Quality of Service Routing for Wireless Sensor Network

by

Muhammad Nur Rizal, M.Eng

Thesis

Submitted by Muhammad Nur Rizal

for fulfillment of the Requirements for the Degree of

Doctor of Philosophy (0190)

Supervisor: Dr. Iqbal Gondal

Associate Supervisor: Dr. Pari Delir Haghighi

Clayton School of Information Technology

Monash University

August, 2014

To my parents, my wife and my children:

You are always my energy and spirit of my life

Contents

List of Tables

List of Figures

Abstract

Over the last few years, the development of sensor networks has gained increasing importance due to their potential to support challenging research in a wide range of applications, and this has led to a new era of Multi-Objective Quality of Service Routing (MoQSR). This MoQSR is a routing algorithm that aims to satisfy multiple quality of service (QoS). MoQSR has the potential to support Wireless Sensor Network (WSN) applications that have different data which have quality and requirements. MoQSR enables the data communication protocols to provide a service differentiation mechanism that can capture the requirements of heterogeneous traffic. A key distinguishing feature of MoQSR is its ability to differentiate the required data related to each type of traffic and to provide a route selection that is based on differentiating QoS requirements according to the data type.

In fact, a WSN has limitations in energy supply and capabilities to ensure real-time guarantees because it poses a low duty cycle and transmission range. Most approaches related to data communication protocols are designed to deal with energy trade-off and not much has been done to optimise quality of service (QoS). Past research and current studies of QoS routing schemes only focus on a limited number of aspects of QoS. These studies do not provide a generic approach/scheme to encompass all attributes of QoS, and have limited support for meeting application specific needs. Thus, it is critical for QoS routing to factor in different application requirements. Therefore, in this dissertation, I propose, develop and validate a new QoS based routing for WSN, named Throughput Delay Guaranteed Routing Algorithm (TeGaR). Our TeGaR is based on differentiating QoS requirements according to the data type, which provides customised QoS metrics for each traffic category. It is modular and uses geographic information, which helps the routing algorithm create the best path to the destination node (BS). Our TeGaR aims to adapt to the requirements of heterogeneous traffic by using traffic diversity, the priority based delivery mechanism with multi-queuing policy and the route differentiation based Fitness Function method, which considers the quality of nodes matching the requirements of data categories. TeGaR is able to find a best node for a certain data category, while considering energy efficiency, reliability, latency and traffic congestion to cast QoS metrics as a multi-objective attribute.

However, there are some areas in which the performance of the proposed TeGaR can be enhanced. First, to address the 'energy hole problem' due to an imbalance of energy consumption in the network, according to the nature of many-to-one routing in WSNs, I propose a non-uniform transmission range strategy, namely Transmission Range Extension based on the neighbourhood Size (TReNs). TReNs allows the nodes to extend their transmission range to a certain level when the number of neighbours decreases below a certain level. This extended approach is able to distribute energy consumption evenly, thus avoiding an early network dysfunction. Second, to reduce the data flooding or unguided packet transmission, due to the characteristic of non-uniform node deployment and link instability in WSNs, I propose and develop a self-adaptation routing region approach, namely the Rule-based Learning Adaptation Approach for Routing Regions (RuLeARR) that aims to control the area (region) of routing for individual communication between nodes. In this way, packet detouring or data flooding can be reduced, thus decreasing routing overheads and the duration for data delivery, further saving energy. Third, to improve the benefits of adaptation of the routing region approach that copes with minor changes in the network QoS parameters, I integrate situation awareness into the routing region algorithm, to develop Situation Aware and Self-Adaptive Routing Regions (SASARR). The SASARR uses a fuzzy logic to represent approximate and imprecise context of the changes of the

network dynamic to represent a fine-grained and gradual adaptation for routing region. SASARRR extends the RuLeARR approach in order to provide a higher level of accuracy and granularity, thus improving QoS performance in a cost-efficient manner.

We validate all the approaches discussed above by conducting extensive experimental simulation. This evaluation clearly demonstrates the ability of the TeGaR scheme, with its extended approaches (TReNs, RuLeARR and SASARRR), to satisfy multiple QoS requirements according to different application requirements, while coping with the challenges due to the nature of data communication characteristics in WSNs. These approaches make a significant contribution to the overall efficiency and effectiveness of the QoS differentiating routing. The contribution of this thesis has resulted in one international peer reviewed paper and additional material for further publication.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Muhammad Nur Rizal
August 21, 2014

Acknowledgments

I would like to thank and pay tribute to everyone who make this possible. It has been an incredible journey of self-discovery, and I respect every one of you.

To my supervisors : I would like to convey my special thanks.

My sincere thanks to Dr. Iqbal Gondal: Thank you for your constant guidance, encouragement and endless support towards the completion of this thesis. I respect you because you have always believed that I can finish the thesis.

To Dr. Pari Delir Haghighi : I owe her so much. Her enthusiasm, thoroughness, desire to understand the content of the thesis, critical comments and also helping me to edit my thesis draft and her invaluable ideas to improve my thesis.

To Assoc. Prof. Bin Qiu : I convey deepest gratitude to him who believed in me to be my supervisor and mentor during my candidature time.

To my parents (Bapak Masduq Alalkaf, Ibu Nurjannah): Thank you for giving me endless support, the freedom and the opportunity to pursue my studies until this level, as always my mere expression of thanks likewise does not suffice. You always pray for me to be a good learner and father for my family. To my other parents (our late father Sunyoto, and Ibu Endang): you were not only my parents but also my friends who advise me about life, study and family.

To my wife (Novi Poespita Candra) : You are my truly source of inspiration. Your continuous love and support, patience, encouragement and standing by me through the good times and hard times is indisputable for my continuation of this study.

To my three angels (Aliya, Jaeza and Soraya): I still remember when you came here, you were still kids and funny. But, you have grown well now. Every day you gave me joy. Thank you for your support which you don't even know to your Daddy to complete this thesis.

To my lovely sister (Farah) and other brothers and sisters and all my close family that I cannot mention you one by one. You have given me unequivocal support throughout my studies. Thanks a lot for your moral supports.

I am especially grateful to Bayu Hastono for his assistance during my PhD study. Without his help, this research work would not be completed. To my colleague : Kiki Maulana, thank you for his assistance with a tutorial of LaText.

To Noriako Sato: My mentor of English, your assistance, dedication, support have improved my writing.

To Diane, Tim Flicker and Edward Stephenson : Thank you so much for proof reading my thesis. You have done an excellent job. Special thanks to Tim and Flicker, you have generously given your time and English expertise to complete the editing process.

To Prof. Bala Srinivasan and Prof Graham Farr, thank you for supporting me to continue my studentship till completion of this thesis.

To Monash University and the staffs of Clayton School of IT (Leanne, Kerry, Garry) : Thanks for providing me with an excellent research environment, and the necessary resources to undertake this research. Akamon Kunkongkapun, thank you for helping me in conference travelling.

To my friends in PIB (Perhimpunan Indonesia Belajar), I am extremely grateful to work with you to develop and run many activities that benefitted the quality of education in Indonesia. Guys, you have made me laugh, relax, optimistic and belief that I could finish our thesis while helping others.

Many of my fellow colleagues and friends that I could not include you in the list, thank you for making my life in Melbourne enjoyable and that would not be forgotten throughout my live.

Thank you to General Directorate of Higher Education, Ministry of Education and Culture Indonesia which provides me scholarship for studying PhD. And to my host institution (Gadjah Mada University) that supports me to pursue PhD at Monash University.

Finally, I would like to express my profound gratitude to the Almighty for giving me strengths and blessing.

Muhammad Nur Rizal

Monash University
August 2014

Chapter 1

Introduction

1.1 Background

The increasing growth of small, low-cost devices called sensors (Warneke and Pister, 2002) has provided potential collaboration amongst the sensors to develop wireless sensor networks (WSNs). These networks have significantly impacted our daily lives (Akyildiz et al., 2007) as they gather information from the physical environment and communicate and deliver this information to users using low-power wireless transmission (Schweitzer and Tilch, 2002). Networking sensors that are deployed in sensory fields may have a profound effect on the efficiency and performance of various applications from military to social issues (Vidhyapriya and Vanathi, 2007). These sensors can be used for security systems, environmental monitoring, location tracking systems and tactical surveillance in combat operations. For example, when I deploy many sensors in disaster fields, connecting them can help to assist rescue operations by tracking the risky paths, locating rescuers or victims as well as making the rescue team easier to identify in the current situation in the disaster area.

In ensuring communication between sensor nodes WSNs are vital as routing is responsible for selecting and maintaining the optimal path in the network under network constraints. By using this optimal path for data transfer, these sensors save energy, thereby extending the lifetime of the WSNs. Extensive research has been conducted in wireless networks to overcome the challenges in data gathering, processing and communication. However, WSNs pose constraints on energy, processing, memory and bandwidth (Al-Karaki and Kamal, 2004), and these constraints cause the problems of connectivity and longevity. Thus, the selection of routing protocols to deal with these constraints must find an efficient route.

