congestion Metric of the q^{th} node of the r^{th} route. Fig. 8.3 represents the block schematic diagram showing the functions of FCP. Let us consider that the nodes N1 and N2 be the trusted sensor nodes in the routing path that have calculated their Trust and CCI values by the methods as discussed in step 1 and step 3 of the proposed algorithm. The Fuzzy based calculations of the parameter TCM is performed at nearby FCP and finally, the FCP returns the TCM (N1) and TCM (N2) crisp values to the nodes N1 and N2 respectively.

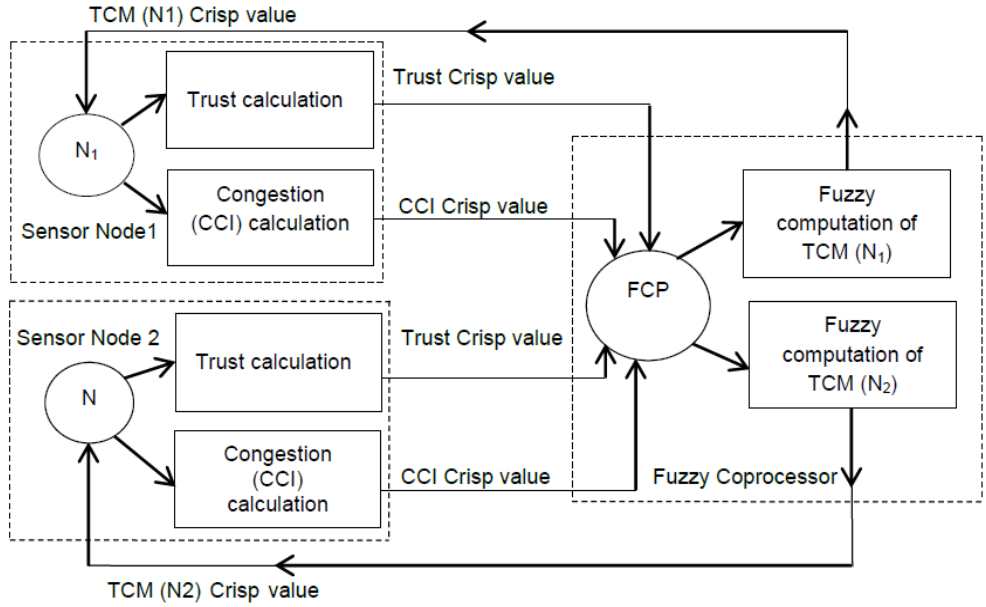


Figure 8.3: Block diagram showing the functions of Fuzzy Coprocessor.

8.4 Architecture of the Proposed Fuzzy Coprocessor

The proposed Fuzzy Coprocessor architecture consists of four components, namely Fuzzifier, Rule Base, Fuzzy Inference System (FIS) and Defuzzifier respectively. The general block diagram of the Fuzzy Coprocessor for two input parameters (trust and congestion) is shown in Fig. 8.4. The membership values and inferences are generated in the Fuzzifier from the input crisp values. The input interval is divided into different fuzzy sets where triangular membership functions are used. In the proposed algorithm, the input interval for trust is divided into four Fuzzy sets, namely VLT (Very Low Trust), LT (Low Trust), MT (Medium Trust) and HT (High Trust) respectively. Similarly, the input interval for CCI is also divided into four fuzzy sets as represented by VLC (Very Low CCI), LC (Low CCI), MC (Medium CCI) and HC (High CCI) respectively. It is to be noted from the definition of CCI that high congestion means low CCI. For example, VLC means highly congested node whereas HC represents the node with very low congestion. The pictorial representation of the triangular input membership functions are depicted in Fig. 8.5.

Fuzzy Inference System (FIS) generates Fuzzy output membership functions in association with the Fuzzy Rule Base. The Fuzzy output membership functions are denoted as VLTCM (Very Low TCM), LTCM (Low TCM), MTCM (Medium TCM) and HTCM (High TCM) respectively, corresponding to the output parameter called Trust Congestion Metric (TCM). Fig. 8.6 shows the pictorial representation of the Fuzzy output membership functions for the parameter TCM. As shown in Figs. 8.5 and 8.6, the membership functions are symmetrical and the slope is same for all the Fuzzy sets. The four consequences of the Fuzzy output membership functions are shown in Table 8.2, which would be inferred by the FIS unit according to the Fuzzy Rule Base. As shown in Table 8.2, there are 16 rules which represent the sensor node condition in terms of Trust and congestion status. For example, if a node has VLT (Very Low Trust) and VLC (Very Low CCI), then it has VLTCM (Very Low TCM) as output membership function. Similarly, sensor node with HT (High Trust) and HC (High CCI) generates HTCM (High TCM) as output membership function which is the most desirable condition.

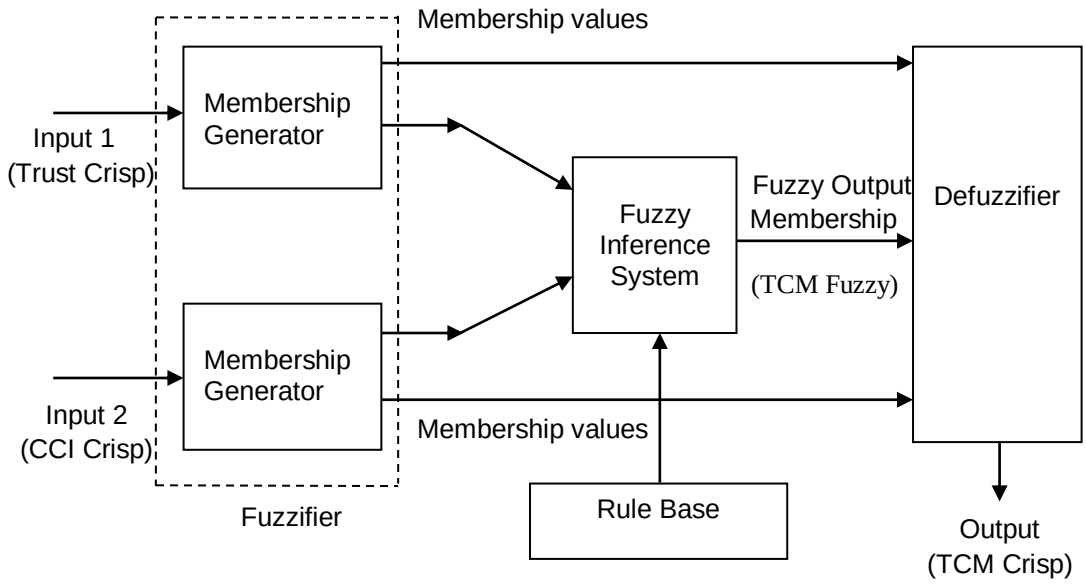


Figure 8.4: Block diagram of the proposed Fuzzy Coprocessor.

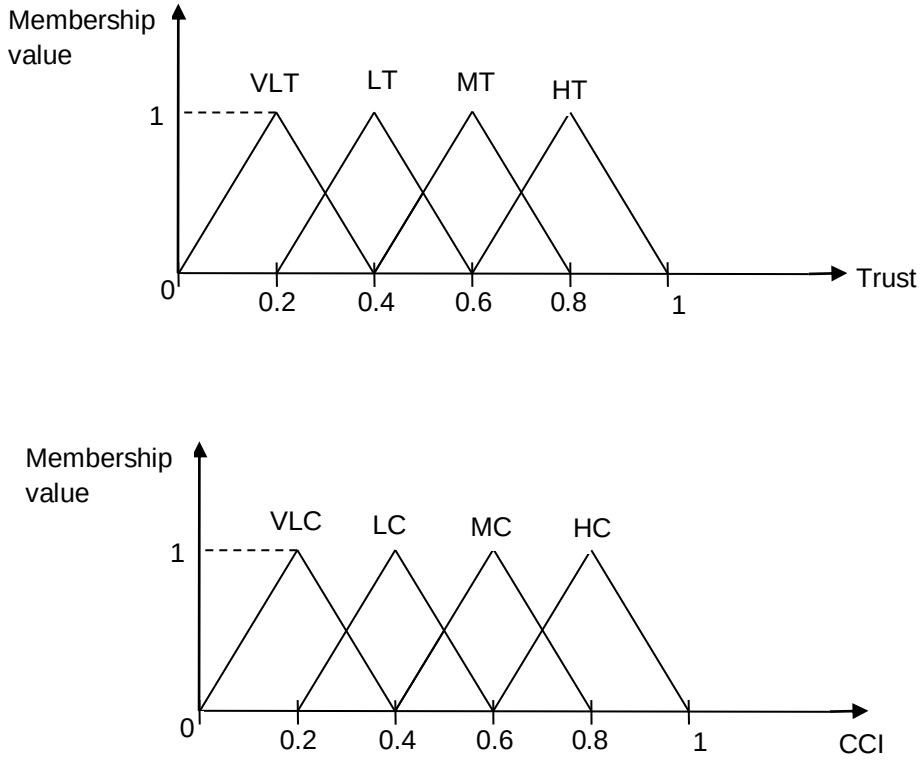


Figure 8.5: Triangular membership functions for the input variables .