The focus in this project is on routing protocol designs, which might differ from the Internet Protocol (IP) or ad hoc network due to the characteristics of WSNs. First, due to the large number of sensor nodes deployed, obtaining and exchanging data is sometimes more important than identifying nodes which send data (Al-Karaki and Kamal, 2004). Thus, the IP-based routing protocols may not be applied to WSNs (Schweitzer and Tilch, 2002). Second, WSNs have restricted energy supply while IP networks have abundant electricity supply thus WSNs need efficient resource management. Moreover, Johansson (2007) and Goodman (1997) show that 70% of the energy is dissipated for data routing in WSNs. Third, WSNs are application-centred, so the routing model is highly influenced by the application-specific needs whereas routing in IP networks is based on network topology. Fourth, data routing in sensor networks is coming from multiple sources to a particular Base Station (BS) that causes imbalanced energy consumption in the network while data transmission in IP networks uses multicast or peer to peer communication model. Due to such differences, designing techniques for energy efficient and reliable routing that takes into account the inherent features of WSNs is relatively considered.

Most routing approaches are designed to deal with energy trade-off and not much has been done to optimise Quality of Service (QoS) (Al-Karaki and Kamal, 2004). The QoS based routing protocols commonly aim to strike a balance between data quality and energy consumption to achieve route optimisation. The quality of the networks commonly includes certain QoS metrics such as energy, success delivery ratio, delay, energy, bandwidth, etc. However, most QoS based routing protocols focus more on energy conservation rather than the quality of data (Chang and Tassiulas, 2004; Kandris et al., 2011; Kim et al., 2011). These typical protocols sometimes reduce the quality of the routing performance because they need to lower energy consumption.

As WSNs are application dependent, the data communication model that drives information flowing into the network could be accomplished differently based on application and the critical nature and time constraints for data reporting (Djenouri and Balasingham, 2011). For instance, for a continuous monitoring application, the sensing data are sent regularly to the user at constant time intervals (periodical). While in applications such as rescue systems (i.e. for earthquakes), sensors react immediately when drastic changes occur and the transmission time is prioritised. Other applications may need a combination of data delivery models. Therefore, a service differentiation mechanism (Djenouri and Balasingham, 2011) is needed to capture the requirements of heterogeneous traffic.

However, the current studies on the QoS routing scheme focus exclusively on the limited number of aspects of QoS (Al-Karaki and Kamal, 2004). They do not provide a generic approach/scheme to encompass all the important attributes of QoS, and have limited support for factoring in different application requirements. To address this gap in QoS based routing protocols, this project proposes a novel approach for QoS routing in WSNs that can support WSN applications, having different types of data traffic using different QoS metrics. The proposed project should be developed in a modular design following its components such as traffic classification, multi-queuing system and route differentiation. The route differentiation is using a system called the fitness function. This modular protocol is able to perform route selection that ensures the required QoS metrics for each type of traffic category in a cost-efficient manner.

Although the proposed novel approach with the differentiating QoS requirements according to the data type potentially can provide a multi-objective QoS, there still are many issues that need to be addressed to deal with inherent WSNs characteristics (Al-Karaki and Kamal, 2004). This study limits the issues on the 'energy hole problem', 'packet detouring' and 'loss of accuracy and efficiency'.

The project is driven by the need for a new localised QoS that caters to different application requirements for supporting heterogeneous traffic in WSNs, while contending with the challenges caused by the multihop communication.

1.2 Research Aims and Objectives

This project aims to design, develop and evaluate a QoS differentiating routing scheme and its enhancement approaches for the purpose of i) supporting heterogeneous traffic in WSNs by differentiating QoS requirements according to the data type and ii) addressing the problems of 'energy hole', 'packet detouring' and 'loss of accuracy and efficiency' due to the communication model in WSNs. The proposed scheme provides customised QoS requirements for each data category and takes into account some routing challenges in WSNs, with the development of several extended routing approaches. The challenges of routing designs include node deployment, the data reporting model, heterogeneous traffic, imbalanced energy consumption, routing overheads and uncertain changes in network attribute values for QoS performance.

In response to the above aims, the following objectives were suggested.

- To satisfy multiple QoS requirements according to various WSN applications to meet specific needs by differentiating data related to each type of traffic based on differentiating QoS requirements. This aims to deal with the different application requirements by tailoring adaptation of route differentiation to suit the application of specific needs.

- To achieve a routing scheme with energy efficiency, a new routing strategy should avoid routing approaches which consume too much energy, while not compromising data quality. The research incorporates the most important attributes of QoS which include availability of energy, reliability, latency and other communication parameters for addressing the constraints in WSNs.

- To improve the adaptation capabilities of the proposed framework, some extended routing approaches are required to address the issues and challenges arising from performing route adaptation according to types of data traffic due to changes in WSNs.

- Several analyses will be carried out, in order to prove the functionality and applicability of the proposed approaches in relation to problems of heterogenous traffic, congestion and topological changes while considering power efficiency.

To address these objectives, the following steps will be taken.

- Conducting a comparative study of routing techniques in the WSNs in relation to QoS routing approach. Throughout the research, relevant and up-to-date literature needs to be continuously analysed to keep up with the latest research. The study reviews various routing protocols as well as existing QoS routing protocols, and highlights problems and gaps in relation to QoS routing.

- Developing the QoS routing framework by designing and developing routing approaches. The design takes the original ideas and conceptualises a new QoS routing scheme that will address the research project aims. Some challenges are faced in designing WSNs into a single QoS routing framework that includes several routing approaches. The framework should reflect routing problems and gaps according to the research aims. The development of software as a simulation tool will implement the proposed routing approaches in various simulation environments and scenarios. The process of software development is based on the framework proposed in the project.

- Implementing the novel QoS routing framework in the simulation tool in order to examine and demonstrate the compatibility of the novel QoS framework under different congestion scenarios as well as types of traffic and node deployment. The framework consists of several adaptation approaches to deal with the abovementioned issues including 'multiple QoS-objectives', the 'energy hole problem', route discovery overheads due to network dynamic' and 'uncertain or minor changes in the sensor node status.

Finally, after the development and the implementation of the QoS routing framework, the project runs simulation tests to evaluate the performance of the scheme. The objectives of the evaluation are:

- to evaluate the performance of the proposed scheme with existing routing protocol in WSNs using performance metrics such as energy consumption, latency, throughput and reliability of networks
- to compare routing strategies in the proposed QoS framework in order to evaluate the benefits between the adaptation/extended approaches in coping with the issues as mentioned above.

The proposed QoS differentiating routing protocol is developed and analysed using a network simulator that I developed for this research because the existing network simulators (e.g. NS2, NS3, Opnet, Omnet etc) are not flexible enough to accommodate the needs of this research and its experiments. The contributions of this tailor-made simulator include: (1) flexibility: to provide and support multiple modules that fit the simulation functionalities for this research (2) extensibility: providing traffic classification module to make simulation engine extensible (3) reusability: to enable using this simulator and its modules in other studies.

1.3 Research significance

The key contribution of the thesis is to introduce a novel QoS routing protocol that can cater for the requirements of different data types while addressing the problems of uneven energy depletion, packet detouring/broadcast storm and energy inefficiency which are very common routing issues due to the nature of multi-hop and many-to-one data communication in dynamic WSNs environments. The proposed algorithms and techniques are presented in Chapter 3, 4, 5 and 6 accordingly.

Compared to conventional networks, QoS based routing studies in WSNs consider availability of limited resources (e.g. energy), bandwidth and memory (buffer size). Moreover, WSNs is developed based on a specific application scenario. For instance, some applications (object tracking) are subjected to time-constrained traffics, and they require bounded latency during data transmission. Other applications (surveillance monitoring) use video or image sensory which require dedicated bandwidths, network reliability and real time traffic services. Further, applications in sensor networks require prolonged network life. Therefore, QoS support in the WSNs is very crucial to ensure that data delivery is accomplished as per the requirements of the applications.

A key factor in developing and performing QoS routing techniques for different application requirements in dynamic and resource constrained WSNs environments is adaptation (Heinzelman et al., 1999). Adaptation in routing protocols enables mechanisms for reconfiguration due to node failures for maintaining network functioning and performing its tasks under unstable networks. The adaptation also relates to standardised quality for data communication with regard to certain application needs.

In addition, to find a path that fulfils the required data related QoS metric for multi-objectives QoS according to specific application need, the first QoS routing approach selects quality nodes that fit the type of traffic using a Fitness Function method as introduced by Goldberg (1989). This function is able to compute path selection based on multiple QoS attributes to enable route adaptation for each type of traffic. Thus , this approach supports heterogeneous traffics with regard to different types of data classes.

To enhance the benefits of the first approach for satisfying multi-objectives QoS, I develop extended routing approaches that incorporate context-awareness into the adaptation process in order to be aware of availability of resources and current condition of the network. This context-awareness enables the extended approaches to cope with the design routing challenges due to the intrinsic characteristics of WSNs. The challenges include i) energy hole problem due to uneven energy depletion (Madkour et al., 2013) ii) route discovery overheads due to network dynamic and iii) uncertain or minor changes in the sensor node status. These extended adaptations aim to maximise the benefits of the novel QoS routing approach in terms of improving energy-efficiency and network performances while dealing with the characteristics of WSNs as described previously.

All the approaches described above perform a holistic QoS routing framework to leverage the full potential of adaptation for routing algorithms in WSNs and maximise the benefits of adaptation in terms of the improvement of either energy-efficiency or network performances such as throughput, reliability and timeliness as well as the reduction of communication overheads.

The proposed routing approaches made significant contributions to QoS routing protocol by providing better results compared to the existing QoS routing in WSNs, and further the three adaptation strategies demonstrated the improvement in terms of energy consumption, overheads, reliability and latency.