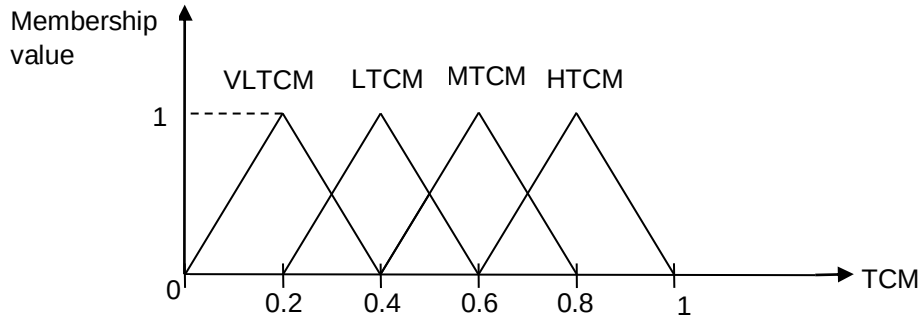


Figure 8.6: Triangular Membership Functions for the Output Variable TCM

Table 8.2: Fuzzy Rule Base

CCI → / Trust ↓	VLC	LC	MC	HC
VLT	VLTCM	VLTCM	LTCM	LTCM
LT	VLTCM	LTCM	LTCM	LTCM
MT	LTCM	MTCM	MTCM	HTCM
HT	MTCM	MTCM	HTCM	HTCM

The architecture of the proposed FCP is presented in Fig. 8.7; the FCP accepts crisp values of congestion parameter (CCI) and Trust as the two inputs from the sensor nodes. Comparators in the Fuzzifier block decides the Fuzzy sets to which the input value belongs to and store it in the memory locations in encoded format (00 for VLT, 01 for LT and so on). For example, we consider that a node sends request to its nearby FCP for fuzzy calculations with input Trust value of 0.3. Then the set of comparators decide the region where it lies. In this case, 0.3 is in between 0.2 to 0.4, as shown in Fig. 8.5. If the crisp value is greater than the mean value, the difference between the higher boundary and the crisp value will be stored. On the other hand, if the crisp value is less than the mean value, then the difference between the crisp value and the lower boundary value will be loaded in the memory. As 0.3 belongs to VLT and is greater than 0.2, so in the first location 0.1 (0.4 - 0.3 = 0.1) is loaded in the Value Memory Unit and 00 is loaded to the Fuzzy Set Memory Unit, shown in Fig. 8.7. Similarly, in the LT set, as it is less than 0.4, the value 0.1 (0.3 - 0.2 = 0.1) is stored in the second location in the Value Memory Unit and 01 is loaded to the Fuzzy Set Memory Unit. The same is applicable for the input

CCI crisp value too. Thus, the memory locations get filled up as shown in Table 8.3. Again, the Fuzzified membership values are also stored in the memory locations corresponding to the Fuzzy set of congestion (CCI) as well as Trust. For Fuzzification, the membership value is calculated multiplying by 5. Since, the membership value is the height corresponding to each input, the desired membership value is obtained by multiplying the slope (in this case, it is 5) with the base of the triangle which is already stored in the memory. For multiplication, add and shift method is used to reduce the computation overhead.

Table 8.4 shows the result after multiplications. Next, the Rule Base (represented in Table 8.2) is applied to the Fuzzy sets and the Fuzzy set values are sent to the Comparator (Min) unit. The smallest of the Comparator (Min) output is kept in an Intermediate Register. The Rule Base decides the Fuzzy output set and provides the address of any one of the four memory locations allocated for the membership values of the output Fuzzy sets. For example, the first memory location is used to store the output Fuzzy set value of VLTCM, the second location is used to store LTCM and so on. The output of the Rule Base is connected to the Control unit which provides the control signal. The value stored at the Intermediate Register is then sent to the Comparator (Max) unit along with the value located at the address of the memory location generated by the Rule Base. The greater of the two is again stored at the memory location generated by the Rule Base. This is Min-Max method of Fuzzification called Mamdani rule, which is implemented to get the Fuzzy output values**.** The membership values stored in four memory locations are then sent to the Weightage Block for Defuzzification of the output. Since the Fuzzy sets are symmetric in nature, the Weighted Average method is used for Defuzzification and the results are almost same as calculated by the Centroid Method for Defuzzification.
The numerator and the denominator are kept at the X and Y registers respectively. Thus, X contains the value given as (0.2*VLTCM + 0.4*LTCM + 0.6*MTCM + 0.8*HTCM), whereas Y contains the value (VLTCM + LTCM + MTCM + HTCM). The Division block performs the division of X/Y by non-restoring algorithm and produces final crisp output value as,

$$\text{Defuzzified TCM value} = \frac{\sum_{i=1}^{i=4} 0.2*i*mem[i]}{\sum_{i=1}^{i=4} mem[i]} \quad \ldots . (8.5)$$

where, mem[i] denotes the membership value stored in the i^{th} memory location after applying rules from the Rule Base. Thus, at the output of the Fuzzy Coprocessor, the crisp values of Trust Congestion Metric (TCM) are obtained.

Table 8.3: Contents of memory locations

Congestion		Trust	
Fuzzy set	Corresponding Value	Fuzzy set	Corresponding Value
00	0.1	00	0.1
01	0.1	01	0.1

Table 8.4: Contents of memory locations after multiplication by the slope

Congestion		Trust	
Fuzzy set	Corresponding Value	Fuzzy set	Corresponding Value
00	0.5	00	0.5
01	0.5	01	0.5

The flow charts of the algorithms for Fuzzifier, Inference Engine and Defuzzifier of the proposed FCP are shown in Figs. 8.8, 8.9 and 8.10 respectively.