After discussing all the sections, I conclude the chapter by outlining the structure of the thesis as follows.

1.4 Structure of the thesis

This chapter provided the background and research aims and objectives, followed by the research significance.

Chapter 2 presents the fundamentals and the constraints of WSNs. According to the research, I review and examine relevant literature on energy efficient routing protocol in WSNs, highlighting the limitations of existing routing protocols. Based on the gaps, combined with routing challenges and design factors in WSNs, I develop a holistic QoS routing scheme/framework for factoring in different application needs. We also present a review of relevant literature to address varying network dynamics such as imbalanced load distribution, communication overheads due to undirected packet relay and uncertain changes in network attributes.

Chapter 3 describes Throughput-delay Guaranteed Routing for Reliability Algorithm (TeGaR) that provides traffic differentiation for different data requirements. We describe a fitness function based Genetic Algorithm, and explain how this fitness is used for differentiating routing based on the types of data, and how it is built and maintained. We also introduce the principal techniques of TeGaR such as Traffic Classification, Route Selection and Priority based Delivery with Queuing Manager. The experimental simulations will compare the performance of TeGaR with SPEED (He et al., 2003), a prominent QoS routing algorithm that has been widely used in WSNs.

Chapter 4 discusses the extension of the TeGaR approach that aims to address the 'energy hole problem' (Madkour et al., 2013) by proposing Transmission Range Extension based on the neighbourhood Size (TReNs), to provide balanced energy consumption. The benefit of TReNs is to reduce uneven energy consumption in the network (Li and Mohapatra, 2007), to further improve the network lifetime, and avoid early network dysfunction. In this chapter, I run simulation to evaluate the performances between TReNs and TeGaR and SPEED (He et al., 2003). Partial results of this chapter have been published (Rizal et al., 2012)

Chapter 5 discusses the design of a new technique that aims to address the 'broadcast flooding that causes communication overheads'due to improper packet routing by controlling the area (region) of routing for individual communication sessions. We develop a Rule-based Learning Adaptation Approach for Routing Regions (RuLeARR) that uses the location information of nodes to guide packet routing to improve route optimisation and restrict undirected data routing that consumes energy. The performance of the network is dependent on the size of the boundary of the region which is adapted using a method proposed in this project. A comparative evaluation will be conducted between this scheme, TeGaR and SPEED (He et al., 2003).

Chapter 6 proposes and develops a situation-aware adaptation for routing region (SASARRR) approach using Fuzzy Situation Inference for Routing Regions (FSIRR) that extends the FSI (Delir Haghighi et al., 2008; Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009). SASARRR enhances the RuLeARR approach to provide a higher level of accuracy and granularity to drive the adaptation of routing region in a situation-aware manner. This study performs an evaluative comparison between SASARRR and RuLeARR for the improvement of QoS performances and energy efficiency. And the results of SASARRR will be published later.

Finally, Chapter 7 summarises the research and describes contributions and possible directions for future research that builds upon the work in this dissertation.

Chapter 2

QoS Routing in Wireless Sensor Networks

2.1 Introduction

As discussed in Chapter 1, the need to design routing schemes/protocols for application-specific needs increases along with the growing development of multiple sensing devices. This challenges a new area of research, that is, to adapt to specific behaviours of application services in wireless sensor networks (WSNs) while contending with network constraints (i.e. limited energy, limited device capabilities, bandwidth, etc.) from a network layer point of view.

The aims of this chapter are (1) to study the hardware platform of sensor nodes to meet the needs and the architecture of WSNs, and allow the development of application specific protocols related to routing; and (2) to review protocols with respect to routing in WSNs. We evaluate the existing routing protocols for QoS to find a new solution related to routing techniques that address the main aim of the thesis.

By examining the current routing protocols in WSNs, I identify two important issues of routing in WSNs environments. These issues are as follows.

1. Energy efficiency. Since sensor nodes are constrained in energy supply and hardware capabilities (Ares et al., 2007), each protocol being developed should manage the resource carefully otherwise the lifetime of the network ends prematurely. This energy constraint causes the problem of connectivity and longevity that further affect the designing of protocols and energy awareness at all layers of the network protocol stack. In the literature on routing, many routing protocols focus on energy balancing techniques to eliminate energy inefficiency to extend the network lifetime (Daabaj et al., 2009; Khamforoosh and Khamforoush, 2009; Kordafshari et al., 2009a).

2. Data quality. The need to balance energy consumption and data quality is required before the sensing networks deliver the data to their destination (i.e. BS). This quality includes certain pre-determined QoS metrics such as reliability, throughput, energy, delay, etc. In order to achieve efficient data communication, designing such QoS based routing protocols need to consider the inherent characteristics of WSNs that are highly unpredictable, unstable and, more importantly, energy constrained environments (Ota et al., 2003). Therefore such adaptation techniques are required in order to better cope with the changes that might frequently arise in these scarce networks. Moreover, many applications in WSNs require different data quality and requirements (Djenouri and Balasingham, 2011). Some applications are subjected to time-constrained traffic, others require dedicated bandwidth or hybrids and the rest focus on prolonging network longevity. Thus, there is a need to go beyond

the capabilities of current routing techniques by promoting a new routing method that can adapt to the specific needs while being aware of energy efficiency based on the knowledge of nodal features/quality (i.e. energy, reliability, latency and other communication parameters) (Dutta et al., 2005; Gu et al., 2005).

According to the routing issues mentioned above, I review state-of-the art routing protocols. The analysis highlights the gaps and problems for routing protocol designs and in particular for existing QoS approaches with regard to application-specific needs as well as preserving energy conservation. The review results are used to provide the background necessary for the development of potential projects discussed in later chapters. Integrating various adaptation approaches with context-awareness into QoS routing algorithms is one of the contributions of this thesis, which aims to improve route optimisation and performances as well as cost-efficiency in WSNs.

This chapter comprises three parts: (i) the architecture of wireless sensor nodes and WSNs requirements; (ii) the comparison between different routing protocols based on system evaluation metrics; and (iii) the underlying concept of the proposed QoS routing scheme and potential projects for further research.

The first part describes the introductory materials of wireless sensor nodes from which to provide the basis for knowledge for the rest of the chapter. This part includes wireless sensor node architecture which presents both the basic operation and hardware capabilities of sensors. A brief description of the class and application of WSNs is presented in this chapter. Finally, in conjunction with application needs, the WSNs requirements and challenges are detailed to give a comprehensive knowledge for the researcher to design a potential project for routing protocols in WSNs.

The second part of this chapter provides state-of-the-art routing protocols for WSNs based on the network structure and protocol operations. It reviews routing techniques and their challenges according to the characteristic of WSNs. We identify each routing protocol with regard to both advantages and drawbacks in WSNs environments, and compare the effectiveness of each protocol using evaluation metrics. The summary results are used to identify gaps or problems that can be used as a baseline to propose new routing schemes in relation to QoS routing protocols.

The final part of this chapter presents the underlying theory of the proposed project with its extended approaches based on QoS routing operations. Performing QoS routing in resource constrained environments introduces further challenges that should be addressed to facilitate WSNs requirements and to deal with network dynamics in WSNs such as heterogeneous traffic (Alanazi and Ouni, 2013; Dunkels et al., 2003), energy hole problems due to unbalanced data communication (Wu et al., 2006), communication overheads due to undirected packet routing (Basagni et al., 1998) and energy inefficiency due to inability in representing uncertain changes in the network. Finally, this chapter recapitulates the discussions.

2.2 The concept of sensor nodes

Wireless Sensor Networks (WSNs) consist of small sensing devices called sensor nodes with sensing, computation and wireless communication capabilities. The first development stage of sensor nodes was established in 1998 with the Smartdust project (V. Hsu, February, 1998). Smartdust is a system of many tiny sensors that create autonomous detection to ambient surroundings, such as light, temperature, vibration, magnetism, etc. and communicate wirelessly within a cubic millimeter of range. Many research projects on sensor nodes have since been developed based on the development of the Smartdust project including Berkeley NEST (Doherty et al., 2001) and CENS (Zhao et al., 2002). Extensive research has since been carried out to increase the capability of physical sensor nodes by integrating more silicon to produce a sensing device with longer wireless range, lower energy consumption and compact architecture for ease of use by users (He et al., 2004).

While he capabilities of sensor nodes increase, energy, computation and communication are limited. However, combining and networking these devices offers many potential applications. To make the vision of WSNs become a reality, the following subsection describes the architecture of WSNs that provide both knowledge and the function of each component required to meet the needs of wireless sensor networks.

2.2.1 Wireless Sensor Node Architecture

This section presents a general architecture that details the basic operations of computation and communication of sensors and addresses both the needs and design challenges of WSNs influenced by limitations on the sensor's technology.

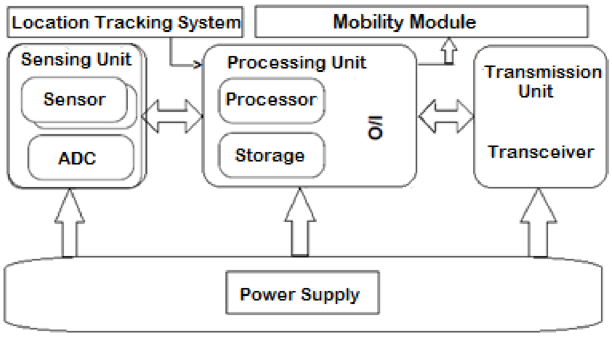

Figure 2.1: Wireless Sensor Node Architecture
redrawn from (Al-Karaki and Kamal, 2004)

Figure 2.1 shows the architecture of a sensor node, which basically combines sensing, processing (computation), transmission (communication), mobility and location tracking systems (optional) and a power unit in a single, tiny device. In addition, some applications in WSNs may include certain specific components such as location tracking systems or mobility modules. For this reason, many commercial sensor node devices provide expansion slots. The following describes the architecture or basic components of sensors as depicted in Figure 2.1.

- Power Supply. To provide a self-sustaining sensor to perform sensing, processing and communicating, a power unit is required for energy reserves (typically formed as a battery) or power scavenging methods (e.g. solar cells). As availability of power is the most critical resource in WSNs, this component supplies the sensors with energy and when the battery is exhausted the sensors are not able to perform their tasks. However, to maintain operation and avoid larger subnetwork disconnectedness, changing the battery is impractical, moreover for environmental monitoring applications where sensors are dropped from helicopters scattered in a risky field. Such conservation techniques are further studied and designed to extend the lifetime of the battery (Heinzelman et al., 2000). The amount of energy usage is equally dependent on the allocated time for sensor reading or sensing or transmitting.

- Transceiver. This component is responsible for the data communication between sensor nodes using a radio frequency (RF) transceiver or the 802.15.4 compliant protocols Zigbee (Kohvakka et al., 2006) that specifies the physical layer and medium access control for low and short transmission range communications.

- Microcontroller. A Microcontroller is an electronic circuit for interfacing between the sensors and an energy source. It provides power for processing data and coordinates the activity of a sensor node in WSNs with limited memory and I/O peripherals. This component connects to the power in order for the sensor to produce a number of samples per second based on the application needs. However, the existence of additional slots in a sensor node can help a sensor to include some external memories to improve the capabilities of the sensing devices.

- Sensing Unit. This component comprises a number of sensors and a tool for digitalising analogue signals called an analogue digital converter (ADC). This sensing circuitry measures the ambient conditions of 'the physical world' such as thermal, acoustic, light, sound, image, temperatures, etc. and transforms them into a continual waveform that must be digitised and analysed by the microcontroller. According to the capabilities and characteristics of the sensing device, this digitalisation can be a complex process, and hundreds of applications could be developed from this modern networking technology.

The descriptions of the above architecture have emphasised the need for combining sensors, radios, CPU processing and power supply modules into an effective wireless sensor network, which requires a detailed understanding of the hardware components of sensor nodes.

2.2.2 Sensor's hardware capabilities

This section provides an overview of the characteristics or capabilities of the modern hardware of sensors in order to make them operational in WSNs while evaluating the energy consumption of nodes as the most important issue in these networks.

1. **Power Supply**. Energy is essential in WSNs as energy supplies power for sensor nodes to perform for both computations and communications. This energy is supplied by the battery. The types of battery in WSNs, such as Alkaline, Lithium, and Nickel Metal Hydride (Hill, 2003), will provide different energy capacity, physical sizes, costs and operations that affect system performance (Heinzelman et al., 2000). However, power supply in sensor nodes is limited. Due to the limitation on energy storage capability of sensor nodes, energy consumption should be managed wisely. As energy consumption is one of the most challenging aspects in WSNs, protocols in these networks should deal with this critical issue.

2. **Radio or Transceiver**. Radio is a crucial component in WSNs because it performs data communication between nodes. The budget for communication consumption requires the highest energy in WSNs (Goodman, 1997). Radio in sensors is symmetric and both data transmission and reception use the same amount of energy. In addition, power output affects the range of transmission where the relationship between them is a polynomial, with an exponential of 3 and 4 in non-line of sight propagation (Barry, n.d.). For example, transmitting data with double range through an indoor environment will need 8 to 16 times more energy. In WSNs, each sensor commonly uses an omnidirectional antenna to allow the node to communicate effectively in all directions, and various modulation types increase channel reliability and tolerance to noise by spreading the signal to be communicated over a wide range of frequencies. Due to limited energy, the bandwidth capacity in sensor motes, such as MICA2 and MICAz, produce low data rates of about 40 kbit/s and 250 kbit/s (Akyildiz et al., 2007)

3. **Processor or Microcontroller**. The processor is a part of the sensor node that is built into the microcontroller with tasks to read and process the sensed data ready for transfer. The processor is responsible for doing computations from data queue scheduling, calculating energy to other computational operations with relatively low speed CPU processing. However, when the sensors need a CPU to boost their computation speed it results in higher energy consumption. In these scarce networks, power consumption in the microcontroller varies and should be managed carefully to achieve power efficiency. When a sensor node is in idle mode, the CPU will stop operations and only maintain its memory and time synchronisation for wake-up when necessary. In WSNs, the transition time for entering and exiting sleep modes differs amongst sensors and significantly affects the energy budget. The more responsive the node performs transition will reduce the energy consumption. Hence, the energy costs for this module is the sum of state energy consumption plus state transition energy consumption.

4. **Memory**. Sensors commonly have a storage capacity between 1 and 128 KB of on-chip program storage for either temporary data storage or program memory (Hill, 2003). In WSNs, modern flash technology is more persistently used in WSNs as this flash produces longer life cycles while minimising energy usage for operations (Chung and Song, 2009).

5. **Sensor**. A sensor is one of the main parts of wireless sensor nodes that aim to sense ambient conditions from text or scalar data to complex digital senses. The energy consumption of a sensor is measured due to fewer operations such as sensor cycles (periodic sleep/wake/idle), signal conversion and modulation, signal sampling and signal sensing. According to taking sampling and sensing, this energy is equal to the working voltage as described in a power supply module multiplied by the current cycle of sensors and the time it takes to operate them. How quick sensors sense and take samples is an important factor in measuring energy costs. Producing higher samples of reading consumes much more energy than lower rates. Table 2.1 presents the review of each sensor platform (Healy et al., 2008).

Mote	Processor	Flash	RAM	ADC	Serial-comm	Current	
						Active	Sleep
Rene	AT90LS8535	8K	0.5K	10 bit	UART	6.4 mA	$< 5\mu$A
MicaZ	ATMega128L	128K	4K	10 bit	UART	8 mA	$< 15\mu$A
IRIS	ATMega1281	128K	8K	10 bit	UART	8 mA	8 μA
SHIMMER	MSP430	48K	10K	12 bit	UART	1.8 mA	5.1 μA
TelosB	MSP430	48K	10K	12 bit	UART	1.8 mA	5.1 μA
SunSPOT	AT91RM920T	4M	512K	12 bit	USART	25 mA	0.5 mA
SunSPOT	NXPLPC1758	512K	64K	12 bit	USB	50 mA	1 A

Table 2.1: The capacity of some sensor nodes

According to Table 2.1, these sensors have tiny memories and they are power constrained. In the future, the size of sensors will be reduced to suit application scenarios and reduce the cost of production. As sensors pose limitations on computation, storage and transmission range, this leads to a new set of architectural issues that address the needs of WSNs. Thus to meet the requirements of WSNs, the following section discusses system designs and challenges that are influenced by the energy constraints embedded in the individual sensor nodes.

2.3 Design Factors and Challenges

Wireless sensor networks (WSNs) basically differ from the internet or ad hoc networks according to their characteristics. According to limited power transmission, the communication in WSNs is done with local peers in a multi-hop fashion, not directly to the BS. Unlike any traditional wireless network, WSNs do not depend on pre-deployed infrastructure because sensor nodes can be added to the networks to expand their sensing area with low-cost budgets. As long as there is sufficient density, many single nodes can be placed or dropped to cover a larger geographic area. These deployed sensors are assembled and organise themselves to fulfil their tasks for supporting many potential applications. In addition, these networks are able to adapt to the changes caused by node failures. This is completely different compared to traditional wireless networks as WSNs have the ability to dynamically adapt to changing environments.

The potential large scale deployment and the demands for WSNs to cope with topological changes combined with resource constraints pose challenges for designers to design energy efficient communication protocols. A well-designed wireless sensor network requires many fundamental challenges to be addressed. To build good WSNs for different applications, the development of a better sensor with regard to modern hardware technology is not enough; various factors make a sensor network good. All of the above factors should be managed well in advance for the network to avoid suffering overhead costs while maintaining power efficiency. The following summarises several important factors that affect the performance and efficiency of the WSNs with respect to routing domain.

- *Large number of sensors or scalability.* WSNs may consist of hundreds or thousands of sensors that are deployed in the sensing area. The number of nodes in a region (termed node density) depends on the application which the sensors are networked and can range from few to many sensors (Heinzelman et al., 2000). In some applications, the density will be extremely high , but the number of nodes in a region decreases significantly when I apply the sensors for vehicle tracking systems (Adipat and Zhang, 2003). However, any protocol particularly for routing protocols that communicate the sensors must be able to scale to meet the requirement for the different node density. Increasing the number of nodes in the networks affect both the lifetime and the performance of the networks.

- *Limited energy versus Network Lifetime.* The primary goal of WSNs is to extend the lifetime of the network, and to improve the performance of the sensor networks (Al-Karaki and Kamal, 2004). However, as described previously, sensors have limited power supply and these sensors are sometimes placed out in the unattended fields where replacement of battery is not possible. Thus, the network must manage the resource carefully to optimise the energy expenditure as the lifetime of a node is dependent on the battery life. In addition, the energy consumption in WSNs is mostly used for radio transceiver activity. Thus, designing protocols for routing in WSNs must consider this energy constraint while not compromising data transmission.