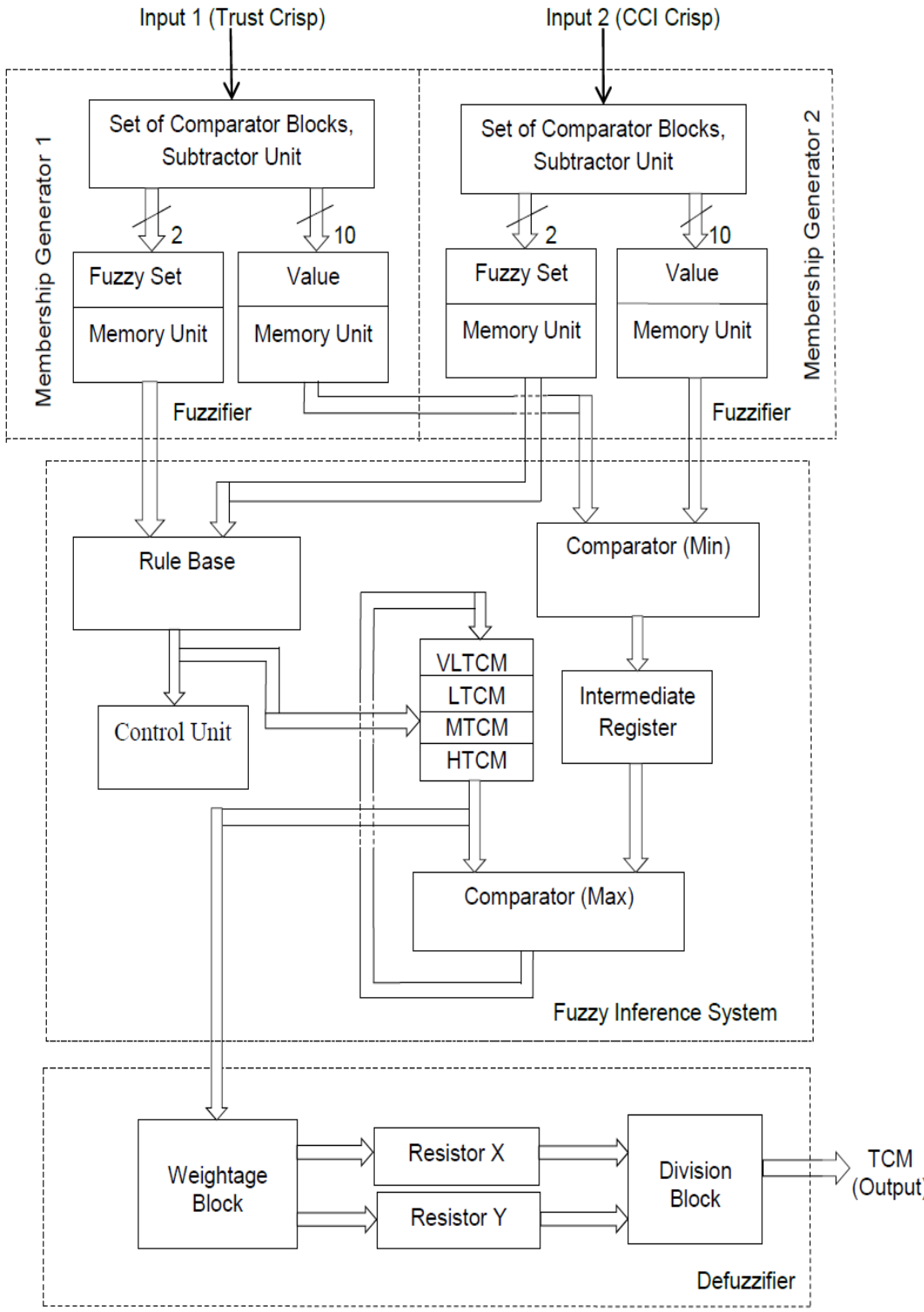


Figure 8.7: Architecture of the proposed Fuzzy Coprocessor

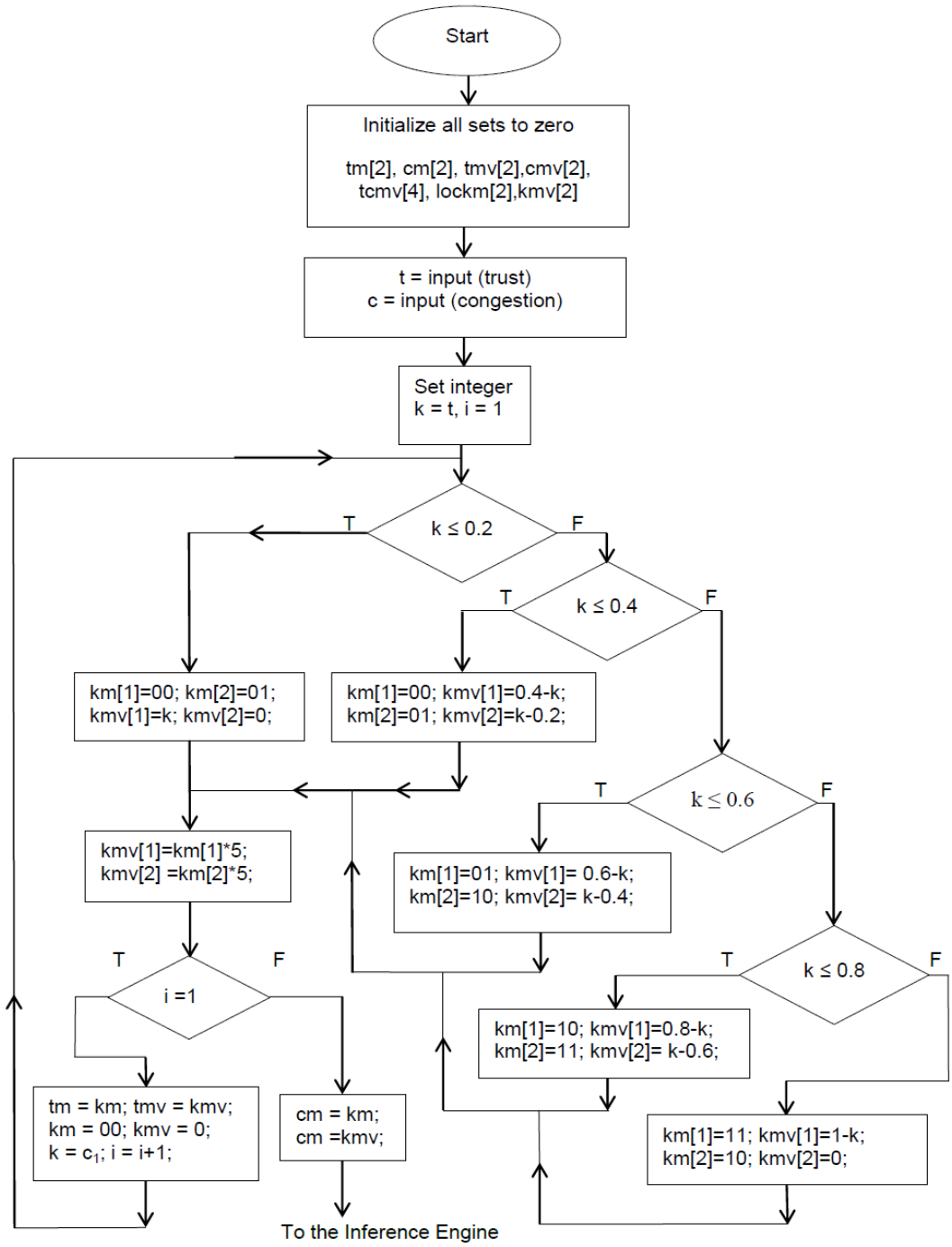


Figure 8.8: Flow chart of the Fuzzifier algorithm of the proposed FCP

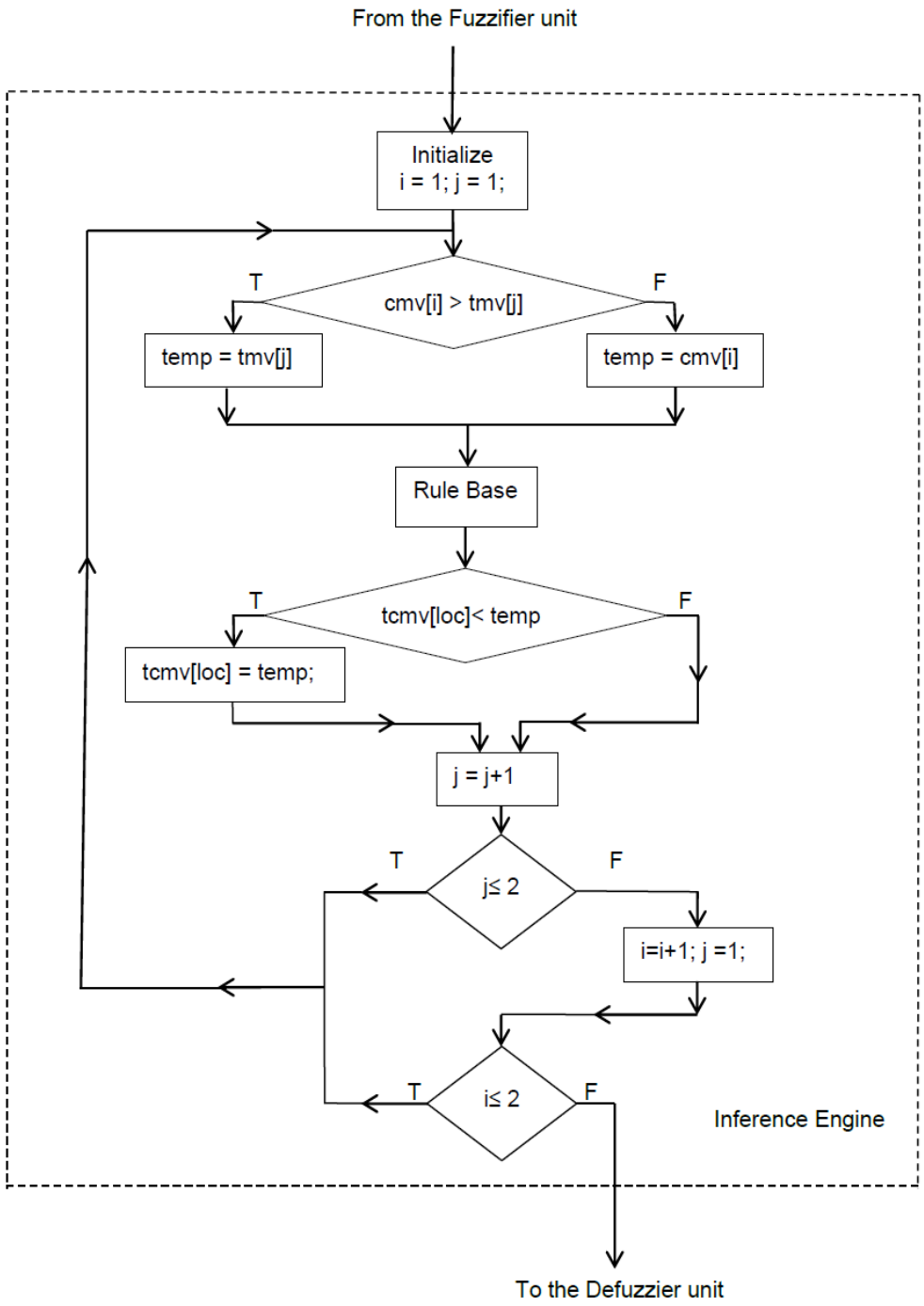


Figure 8.9: Flow chart of the inference Engine of the proposed FCP

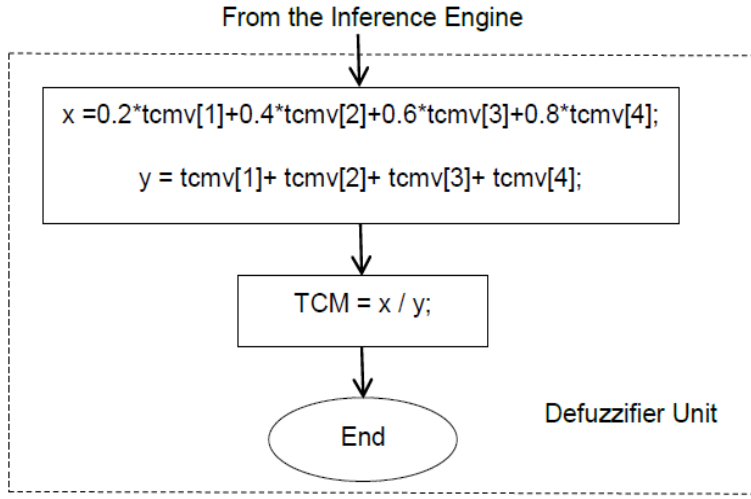


Figure 8.10: Flowchart of the Defuzzification algorithm of the FCP

.

8.5 Simulation Results

The simulation experiments of the proposed scheme have been conducted in two phases, viz. performance analysis of the dedicated Fuzzy Coprocessor and performance analysis of the proposed trust based congestion aware routing with Fuzzy Coprocessor (TCR-FC) algorithm.

8.5.1 Performance analysis of the dedicated Fuzzy Coprocessor

In the proposed model, it is assumed that the Fuzzy sets are symmetric in nature. Hence, VLSI architecture for the weighted average method of defuzzification is implemented for the sake of simplicity. The functional analysis of the proposed FCP model is performed in Verilog and the defuzzified value or the crisp value of TCM is obtained. The performance of the proposed architecture is verified by calculating the crisp value of TCM in MATLAB Fuzzy Toolbox where Centre of Gravity (COG) method of defuzzification is used. It has been observed that the TCM