- *Network self-organisation or fault tolerance.* According to the large number of sensors being connected and collaborated in WSNs, it is important that the networks are capable of reconfiguring and self-organizing the failed sensors (due to energy depletion or from physical destruction) in order to maintain the function of the system. The failure of sensor nodes should not affect the overall task of the sensor network. Furthermore, according to coping with frequent topological changes when new sensor nodes are added to join or leave the system, it requires special routing protocols to reconfigure the network so that a high degree of connectivity must be maintained. The ability of the sensor network to sustain its functionalities without any interruption is called fault tolerance (Hoblos et al., 2000; Iyengar et al., 1994) and the level of fault tolerance depends on the application of the sensor networks.

- *Hardware constraints.* The sensor networks made up a large number of wireless sensor nodes which have limitation on energy, memory, storage capability, processing and low-duty cycle of transmission. Thus designing energy-efficient protocols while satisfying data quality according to the application needs are required to be developed in WSNs.
- *Dynamic environment.* According to the characteristics of WSNs that span a network in a large area, communication between sensors will meet and should adapt to unpredictable wireless environment combined with constraints on the hardware of sensor nodes. Designing protocols for sensor communication that are adaptive with dynamic environment is rarely considered to achieve energy-efficient communication in the densely populated environments (i.e. high node density).

We have discussed the critical design and challenge issues that must be addressed to allow flexibility and efficiency for research efforts in WSNs. These efforts are made for the possible development of protocols in these sensing domains that offer new technological possibilities for many applications that range from societal to economical issues (Vidhyapriya and Vanathi, 2007). For instance, target tracking, habitat monitoring, intelligent transportation system, health-care applications (Ennaji and Boulmalf, 2009; Martinez et al., 2007; Gsottberger et al., 2004; Mainwaring et al., 2002). However, this study first limits the set of target applications in WSNs that could be considered for most potential usage scenarios.

2.4 WSNs application scenarios

Based on data reporting model that represents the time criticality of the sensed data to be delivered, the majority of WSNs applications will fall into scenarios as follows (Hill, 2003).

2.4.1 Monitoring systems

One of the most straightforward application scenarios in WSNs is to monitor remote environments. In this monitoring system, sensor nodes are deployed and configured to perform tasks to periodically sense the conditions of their surroundings and transmit the sensed data to the user in order to get the required information. The sensors surroundings can include environmental detection/monitoring systems (Othman and Shazali, 2012; Mainwaring et al., 2002; Cerpa et al., 2001), fire detection and rescue systems (Sha, Shi and Watkins, 2006), building/home appliances for automation (Herring and Kaplan, 2000) or Residential Laboratory (Essa, 2000) and wireless body area networks for patient monitoring systems (Yu and Tseng, 2007; Jovanov et al., 2005). According to applications in this scenario, the interval period for reporting the conditions of surroundings varies based on the demand of the users. Further, as data samples can be delayed at reasonable period of time, sensors may not need to support real-time transmission thus the lifetime of the sensors can be maximised while meeting the schedule for data reporting. Any failure in the networks due to congestion or energy depletion will trigger such reconfiguration technique.

2.4.2 Security monitoring

Security monitoring differs from the environmental monitoring where each node in this scenario of application only has to transmit a data report when there is a security violation. This node does not need to periodically collect any data from the surrounding environments so it reduces the energy-costs. The node deployment in this scenario is usually fixed so that each node can detect and report to the user if a particular node is not functioning like an alarm system. Unlike environment data collection, this scenario can distribute power consumption evenly using a minimum tree network structure where each node only has one child node. When a violation occurs, the network reports and responds the detected intrusion quickly to reduce the latency otherwise the data will be useless. Thereby the cost for data communication increases as the routing has to monitor the data transmission frequency.

2.4.3 Object tracking system

The topology of this scenario will be dynamic or change due to node mobility. Some applications in this scenario (Ye et al., 2002) tracks the location of nodes that move through the network such as for tracking doctors or patients for ambulatory health system (Rabaey et al., 2000), target tracking for the military (He et al., 2004), etc. Unlike other scenarios, the topology in this object tracking will continually change as the objects enter and leave the network. The applications in here attach the sensors to the object in order to track easily the location of the detected objects. Due to frequent topological changes, it is essential that the networks should be able to detect the occurrence of the node's movement that either enter or leave the networks.

Now, I have determined a set of application scenarios as described above. We use this set of application scenarios to further analyse the requirements for the Quality of Service (QoS) imposed by the applications on these WSNs and due to the unique characteristics of these underlying networks. The following section describes the QoS requirements in WSNs.

2.5 Quality of Service (QoS) requirements in WSNs

Quality of service (QoS) is overused term with different technical meaning depending on which perspective is being used (Ganz et al., 2004). According to the work in (Chen and Varshney, 2004), the QoS is described in two different perspectives (i.e. application and network perspectives). At the application perceptions, QoS is generally overused as the quality of services that the users obtain, while at the network perspective it is to provide a certain grade of service that the users need while maximising the network utilisation. In addition, RFC 2386 (Crawley et al., 1998) defined QoS as a set of service assurance that includes some measurable metrics that the networks to be fulfilled to deliver packets with respect to the application requirements.

In the conventional wireless networks, QoS requirements are designed to support applications that require dedicated bandwidth and end-to-end throughput. It needs such end-to-end QoS parameters like throughput, delay, jitter and loss rate to evaluate the QoS mechanisms. In such networks with higher resources, they provide three levels of QoS such as hard-QoS (guaranteed services), soft-QoS (differentiated services) and best-effort QoS (without any service warranty) for end-to-end user applications (Chen and Varshney, 2004). While in traditional wireless networks which have bandwidth constraints and dynamic topology make the QoS support more challenging (Crawley et al., 1998). They include Mobile Ad Hoc Networks (MANET) (Mohapatra et al., 2003), bandwidth reservations (Wu et al., 2001), GVGrid (Sun et al., 2006), CEDAR (Sivakumar et al., 1999), INORA (Dharmaraju et al., 2002) and AODV and DSDV for mobile networks (Patel and Kulkarni, 2011).

Although extensive studies of QoS in conventional networks are well-conducted, the research of QoS in WSNs suggests a different direction of study due to the nature of

WSNs and the energy constraints embedded in the individual sensor node. First of all, since the WSNs pose constraints on energy, bandwidth and memory (buffer size), QoS in WSNs is more for such energy aware QoS support rather than such end-to-end application models even these models can be applied in some WSNs scenarios. Second, QoS in WSNs is mainly not concerned for applications requiring dedicated bandwidth like in any traditional network. Third, some types of application in WSNs can tolerate packet loss or latency to a certain level due to the data reporting model of specific applications. Finally, WSNs are application centred where the applications are often deployed with different purposes and importance.

In addition, as the traffic in WSNs mainly flows from the numbers of sensors towards one or a small set of destination nodes (i.e. sink or BS). It causes the nodes around the BS to relay heavier traffic loads thereby the energy of these nodes deplete quickly that would end the networks prematurely while the energy in other areas of WSNs would remain unused as shown in Figure 2.2. If I ignore this characteristic in terms of routing methods, the congestion not only wastes the scarce energy, but hampers the reliability and resource utilisation.

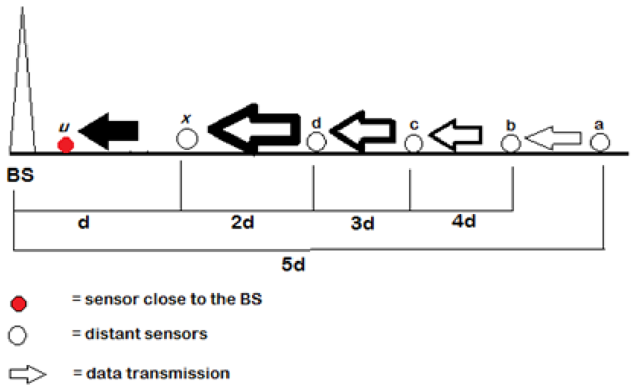

Figure 2.2: The architecture of data transmission in WSNs

As nodes are prone to failure and the node deployment in WSNs is application dependent and can be randomised for some scenarios, in flat routing where such situations occur in which some nodes have failed medium access control (MAC) operations, 'broadcast flooding' or 'packet detouring' may occur frequently that causes unguided packet transmission resulting in the loss of many packets and significant overheads of data communication. This degrades the network performances and energy efficiency as well as the benefits of the routing algorithms as shown in Figure 2.3.

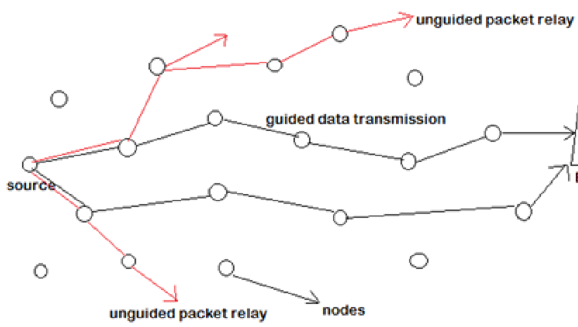

Figure 2.3: The guided versus the unguided packet routing

Due to the unique nature of WSNs, the model of end-to-end QoS parameters based on peer-to-peer applications in conventional networks may not be applied in WSNs (Schweitzer and Tilch, 2002). Therefore the QoS support in WSNs is typically such collective QoS parameters from several metrics such as throughput, delay, bandwidth and packet loss. Hence, different QoS approach should be considered for the WSNs (Basaran and Kang, 2009).

From the aforementioned QoS requirements, the following discussion explores the parameters that will be used to evaluate both the performance and efficiency of the networks for measuring the QoS support to applications and the performance of different topologies that are applied in these networks.

2.6 QoS evaluation metrics

We have established the key evaluation metrics for WSNs to make a reality the vision of WSNs. This section explores a standard set of evaluation metrics. Each metric has impact with each other where decreasing one metric is able to increase another metric. Thus taken together all metrics can be used to examine the WSNs performance and efficiency with respect to QoS since the WSNs are application dependent and resource constraints. The following sections describe these metrics.

2.6.1 Energy usage and lifetime

Energy analysis will be conducted on the proposed protocols, since WSNs are energy constrained. Two types of energy will be examined, namely energy consumption that is represented in the residual energy of nodes and energy distribution. Energy consumption is measured by calculating the total energy used by each sensor in the network for either sensing, computation or communication. Further, the residual energy at each node is very important to determine, because this energy can be used for a routing scheme to select nodes to build a path. The nodes with maximum residual energy have higher possibility to be chosen for reliable data forwarder.

Another important metric is energy distribution. This metric measures the capability of the protocol to balance the energy consumption between nodes in the network to avoid the network ends prematurely. It means that higher number of nodes participating in data delivery reflects the increasing data distribution, and this adversely affects the system lifetime of the network. In this project, we use the most commonly definition about the lifetime of a sensor network as the time to the first node failure (Chen and Zhao, 2005; Vass et al., 2006). Since the lifetime is mostly dependent on the energy supply, a good protocol design should be able to manage carefully the energy of nodes for prolonging the network lifetime.

2.6.2 Coverage and Network Utilisation

Coverage is the evaluation metric for a wireless network to have the ability to support the situation when a network grows larger or as the workload increases (Zhao and Raychaudhuri, 2009). As described in Section 2.3, a good system has to be able to scale to different sizes of WSNs and to adapt to the changes in the network while still considering energy sensitiveness. Therefore, coverage is an important metric to measure how well different protocols perform as node density varies that impact the data transmission models. Another important metric is the optimum number of sensors being active imposed by the application specific needs on the deployment of sensors (Iyer and Kleinrock, 2003). This metric may directly relate to the quality of applications.

2.6.3 Reliability

To get an overall picture of the system or proposed protocol performance, some evaluation will be conducted to examine fault tolerance and packet recovery. It can depict how stable the protocol to maintain the network functions in various situations in the WSNs that are prone to failures. To achieve this, the protocol being applied should be able to tolerate and adapt to individual node failure in order to provide self-organisation capability to meet the demands of the lifetime requirements.

The analysis will use heterogeneous traffic due to different types of applications. This metric is measured as the ratio of the number of packets received by a destination node to the number of packets sent by source nodes. The results of the ratio will be compared to evaluate the performance of the systems/protocols to maintain the function of network for the occurrence of changes in WSNs environments.

2.6.4 Flexibility

The other metrics which are related to evaluating the performance are flexibility. Flexibility is defined as the capability of the WSNs protocol to accommodate a wide range of application scenarios (Pathan et al., 2007) such as target tracking, security system, habitat monitoring, environmental control, traffic monitoring, and health-care applications. This metric measures the capability of the protocol to support multiple sensing capabilities (Dutta et al., 2005; Gu et al., 2005) where each application scenario has its own requirements in relation to latency, traffic rate, reliability, energy dissipation, etc.

2.6.5 Latency

One of the performance measurements in WSNs is conducted to estimate end-to-end delay performance. The higher tasks such as multimedia or target tracking applications require bounded latency to deliver data to the BS, otherwise data will be worthless. During data transmission, a node will take a certain period of time to process a packet before transmitting the packet to the BS. The time taken by a node to forward a packet until the packet arrives at the destination node is generally called a node delay. End-to-end delay performance can be estimated by calculating the node delay to process a packet, and the delay time needed to propagate a packet from the source node to the BS. During the research, the delay performance will be examined in relation to all traffic conditions to get optimised results.

2.6.6 Throughput

In general, throughput is measured as the total number of data received successfully by the BS. Higher end-to-end throughput represents the performance of routing technique becoming more effective to minimise packet loss. On the other hand, current data delivery models in the WSNs are not end-to-end applications, but collective parameters. Measuring throughput of each node is thus needed, because it can provide information about the congestion status of each node. When the throughput at a node is high, it indicates that the node can avoid congestion well. Hence, the proposed scheme can select the nodes with higher throughput to build paths. By using the analyses explained above, the results of the proposed protocol can be compared with other existing protocols in terms of throughput performance.

2.6.7 Congestion rate

Congestion is one of the important QoS parameters that is relevant to improve the QoS and to achieve the target of the set of applications. Congestion is produced by creating data flows from several sources with different traffic rates (i.e. total packet per-second). Different application in WSNs produces different data sample rate. For applications posed by video and imaging sensory that requires real-time and dedicated bandwidth, the system sets the data collection rate to increase with higher volume. Additionally, for environmental data collection applications typically needs only 1-3 data samples per-minute. However, in a military such as battlefield, the sensors have to capture and send huge data with higher sampling rates up until 30 samples in a second.

I have identified that these networks have limitations on power supply, computation process, memory and both transmission range and bandwidth of the wireless link. The embedded constraints on hardware platform bring various WSNs requirements that should be considered in their core design issues. Due to the WSNs inherent characteristics and the growing interests of WSNs applications, many research efforts have been proposed to establish paths for carrying out the sensory data from the sources to the destination (i.e. BS) with energy sensitiveness while attempting to cope with frequent and unpredictable topological changes in WSNs. The following section describes the state of the arts of the routing protocols in details with various approaches.

2.7 Routing designs and challenges in WSNs

Routing becomes necessary in WSNs as routing consumes the most energy in WSNs (Ares et al., 2007; Goodman, 1997) and is responsible to fulfil one of the main tasks of the WSNs that is to disseminate data from sources to a destination in a multi-hop way. The aim of this section is to discuss and review understanding of the routing protocols as well as routing design challenges and identify potential research issues that can be further pursued.

Routing in WSNs has specific requirements due to the nature and dynamic topology of this sensing technology. They are as follows: (i) large number of nodes (ii) densely deployed (iii) prone to failure (iv) resource constraints (iii) application-specific based (iv) intrinsic data delivery model, (v) frequent topological changes and (vi) data centric. The design of WSNs is influenced by many factors (Akyildiz et al., 2002), and these factors may have a conflicting impact on each other so that trade-off decisions are considered to achieve an efficient data communication in WSNs. The following summarises the design and the challenge issues that affect the routing process.

- *Node deployment.* Sensor nodes deployment can be either deterministic or randomised. It depends on the applications, and this deployment affects the network topology. The data will be easily transmitted through a pre-defined path when the nodes are placed manually. In contrast, for some applications that require the nodes are scattered randomly, it is highly recommended to perform such clusterring techniques for sensor nodes, so that it can still allow the network connectivity and perform energy efficient communication.

- *Resource constraints.* Sensor nodes have an energy constraint that is dependent on the lifetime of the battery (Heinzelman et al., 2000) where replacing of battery is sometimes difficult to be done in some applications that place sensors in unattended sensing fields. In WSNs, the communication is conducted both in short-range and multi-hop ways thus each sensor acts as both a data sender and a data router for data transmission. This causes the fast depletion of energy. This node failure may affect the network topology, so a fault tolerance routing approach is needed to maintain packet routings and reorganise the topological changes.

- *Data reporting model.* According to the way the sensors sense and deliver the data based on the types of applications, WSNs commonly classify the data reporting model into four categories (Yao and Gehrke, 2002): a) continuous-driven b) event-driven c) on-demand driven and d) Hybrid. These data delivery models highly influence the routing protocol designs as they can affect the process of data delivery from sources to the BS. In addition, the routing protocol is highly influenced by the energy consumption and the types of application services.

- *Heterogeneity.* WSNs can be distinguished based on the type of sensors being deployed in the network. If the deployed sensors have similar capacity in terms of sensing, communication and energy supply, then it is called homogeneous network. Almost any application in WSNs could be classified as homogeneous. However, there is an increasing reliance on the deployment of sensor networks with multiple sensing capabilities (Dutta et al., 2005; Gu et al., 2005) for many applications where each sensor might has different functionalities for sensing. This is called heterogeneous network whereas the heterogeneity node produces data being transmitted at different rates as well as in various sizes of packets (Al-Karaki and Kamal, 2004)

- *Scalability.* This problem would appear, when the number of sensor nodes deployed reaches an extreme value of millions with different sets of applications purposes. Any routing scheme must be able to support this huge number of sensors and the problems of node coverage.

- *Connectiveness and Responsiveness.* Another critical issue that could arise in WSNs is connectivity due to the increasing number of nodes deployed in the sensing fields because high node density may cause sensors being isolated from each other. The density of WSNs is relatively high when the network grows larger at hundreds of nodes. A good routing protocol should be scalable enough with this large networks and not affected from the topological changes due to failures of nodes caused by physical damages or lack of power. The ability of the network to quickly adapt to the changes is commonly called responsiveness.