crisp values obtained from the proposed architecture by using weighted average method of defuzzification are almost same as that calculated in MATLAB Fuzzy Toolbox using COG method of defuzzification as shown in Table 8.5.

Table 8.5: Comparison of the proposed FCP with MATLAB simulation

Trust crisp value	CCI crisp value	Fuzzy Processor TCM crisp value	MATLAB Simulation TCM crisp value
0.1	0.1	0.2	0.2
	0.3	0.2	0.2
	0.5	0.3	0.3
	0.7	0.4	0.4
	0.9	0.4	0.4
0.3	0.1	0.2	0.2
	0.3	0.3	0.3
	0.5	0.3	0.3
	0.7	0.4	0.4
	0.9	0.4	0.4
0.5	0.1	0.3	0.3
	0.3	0.4	0.4
	0.5	0.5	0.5
	0.7	0.6	0.6
	0.9	0.6	0.6
0.7	0.1	0.5	0.5
	0.3	0.5	0.5
	0.5	0.7	0.7
	0.7	0.7	0.7
	0.9	0.8	0.8
0.9	0.1	0.6	0.6
	0.3	0.6	0.6
	0.5	0.7	0.7
	0.7	0.8	0.8
	0.9	0.8	0.8

The comparison of TCM crisp values for randomly selected trust and CCI from uniform distribution shows negligible error which is presented in Table 8.6.

Table 8.6: Comparison of TCM for randomly selected trust and CCI

Trust	0.933	0.392	0.171	0.031	0.046	0.823	0.317	0.034	0.381
CCI	0.849	0.743	0.655	0.706	0.277	0.097	0.695	0.950	0.438
FCP output	0.800	0.400	0.400	0.400	0.200	0.600	0.400	0.400	0.380
Simulation output	0.800	0.400	0.400	0.400	0.200	0.600	0.400	0.400	0.374
Error in %	0	0	0	0	0	0	0	0	1.6

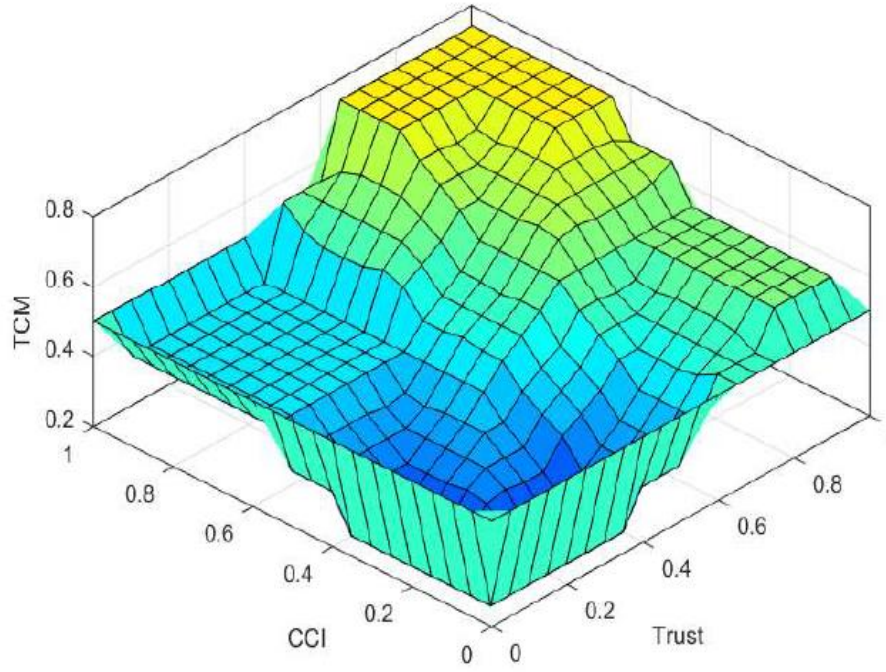


Figure 8.11: Surface view simulation graph from MATLAB Fuzzy Toolbox.