- *Network dynamic.* Most of the WSNs assume that the sensor nodes are stationery. However, mobility becomes more challenging in the WSNs (Ye et al., 2002), in relation with resource constraints that sensors have. Further, mobility causes frequent changes in the network topology as the nodes may enter and leave the network any time. Therefore route stability becomes an important issue in WSNs.

- *Data quality.* Data quality or quality of service is dependent on the application purposes. Some applications require bounded latency, while others focus more on data reliability, and the rests concern on energy conservation than getting high quality data.

Regarding the factors explained above, specific routing tactics are explored and reviewed in order to classify each routing protocol that fit the routing issues stated above.

2.8 Routing techniques in WSNs

In general, some comprehensive surveys from the last decade up to now divide the process of routing in WSNs into the network structure and the protocol operations (Patil and Biradar, 2012; Kumar et al., 2012; Goyal and Tripathy, 2012; Zhaohua et al., 2010; Al-Karaki and Kamal, 2004). However, in this research works, due to the issues of routing in WSNs as described in Section 2.1, I then identify and study various types of routing protocols of WSNs with respect to all the mentioned routing issues such as energy efficiency and data quality. The following describes different routing protocols with their disadvantages and drawbacks in detail following the structure of each routing issue mentioned above.

2.8.1 Energy-efficient routing protocol

The development of routing protocols in WSNs should take into consideration some factors including the application needs, the network architecture, node deployment, but the most important factor is energy efficiency as it affects the lifetime of the network. Hence, presenting energy-aware routing is required to meet the main objective of WSNs, to prolong the function of the sensor in performing the tasks (sensing, computing and communicating). In the literature routing, most routing protocols mainly focus on energy efficiency (Kordafshari et al., 2009b; Khamforoosh and Khamforoush, 2009; Daabaj et al., 2009; Zhang, He and Jiang, 2008). Further, the applied routing protocols have been established to provide efficient routing approaches that deliver data to the destination (BS) with minimised energy consumption in order to maintain the lifetime of the network. These protocols describe their approaches related to routing to save and use energy efficiently.

In the following subsection, I discuss the sTReNsgths and the weaknesses of each protocol in terms of its approach in conducting such a way for achieving energy-efficient routing for WSNs. Moreover, I provide a comparison amongst these protocols to help the engineers to enhance the capability of each protocol on which issues of operations can be extended for better energy-efficient routing algorithms. In here, I classify these energy-efficient routing protocols into the two main categories, that is modified from the survey in (Al-Karaki and Kamal, 2004). The first category that is Network Structure can be further classified as Flat, Hierarchical and Location-based routing. While the second category that is Communication Model consists of negotiation, multi-path and query -based routing.

According to the first category, routing strategies can be highly influenced by the way the nodes are connected and how each node plays its role to route packets based on the network structure. This means the node connection can be configured with the same level or with different hierarchies and further determined dependent on the node's location. The following describes the protocols due to network structure category.

Flat Routing

In a Flat network, data is transmitted in a multi-hop fashion and each node plays the same role in sensing and transmission, so that there is no superior node that acts and computes overload operations as depicted in Figure 2.4. The protocols in the flat network commonly provide good-quality routes for data delivery from source nodes to sink nodes by using flooding. A technique that broadcasts packets to all nodes in the network until the packets reach the destination (BS). Flooding is simple, robust as guarantee data delivery and fundamental communication primitives for flat routing in WSNs (Al-Karaki and Kamal, 2004). Flooding is suitable for a battlefield application, but it has drawbacks in terms of overlap and implosion that can waste energy due to data redundancy and duplication. This can therefore be termed as an energy inefficiency technique (Lim and Kim, 2001).

However, this flooding technique has motivated many other protocols such as Wireless Routing Protocol (WRP) (Murthy and Garcia-Luna-Aceves, 1996), The Topology Dissemination Based on Reverse-Path Forwarding Protocol (TBRPF) (Bellur and Ogier, 1999), Re-active or Source-Initiated On-Demand Routing Protocols (Pucha et al., 2007), Temporarily Ordered Routing Algorithm (TORA) (Amer and Hamilton, 2008; Park and Corson, 1997), Gossiping (Hedetniemi et al., 1988) and Rumor Routing (Braginsky and

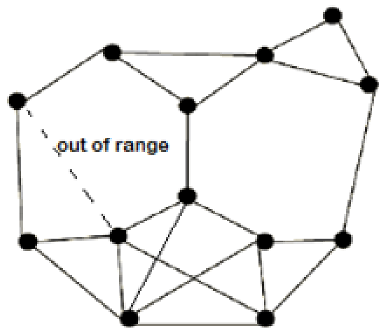

Figure 2.4: Flat routing structure

Estrin, 2002). These protocols use different approaches to address the drawbacks of Flooding due to resource consuming and to maintain the robustness of the network. However, they do not take into account other QoS routing metrics (i.e. throughput, latency and flexibility for supporting different application requirements).

Hierarchical Routing

A hierarchical network utilises a cluster zone approach that organises nodes being grouped into clusters. This routing hierarchy reduces the complexity for a large-scale WSNs thereby improving routing scalability. One of the sensor nodes in a cluster zone will be selected as a cluster head which has tasks to aggregate messages coming from member nodes and to transmit the aggregated data to the BS as shown in Figure 2.5. Thus, reducing traffic overhead and performing efficient energy consumption. The protocols in hierarchical network include Low Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman et al., 2000), Power-Effcient Gathering in Sensor Information Systems (PEGASIS) (Lindsey and Raghavendra, 2002), Threshold-sensitive Energy Efficient Protocols (TEEN) and Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol (APTEEN) (Manjeshwar et al., 2002), Small Minimum Energy Communication Network (MECN) (Rodoplu et al., 1999), Self Organizing Protocol (SOP) (Subramanian and Katz, 2000), Sensor Aggregates Routing (Medeiros et al., 2007), Virtual Grid Architecture routing (VGA) (Al-Karaki et al., 2004), Hierarchical Power-aware Routing (HPAR) (Li et al., 2001) and and Two-Tier Data Dissemination (TTDD) (Ye et al., 2002).

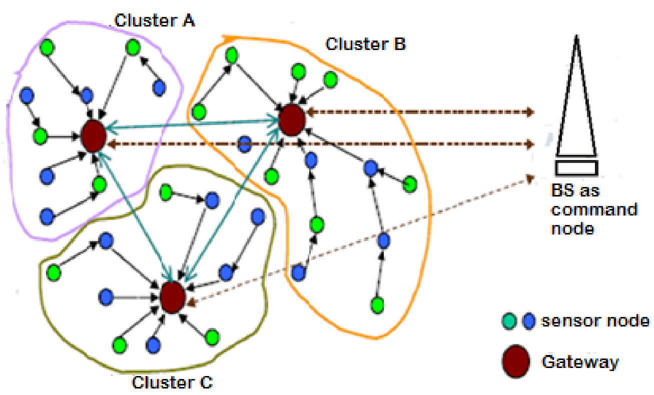

Figure 2.5: Hierarchical routing structure
redrawn from (Heinzelman et al., 2000)

Each protocol mentioned above try to save energy consumption to achieve energy efficiency and maximise the network lifetime. Each protocol is developed for specific reasons like for sudden changes (TEEN/APTEEN), mobile environment (TTDD), limited network. Despite the benefits of hierarchical routing to save energy amongst the sensors by clustering the area of sensors, this routing technique has drawbacks as follows:

- clustering zone introduces additional overheads moreover in APTEEN protocol (Manjeshwar et al., 2002) that uses the threshold function for triggering data transmission.
- clusterig zone also causes excessive delay and complexity especially for large-scale WSNs.

Location based Routing

In location-based networks, the routing is initiated based on the locations of sensor nodes. The locations are usually determined from GPS devices or multilateration measurement methods by exchanging location information between neighbours (Savvides et al., 2001; Capkun et al., 2001). The location-based protocols try to save energy by forcing the nodes to sleep when there is no activity. This can increase the durability of the network function. In general, the location-based routing aims to use the geographic information of the nodes to route data by reducing the communication cost while achieving data quality. The following approaches have been described on how the information of location can be used (Stojmenovic, 2002).

1. To forward packets in greedy mode. This greedy mode approach commonly select next nodes based on three techniques; first, selecting nodes whose distances are close to the sender node like Greedy Other Adaptive Face Routing (GOAFR) (Kuhn et al., 2003); second, finding nodes who have locations at the edges of the graph when no closer neighbours are available. Greedy Perimeter Stateless Routing (GPSR) (Karp and Kung, 2000) is one of the routing protocol for this technique. The third technique is determining nodes which have distances close to the destination (i.e. BS) as Geographic and Energy Aware Routing (GEAR) has done (Yu et al., 2001)
2. To divide the routing area into smaller rectangle zone (grid) and perform packet routing in a grid-by-grid manner. The protocols in this approach include Geographic Adaptive Fidelity (GAF) (Xu et al., 2001) and Grid (Liao et al., 2001) where the GRID approach is suitable for a large and dense network.
3. To exploit the location-based information to limit the area of packet routing to achieve route optimisation while reducing communication overheads. The two early protocols in this approach include location-aided routing protocol (LAR) (Ko and Vaidya, 2000) and DREAM (Basagni et al., 1998). This zone can be improved by adapting the shape based on the speed of mobile node. They represent three shapes: rectangle, bar and fan.