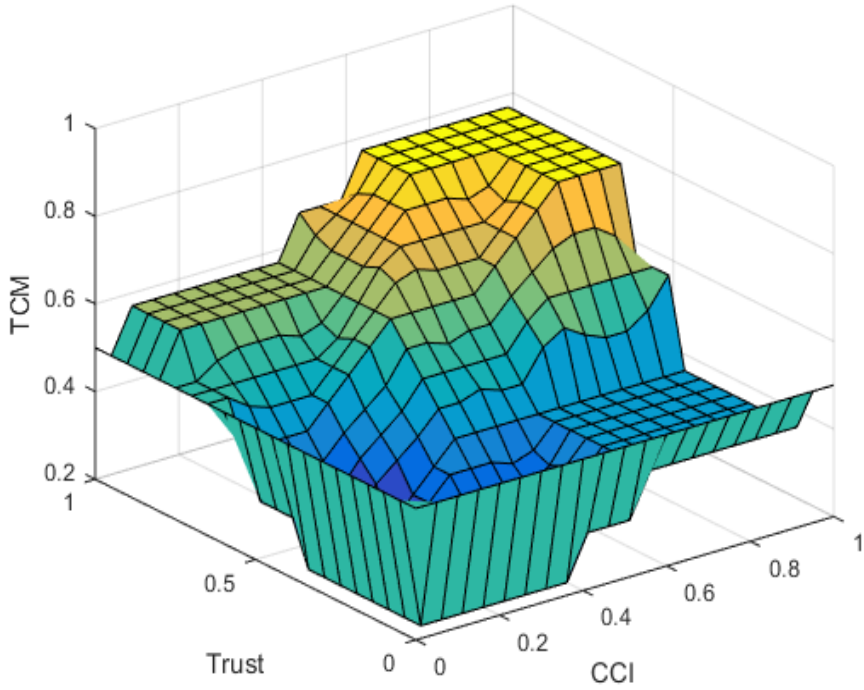


Figure 8.12: Surface view simulation graph from the proposed Fuzzy Coprocessor

Three dimensional surface view simulation graph generated in MATLAB Fuzzy Toolbox is almost same as that obtained from the proposed FCP architecture, as shown in Fig. 8.11 and 8.12, respectively.

The hardware implementation of the proposed Fuzzy Coprocessor (FCP) is performed in Spartan 3 XC3S50 -4PQ208 FPGA platform from Xilinx. It has around 1536 1536 Slices, 1536 4-input look up table (LUT) and 124 bonded input /output buffers (IOB). Table 8.7 represents the FPGA implementation results of the proposed Fuzzy Coprocessor.

Table 8.7: FPGA implementation results of the proposed FCP

Device Utilization Summary (FPGA: XC3S50-4PQ208)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Latches	189	1536	12%
Number of 4 input LUTs	691	1536	44%
Number of Occupied Slices	389	768	50%
Number of Slices Containing only related Logic	389	389	100%
Number of Slices containing unrelated logic	0	389	0%
Total number of 4 input LUTs	693	1536	45%
Number of Used as Logic	691		
Number used as route-thru	2		
Number of bonded IOBs	32	124	25%
Number of BUFGMUXs	1	8	12%
Number of used MULT18X18s	4	4	100%
Average Fan-out of non-clock Nets	3.07		
Timing Summary (Speed Grade: -4)			
Minimum period		17.551ns (Maximum Frequency: 56.977MHz)	
Minimum input arrival time before clock		1.825ns	
Maximum output required time after clock		7.165ns	

8.5.2 Performance analysis of the proposed algorithm

The proposed algorithm is simulated considering an arbitrary WSN with 100 immobile sensor nodes deployed uniformly over an area of 200 X 200 square meters. It is assumed that 10% of the total nodes are Fuzzy Coprocessors that are equipped with special hardware for Fuzzy calculations. Initially, all nodes are considered as trusted nodes having equal trust values. It is assumed that the date packets are generated randomly with Poisson distribution probability function. As time goes on, 0 to 25% of the nodes behave as malicious nodes during the simulation time frame of 1 hour. The Trust Threshold (TTH) value is considered as 0.5, which is application specific. High values of TTH imply high level of the network security. Trust and congestion status (CCI value) of a node is updated periodically after time interval of Δt equal to 5 seconds. The performance of the proposed algorithm is studied by observing the life time of the system, in terms of the percentage of dead nodes against the simulation time. The comparison graphs of the proposed TCR-FC algorithm with the existing Fuzzy algorithms TCEER [ACS15] and FCCTF [MZA10] are given in Fig. 8.13, considering initial node energy of 1 Joule per node.

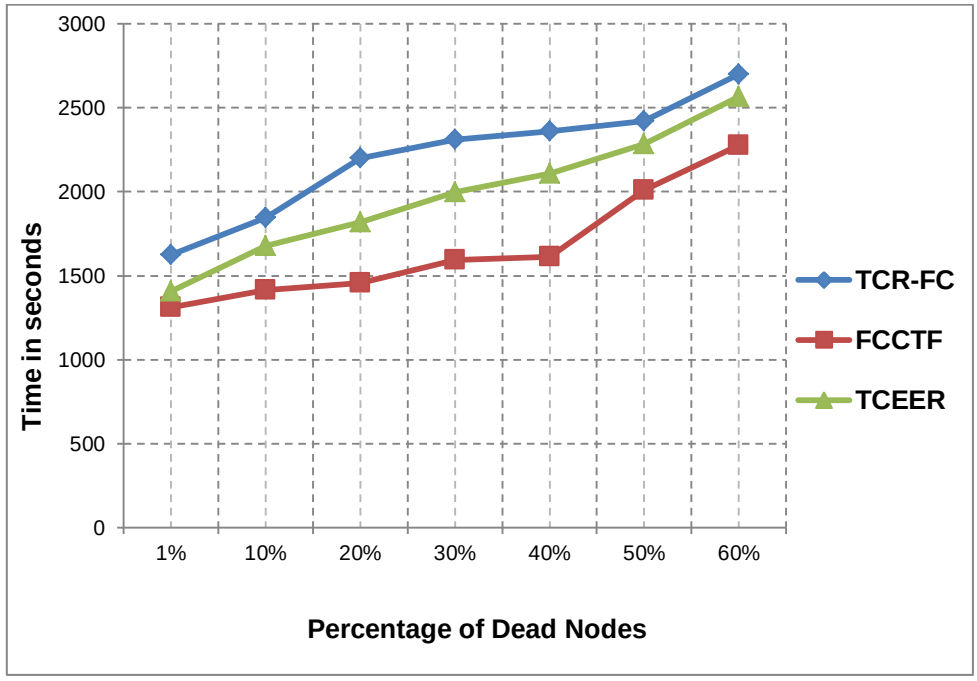


Figure 8.13: Comparison of TCR-FC algorithm with existing algorithms.

It is observed that the proposed scheme provides 21.7% to 36.4% improvement over FCCTF [MZA10] and 15.3% to 29.6% improvement over TCEER [ACS15] respectively. The results obtained are quite justified because introduction of the Fuzzy Coprocessors relief other sensor nodes from computation intensive fuzzy calculations and this makes the proposed scheme energy efficient and superior to its peers.

8.6 Summary

In this chapter, VLSI architecture of the dedicated Fuzzy Coprocessor for a new trust based congestion aware data routing algorithm (TCR-FC) for WSNs has been proposed. The hardware implementation of the Fuzzy Coprocessor is tested in Spartan 3 FPGA and the performance is examined through simulation in MATLAB Fuzzy Toolbox. The merits of the proposed TCR-FC algorithm with Fuzzy Coprocessor is verified by comparing with other existing fuzzy related similar algorithms which shows significant improvement over similar algorithms like TCEER [ACS15] and FCCTF [MZA10] respectively. In future, it is planned to verify the proposed scheme in comparatively large network with different conditions. It would be also like to work with optimal placement of the Fuzzy Coprocessor in the network to maximize the life time and throughput of the Wireless Sensor Networks.

CHAPTER 9

Conclusions

9.1 Contributions and Findings of the Thesis

In this thesis, the congestion control problems in Wireless Sensor Networks have been studied, focusing on the congestion obtained from the faulty behavior of the malicious nodes. The main scope of this study is to research the performance of the trust based congestion control, where the concept of trust is used as a mathematical tool to design the congestion control framework, by detecting and eliminating the malicious nodes from the data packet routing path. For this purpose, integration of trust and congestion control in the routing protocol has been considered, which is the new research domain in the nascent stage and has a great potential to develop further.

In *Chapter 2*, a survey of most commonly occurred security attacks and its effect upon the network congestion in the Wireless Sensor Networks is presented. This chapter also includes important trust evaluation methods and trust based routing protocols. The description and limitations of the existing congestion control algorithms are reviewed and discussed.

In *Chapter 3*, two new trust integrated data routing algorithms ITLSRP and FTSRP have been proposed where Link State Routing Protocol (LSRP) is used as the basic data routing scheme. In the ITLSRP algorithm, geometrical mean based indirect trust evaluation mechanism is considered for the calculation of trust of the individual sensor

nodes. On the other hand, a new fuzzy logic based trust evaluation model is proposed in FTSRP protocol.

In *Chapter 4,* a new congestion control protocol GACCTR is modeled with the help of the Genetic Algorithms, for balanced distribution of traffic among the different existing paths from the source node to the sink node, in accordance to the different route trust values. The merits of the proposed protocol in comparison to the existing routing protocols are justified through the simulation results.

In *Chapter 5,* two new Fuzzy algorithms (TFCC and TCEER) have been proposed for trust based congestion control in Wireless Multimedia Sensor Networks, where the faulty malicious nodes are identified and blocked from the data routing path by using the concept of trust. The merits of the proposed TFCC and TCEER algorithms are discussed by comparing with existing protocols.

Chapter 6 describes the TC-ACO algorithm where a new trust based congestion aware, energy efficient, data routing approach by utilizing Ant Colony Optimization techniques has been proposed. The merits of the proposed TC-ACO scheme are verified through simulations.

In *Chapter 7,* three of the previously proposed trust based congestion aware data routing algorithms (TFCC, TCEER and TC-ACO) are integrated into a single protocol suite (CET-PS), where the routing path is selected adaptively on the basis of the congestion status of the sensor node and the efficiency of the selected protocol.

In *Chapter 8,* a new trust based congestion control algorithm has been proposed, where dedicated state-of-the-art Fuzzy Co-processor (FCP) architecture is modeled. The Fuzzy calculations are done at FCP, causing reduction of computation overhead of the nodes. The merits of the proposed scheme are tested through simulations which show significant improvement in lifetime over the previous works.

In summary, the main contributions and findings of this thesis are listed as below.

- The outcome of the literature survey on the security attacks and the congestion control of the Wireless Sensor Networks represent that the traffic control and the resource control are not the only solutions for the improvement of congestion status of the network.

- The faulty behaviors of the malicious nodes aggravate congestion and make the situation worst.

- The trust based congestion control is the new research direction where the concept of trust is used as a tool to identify the malicious nodes creating congestion in the Wireless Sensor Networks.

- Presentation of the two new trust integrated data routing algorithms (ITLSRP and FTSRP), considering Link State Routing protocol (LSRP) as the basic data routing scheme. Instead of the shortest route, the proposed schemes select the most trusted route (best trusted route) dynamically. The methods of trust computation as described in ITLSRP and FTSRP respectively are light weight and show better performance compared to the existing trust evaluation methods.

- Proposal of a novel scheme (GACCTR algorithm) for distribution of traffic load among the existing routing paths by utilizing Genetic Algorithm. Rigorous simulation experiments are performed on GACCTR, which show improvement in the network lifetime and the percentage of successful packet transmission, compared to the existing similar algorithms.

- Proposal of two new trust based Fuzzy algorithms (TFCC and TCEER) for congestion control, where we get significant improvement in network lifetime over the existing similar protocols.

- Proposal of TC-ACO algorithm, where trust based congestion control is modeled using Ant Colony Optimization. The simulation experiments on TC-ACO shows better results compared to GACCTR and other similar protocols.

- Proposal of CET-PS algorithm, where the efficiency of the three protocols TFCC, TCEER and TC-ACO are compared. The simulation results of the proposed CET-PS scheme are presented to demonstrate its effectiveness compared to the standalone mode implementation of the individual protocols.

- Proposal of a dedicated state-of-the-art Fuzzy Coprocessor (FCP) architecture is given, which is implemented in Xilinx Spartan 3 Field Programmable Gate Array (FPGA).

9.2 Future Research Direction

As a future research work, the following would be considered:

- It would be interesting to study the feasibility of the proposed trust based congestion control algorithms in hardware, considering live environment with programming motes in TinyOS systems, written in nesC language.

- The proposed algorithms presented in this thesis are tested only on the networks having limited number of sensor nodes. The scalability issues of the proposed algorithms may be studied, considering heterogeneous multimedia sensor nodes.

- The proposed scheme of FPGA implementation for the Fuzzy Coprocessor architecture could be extended further to get its adaptability under various network conditions.

- In the proposed work, Fuzzy Coprocessor deployment is taken arbitrarily. The optimal deployment of the Fuzzy Coprocessors could be studied.

- In the present research work, it is considered that the distance of the sensor nodes from the base station or the sink is constant. The proposed algorithms could be modified for congestion control in case of the mobile sensor nodes.

9.3 Concluding Remarks

Finally, it is concluded that the outcome of the present research shows light on the congestion developed due to the faulty behavior of the malicious nodes. The integration of trust with congestion control in data routing protocol is the energy efficient way to restrict negative impact of the security attacks on the overall network congestion in the sensor networks. Here, the merits of the trust based congestion control over the conventional congestion control algorithms, in terms of network lifetimes and throughput have been claimed. The simulation experiments of the various proposed algorithms show significant improvement in network lifetime in case of trust based congestion control over the traditional approach. The overall performance of the trust based congestion control is more impressing compared to the conventional congestion control in Wireless Sensor Networks. However, this research works have mainly considered the simulation environment for implementation of the proposed schemes. The feasibility and utility of the proposed algorithms are yet to be tested in hardware and in actual practical field.

Bibliography

[AAP07] Asad Amir Pirzada and Chris McDonald, "Trusted Greedy Perimeter Stateless Routing", IEEE, ICON (2007).

[ABV02] A. Bharathidasan and V. A. S. Ponduru, "Sensor Networks: An Overview", Technical Report", Dept. of Computer Science, University of California, (2002).

[ACA12] A. Chakraborty, A. Raha, S. Maity, A. Karmakar, M. K. Naskar, " A Fuzzy based Trustworthy Route Selection Method using LSRP in Wireless Sensor Networks (FTRSP)", published in the proceedings of International Conference CCSEIT, ACM 978-1-4503-1310-0/12, (2012).

[ACS11] A. Chakraborty, S. K. Mitra and M. K. Naskar, "A Genetic Algorithm Inspired Routing Protocol for Wireless Sensor Networks", International Journal of Computer Intelligence- Theory and Practice, Vol. 1(6), (2011).

[ACS13] A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, "A Trust Based Fuzzy Algorithm for Congestion Control in Wireless Multimedia Sensor Networks (TFCC)", published in the proceedings of IEEE International Conference on Informatics, Electronics & Vision, (2013).

[ACS15] A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar,"Trust Integrated Congestion Aware Energy Efficient Routing for Wireless Multimedia Sensor Networks (TCEER)", International Journal of Computing and Information Technology, Vol. 23(2), (2015).

[ACK09] A. Chakraborty, K. Chakraborty, S. K. Mitra, M. K. Naskar, "An Optimized Lifetime Enhancement Scheme for Data Gathering in Wireless Sensor Networks", published in the proceedings of the IEEE International Conference on Wireless Communication and Sensor Networks (WCSN), (2009).

[ALG15] Ali Ghaffari, "Congestion Control Mechanisms in Wireless Sensor Networks: A Survey", Journal of Network and Computer Applications, Elsevier, Vol. 52, Pp.101-115,(2015).

[AMH14] A. Majidi, H. Mirvaziri, "A New Mechanism for Congestion Control in Wireless Multimedia Sensor Networks for Quality of Service and Network Life Time", American Journal of Computing Research Repository, Vol. 2 (1), Pp. 22-27, (2014).

[AMK05] Amit Konar, Computational Intelligence: Principles, Techniques and Applications. Springer, (2005).

[AML11] A. Martelli, L. A. Grieco, M. Bacco, G. Boggia, P. Camarda, “Selective Dropping Congestion Control for Wireless Multimedia Sensor Networks”, published in the proceedings of IEEE symposium on Computers and Communications, (2011).

[ANA07] Antonis Antoniou, Master’s Thesis on “Congestion Control in Wireless Sensor Networks”, University of Cyprus, department of Computer Science, May, (2007).

[APJ04] A. Perrig, J. Stankovic and D. Wagner, "Security in Wireless Sensor Networks", Communications of the ACM, Vol. 47, pp. 53-57, (2004).

[APR01] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar, “SPINS: Security Protocols for Sensor Networks”, 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001), (2001).

[ARM11] A. Raha, M. K. Naskar, S. S. Babu, O. Alfandi and D. Hogrefe “Trust integrated link state routing protocol for wireless sensor network (TILSRP)”, published in the proceedings of the IEEE International conference on Advanced Networks and Telecommunication Systems, (2011).

[ASA13] A. Chakraborty, S. Ganguly, A. Karmakar, M. K. Naskar, “A Trust Based Congestion-aware Hybrid Ant Colony Optimization Algorithm for Energy Efficient Routing in Wireless Sensor Networks (TC-ACO)”, published in the proceedings of the IEEE International Conference on Advanced Computing, (2013).

[AST89] Andrew S. Tanenbaum and David J. Wetherall, “Computer Networks”, Prentice Hall, (1989).

[BAF07] Behrouz A. Forouzan, “Data Communications and Networking”, 4th Edition, McGraw-Hill Higher Education, (2007).

[BKH00] Brad Karp, H.T.Kung, “GPRS: Greedy PerimeterStateless Routing for Wireless Networks”, MobiCom (2000).

[BHK04] Bret Hull, K. Jamieson, H. Balakrishnan, “Mitigating Congestion in Wireless Sensor Networks”, SenSys’ 04, Baltimore, Maryland, USA ACM, (2004).

[BRH12] Bavitha R and Hemalatha R, “Optimisation of Path using Genetic Algorithm for Wireless Sensor Networks”, published in the International Journal of Communications and Engineering, Vol. 05 (03), (2012).

[CHS12] Charalambos Sergiou, Ph. D Thesis on "Performance –Aware Congestion Control in Wireless Sensor Networks using Resource Control", Department of Computer Science, University of Cyprus, (2012).

[CKD03] C. Karlof, D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols, Vol. 1, Pp. 293-315, (2003).

[CSV13] C. Sergion, V. Vassilion, A. Paphitis, "Hierarchical Tree Alternative Path (HTAP) Algorithm for Congestion Control in Wireless Sensor Network", Ad Hoc Networks Vol.11, Pp. 257-272, (2013).

[CSZ14] Cagata Sonmez et al. "Fuzzy Based Congestion Control for Wireless Multimedia Sensor Networks", EURASIP Journal on Wireless Communications and Networking, Vol. 63, (2014).

[CWK 06] C. Wang, K. Sohraby, V. Lawrence, Bo li, Yueming Hu, "Priority Based Congestion Control in Wireless Sensor Networks", published in the proceedings of IEEE International Conference on Sensor Netyworks (SUTC'06), (2006).

[CYW03] C. Y. Wan, S. B. Eisenman and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", SenSys' 03, Pp. 266-279, ACM, (2003).

[CYW05] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks", IEEE Journal on Selected Areas in Communications, Vol. 23, No. 4, (2005).

[CYW09] C. Y. Wan, S. B. Eisenman, A. T. Campbell, J. Crowcroft, "Overload Traffic Management for Sensor Networks", ACM Transactions on Sensor Networks, Vol. 3, No. 4, Article 18, October, (2007).

[DCD04] D.Culler, D.Estrin, M.Srivastava, "Overview of Sensor Networks", IEEE Computer Society, August (2004).

[DGB89] D. Goldberg, B. Karp, Y. Ke, S. Nath, and S. Seshan, "Genetic algorithms in search, optimization, and machine learning", Addison-Wesley, (1989).

[FZL04] Feng Zhao & Leonidas Guibas, "Wireless Sensor Networks: An information Processing Approach", Elsevier, (2004).

[GNW10] G. Nagib and W. G. Ali, "Network Routing Protocol using Genetic Algorithms", published in the International Journal of Electrical & Computer Sciences, IJECS-IJENS, Vol. 10 (02), (2010).

[GZW02] Guoxing Zhan, Weisong Shi, and Julia Deng, "TARF: A Trust-aware Routing Framework For Wireless Sensor Networks",Springer -Verlag Berlin, (2010).

[HCA03] H. Chan, A. Perrig, "Security and Privacy in Sensor Networks", IEEE Computer Journal, Vol. 36, Pp. 103-105, (2003).

[HLJ09] Hong Luo, Jiaming Tao, Yan Sun, "Entropy-based Trust Management for Data Collection in Wireless Sensor Networks", published in Wireless Communications, Networking and Mobile Computing, WiCom'09, 5th IEEE International conference, (2009).

[HTS06] H. T. Shen, Lei Huang, Lei Li and Qiang Tan, "Behavior based trust in wireless sensor network", APWeb workshop, LNCS 3842, Pp 214-223, (2006).

[IFA02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: a survey", Computer Networks, Elsevier, Vol. 38, Pp 393-422, (2002).

[IFA07] I. F. Akyildiz, T. Melodia, K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks", Computer Networks, Vol. 51(4), Pp. 921-960, (2007).

[ISM00] IRIS senor mote data sheet http://www.memsic.com/wireless-sensor-networks

[ISM01] IRIS sensor mote data sheet - http://www.xbow.com

[JBI06] J. Bih, "Paradigm Shift- An Introduction to Fuzzy Logic", IEEE Potentials, Vol.25 (1), Pp. 6-21, (2006).

[JHJ09] J. Hong, J. Kook, S. Lee, D. Kwon, S. Yi, "T-LEACH: The Method of Threshold-based Cluster Head Replacement for Wireless Sensor Networks". Journal Information Systems Frontiers, Vol. 11 (5), Pp. 513-521, (2009).

[JKY07] J. Kang, Y. Zhang and B. Nath, "TARA: Topology Aware Resource Adaptation to Alleviate Congestion in Sensor Networks", IEEE Transactions on parallel and Distributed Systems, Vol.18 (7), Pp. 919 -931, (2007).

[JPS09] Jang-Ping Sheu, Li-Jen Chang and Wei-Kai Hu, "Hybrid Congestion Control Protocol in Wireless Sensor Networks (HCCP)", Journal of Information Science and Engineering, Vol. 25, Pp. 1103-1119, (2009).

[JSE09] Jaydip Sen, "A Survey on wireless sensor network security", International Journal of Communication Networks and Information Security , Vol. 1 (2), (2009).

[JYB08] J. Yick, B. Mukherjee, D. Ghosal, "Wireless Sensor Network Survey", Computer Networks, Elsevier, Vol.52, Pp. 2292 -2335, (2008).

[JYM10] Jing Yang, Mai Xu, Wei Zhao and Baogue Xu, "A Multipath Routing Protocol Based on Clustering and Ant Colony Optimization for Wireless Sensor Networks", Sensors, ISSN 1424-8220, Vol.10, Pp. 4521-4540, (2010).

[JZL10] Jing Zhao, L. Wang, S. Li, X. Liu, "A Survey of Congestion Control Mechanisms in Wireless Sensor Networks", published in the proceedings of 6th IEEE International conference on Intelligent Information Hiding and Multimedia Signal Processing, (2010).

[LLP07] Larry L. Peterson and Brace S. Davie, "Computer Networks: A Systems Approach", 4th Edition, Morgan Kaufmann publications, Elsevier, (2007).

[LXL03] Li Xiong, Ling Liu, "A Reputation-Based Trust Model for Peer-to-Peer e-Commerce Communities", Proceedings of the ACM Conference on Electronic Commerce, Pp. 228-229, (2003).

[MAK14] Mohamed Amine Kafi, Djamel Djenouri, Jalel Ben-Othman and Nadjib Badache, "Congestion Control Protocols in Wireless Sensor Networks: A Survey", IEEE Communications Surveys and Tutorials, Vol. 16, No.3, Third Quarter, (2014).

[MDG99] M. Dorigo, G.Di Caro, "Ant Colony Optimization: a new meta-heuristic", Published in Evolutionary Computation, (1999).

[MDM06] M. Dorigo, M. Birattari, and Thomas Stiitzle "Ant Colony Optimization-Artificial Ants as a Computational Intelligence Technique" IRIDIA Technical Report Series:TR/IRIDIA/2006-023, (2006).

[MDO92] M. Dorigo, "Optimization, Learning and Natural Algorithms", PhD thesis, DEI, Politecnico di Milano, Italy (1992).

[MHY08] Mohammad Hossein Yaghmaee and Donald Adjeroh, "A New Priority Based Congestion Control Protocol for Wireless Multimedia Sensor Networks", published in the proceedings of IEEE International conference 978-1-4244-2100-8/08, (2008).

[MKD09] M. K. Denko, T. Sun, I. Woungang, Joel J.P.C. Rodrigues and Han-Chieh Chao, "A Trust Management Scheme for Enhancing Security in Pervasive Wireless Networks", in the Proceedings of IEEE "GLOBECOM" (2009).

[MMM08] Muhammad Mostafa Monowar, Md. Obaidur Rahman, Al-Sakib Khan Pathan, Choong Seon Hong, "Congestion Control Protocol for Wireless Sensor Networks Handling Prioritized Heterogeneous Traffic", published in the proceedings of the 5th Annual International ACM Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Mobiquitous'08, Article No. 17, (2008).

[MOM08] Mohammad Momani, Ph.D thesis on "Bayesian methods for modeling and management of Trust in Wireless Sensor Networks", University of Technology, Sydney, July, (2008).

[MZA09] Mani Zarei, Amir Msoud Rahmani, Avesta Sasan, Mohammad Teshnehlab,"Fuzzy Based Trust Estimation for Congestion Control in Wireless Sensor Networks", International Conference on Intelligent Networking and Collaborative Systems, (2009).

[MZA10] Mani Zarei, Amir Msoud Rahmani, Razieh Farazkish, Sara Zahirnia,"FCCTF: Fairness Congestion Control for a Distrustful Wireless Sensor Network using Fuzzy logic", 10th International Conference on Hybrid Intelligent Systems, (2010).

[NPL04] nesC: A Programming Language for Deeply Networked Systems, UC Berkeley WEBS Project, December (2004).

[OBS12] Omar Banimelhem, Samer Khasawneh, "GMCAR: Grid –based multipath with congestion avoidance routing protocol in wireless sensor networks", Ad Hoc Networks, Vol. 10, Pp. 1346-1361, (2012).

[PLD06] Philip Levis, David Gay, "TinyOs Programming", www.cambridge.org, June 28, (2006).

[RCC10] R. Chakraborty, C. Gomathy, S. Sebastian, K. Pushparaj and V. B. Mon, "A Survey on Congestion Control in Wireless Sensor Networks", International Journal of Computer Science and communication, Vol. 1(1), Pp. 161-164, (2010).

[RRM09] Rahim Rashidi, M. A. J. Jamali, A. Salmasi, R. Tati, "Trust Routing Protocol based on Congestion Control in MANET", in the proceedings of the IEEE International Conference on Application of Information and Communication Technologies, (2009).

[SAM11] S. S. Babu, A. Raha, and M. K. Naskar, "A Direct Trust Dependent Link State Routing Protocol using Route Trusts for Wireless Sensor Networks (DTLSRP)",Wireless Sensor Network journal, USA, Pp. 125-134, (2011).

[SOD09] Selcuk Okdem and Dervis Karaboga, "Routing in Wireless Sensor Networks using an Ant Colony Optimization (ACO) Router Chip", Sensors, ISSN 1424-8220, Vol (9), Pp. 909 -921, (2009).

[SLC02] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient Gathering in Sensor Information System", published in the proceedings of IEEE International Conference on Computer Systems, San Francisco, USA, (2002).

[SMA13] Shahin Mahdizabeh Aghdam et al, "WCCP: A Congestion Control Protocol for Wireless Multimedia Communication in Sensor Networks". Ad Hoc Networks, Vol. 13, Pp. 516 – 534. (2013).

[SMT00] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad-hoc Networks", in Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom), ACM Press, Pp. 255 – 265, (2000).

[SSB11] S. S. Babu, A. Raha, M. K. Naskar, "Geometric Mean Based Trust Management System for Wireless Sensor Networks (GMTMS)", published in the proceedings of the IEEE International World Congress on Information and Communication Technologies, Pp. 444-449, (2011).

[SSB14] S. S. Babu, A. Raha, M. K. Naskar, "Trust Evaluation Based on Node's Characteristics and neighbouring Nodes' recommendations for WSN.Wireless Sensor Networks, Vol. 6, Pp.157-172, (2014).

[STP04] S. Tanachaiwiwat, P. Dave, R. Bhindwale, A. Helmy, "Location-centric isolation of misbehavior and trust routing in energy constrained sensor netwoks", published in the proceedings of the IEEE International Conference on Performance Computing and Communications, Pp. 463-469, (2004).

[TKD10] T. Kavita, D. Sridharan, "Securities Vulnerabilities in Wireless Sensor Networks: A Survy". Journal of Information Assurance and Security, Vol.5, Pp. 031-044, (2010).

[TKM08] Tae Kyung Kim, and Hee Suk Seo, "A Trust Model using Fuzzy Logic in Wireless Sensor Network", World Academy of Science, Engineering and Technology 42, Pp. 63-66, (2008).

[TZH09] Theodore Zahariadis, Helen C. Leligou, Panagiotis Trakadas and Stamatis Voliotis, Sotiris Maniatis, Panagiotis Trakadas, Panagiotis Karkazis , " An Energy and Trust Aware Routing Protocol for Large Wireless Sensor Networks" proceedings of the 9th WSEAS International Conference on Applied Informatics and Communications, (2009).

[TZH10] Theodone Zahariadis, Helen C. Leligou, Panagiotis Trakadas and Stamatis Voliotis, "Mobile Networks, Trust Management in Wireless Sensor Networks". European Transactions on Telecommunications, Vol 21, Pp. 386-395, (2010).

[UKC93] U. K. Chakraborty, D. G. Dastidar, "Using reliability analysis to estimate the number of generations to converge in genetic al;gorithm", Information Processing Letters, Vol. 46, Pp. 199-209, (1993).

[UKC97] U. K. Chakraborty, H. Muehlenbein, "Linkage Equilibrium and Genetic Algorithms", published in the proceedings of 4th IEEE International Conference on Evolutionary Computation, Pp. 25-29, (1997).

[VBR95] Valluru B. Rao and Hayagriva Rao, "C++ Neural Networks and Fuzzy logic: Introduction to Neural Networks", M & T Books, IDG Books Worldwide, Inc. ISBN: 1558515526, (1995).

[WRH00] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks", published in the proceedings of the 33 Hawaii International Conference on System Sciences, (2000).

[XLL04] Xiaoqi Li, Lyu, M.R., Jiangchuan Liu, "A Trust Model Based Routing Protocol for Secure Ad-hoc Networks", IEEE Proceedings on Aerospace Conference, Vol. 2, (2004).

[YCH00] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks", Proc. Eighth Annual International Conference on Mobile Computing and Networking (MobiCom), Pp. 12-23, (2000).

[YSO03] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks", published in the proceedings of the 4th ACM Symposium on Mobile Ad Hoc Networking & Computing", Pp. 177- 188, (2003).