The above-mentioned protocols use the location of nodes to find efficient paths for delivering packets to the destination (BS) by using i) the shortest path, ii) the nearest neighbour, iii) the nodes with highest level or iv) any approach that aims to balance energy dissipation during transmission. Therefore they attempt to minimise energy consumption of the nodes to extend the nodes' lifetime. However, the location-based protocols do not consider all the important attributes of the QoS service.

Multipath-based routing

Routing with the multipath-based approach uses more than one path to deliver packets from the source to the destination nodes. When one of the paths as the current path fails, other paths can be selected as back up paths to continue the packets transmission. This tries to achieve network reliability, but it consumes more energy either to maintain the alternate paths or to transmit copied packets through many other paths. This multipath-based approach also increases the traffics because the same packets forward on several paths. Hence, a trade-off between the amount of traffic and the reliability of the network should be further studied (Dulman et al., 2003). Some protocols like Gradient Broadcast (GRAB) (Salber et al., 1999) is designed specifically for robust data transmission that use multiple nodes for data transmission without dependency on any individual node. In Hierarchy-Based Multipath Routing Protocol (HMRP) (Wang et al., 2005) and Cluster based-Multi Path Routing (CBMPR) (Dey, 2001), they combine clustertering technique and multipath routing to achieve energy efficiency by providing paths that can decrease routing overheads and energy consumption while improving the network scalability. Although multipath approach wastes bandwidth usage, it is a good alternative for recovery from failures in the WSNs. Many protocols in the WSNs use this multipath approach (Chang and Tassiulas, 2004; Dulman et al., 2003; Shah and Rabaey, 2002; Ganesan et al., 2001; Intanagonwiwat et al., 2000).

Query-based routing

In query-based approach, the energy conservation (lower energy consumption for data routing and aggregation) is achieved by performing data delivery based on the query of the destination nodes. The query is propagated as the BSs interest throughout the network, if the source node has data matching the interest, the source sends data along the selected gradient path to the destination node that initiates the query as shown in Figure 2.6. This kind of routing includes Directed Diffusion (DD) (Intanagonwiwat et al., 2000) and Rumor Routing (RR) (Braginsky and Estrin, 2002). The drawback of the approach is it costs much energy for matching the sensed data with the query moreover if the number of events is small like in Rumor Routing. This query approach may not be applied for applications that require continuous data delivery.

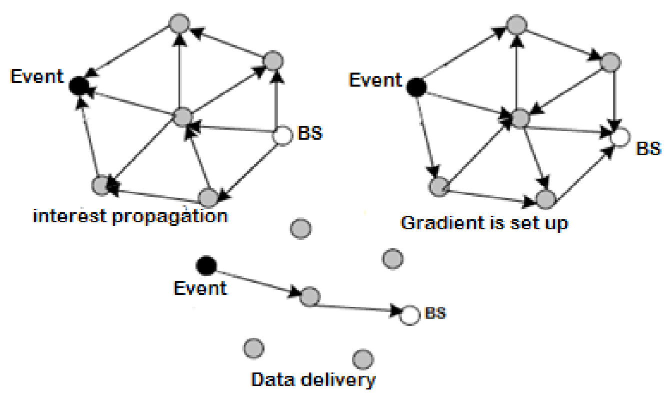

Figure 2.6: Gradient based approach
redrawn from (Intanagonwiwat et al., 2000)

The other query-based protocols include COUGAR (Yao and Gehrke, 2002) and ACQUIRE (Sadagopan et al., 2003). COUGAR uses additional abstraction for the query according to leader selection for query and aggregation in order to provide energy efficiency for huge traffic generation. This additional query causes extra overheads. While, ACQUIRE attempts to make complex queries more efficient by splitting the queries into many sub-queries. This technique is ideal for one-shot service but, it causes flooding in the network. These protocols can extend the lifetime of the network by reducing the number of transmission with the query approach but they mainly focus on energy efficiency not other QoS performance in a generic way.

Negotiation-based routing

This approach uses meta-data (level of data descriptor) as a series of negotiation (Kulik et al., 2002) to pursue a data-centric routing technique (Heinzelman et al., 1999). This meta-data approach, termed Sensor Protocols for Information via Negotiation (SPIN) (Kulik et al., 2002; Heinzelman et al., 1999) uses an attribute-based naming data that specifies the properties of data in response to the queries of the BS for particular events in order to reduce implosion and to eliminate redundant data as depicted in Figure 2.7.

The main idea of SPIN is i) to reduce traffic flows and energy usage by performing data routing between nodes if they really need the data to obtain and ii) to provide energy conservation by monitoring and adapting to the changes in the energy levels of the nodes to extend the network lifetime (van Bunningen et al., 2006). Referring to Figure 2.7, SPIN exploits three-way handshake model by using a 3-stage protocol ; advertisement (ADV), request (REQ) and DATA. SPIN aims to ensure that only the required nodes will receive the information from neighbouring nodes. When a SPIN node has the targeted data, it broadcasts an ADV message containing meta-data. If a neighbour is interested in data, it sends a REQ message to request DATA. The process is repeated to other neighbours until it arrives at the destination node.

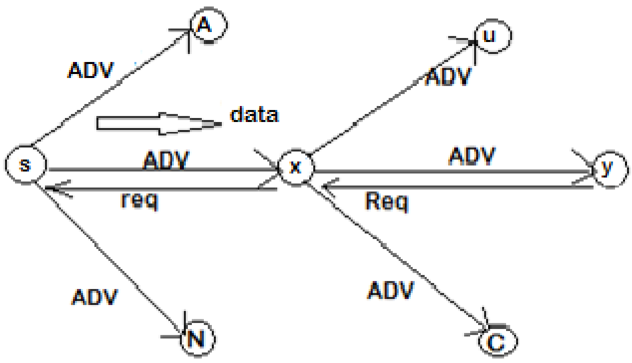

Figure 2.7: Meta-data negotiation based approach

There are four types of SPIN protocols i.e. SPIN-PP (SPIN for point to point communication), SPIN-EC (SPIN for energy conservation), SPIN-BC (SPIN for broadcast network) and SPIN-RL (SPIN with reliability) (Cordeiro and Agrawal, 2006). The SPIN-PP is a 3-stage protocol as described above. The SPIN-EC is an extended model of SPIN-PP, which adds an energy threshold mechanism, to ascertain that the energy required to complete the process of data delivery is sufficient. SPIN-BC and SPIN-RL extend their approaches for disseminating data for better network performance even the approaches encounter overheads with respect to delay and time consuming. The advantage of the SPIN family as negotiation-based protocols is their ability to reduce and save energy transmission by eliminating redundant data and aware of node mobility. However, the problem appears when the intermediate nodes between the source and the destination nodes are not interested in the data resulting in unguarantee data delivery that affects throughput, reliability and increased.

We have presented a comparison between routing techniques as described above that have common objectives of minimising energy consumption to prolong the lifetime of the network by considering energy awareness in every aspect of the operation of data communication. However they have not been designed to take into consideration other QoS metric performances such as reliability, latency, scalability and flexibility to cater the requirements of the applications in WSNs.

2.8.2 Data Quality-based routing protocol

Unlike energy-efficient routing, the main aim of the QoS protocol is to achieve data quality and to satisfy the requirements of applications. Data quality is measured from QoS metrics such as throughput, delay, network utilisation and bandwidth. However, since sensor nodes are energy constraint, this QoS protocol should balance energy consumption while satisfying certain QoS requirements to meet the requirement of a connection oriented communication for individual connection (Al-Karaki and Kamal, 2004; Schilit and Theimer, 1994). However, since sensor has limitations on energy supply and device restrictions, the established QoS routing protocols for other wireless networks cannot be directly applied to WSNs.

Early QoS based protocols such as Sequential Assignment Routing (SAR) (Sohrabi et al., 2000) and SPEED (He et al., 2003) tried to achieve data quality with different approaches. SAR used the notions of QoS to ensure fault tolerance and easy recovery, while SPEED further concerned end-to-end delay estimation. The following subsections present and summarise various works on QoS routing protocols that have been published with the QoS issues that are being addressed.

Sequential Assignment Routing (SAR)

SAR is one of the early QoS routings (Sohrabi et al., 2000) that introduces the ideas of QoS, energy resources and the priority level of each packet in its routing decision. This protocol uses a tree rooted approach to build multi routes to avoid single route failures. The tree paths start at the source to the destination nodes, so that at the end of this route setup, each node will be part of the tree structures multipath. Nodes with low energy or QoS guarantees will not be selected in the tree path. Hence, this routing protocol can ensure fault tolerance. When the network fails, path re-computation is needed. BS also can trigger the nodes to run a periodic re-computation to prevent potential failure. Other recovery procedures can be initiated by a handshake procedure between the neighbouring nodes. Although SAR uses the notion of energy, and can support recovery, it has to maintain the tables and states of each node, particularly when the number of nodes increases significantly.

SPEED

SPEED is another early QoS routing protocol that provides end-to-end guarantees for certain applications. It maintains information about its neighbours, and exploits a geographic forwarding technique to establish paths (He et al., 2003). Before packet transmission, SPEED ensures a certain speed of each packet, so the application can estimate end-to-end delay for each packet. The link delay estimation is measured by dividing the distance between the source and the BS by the speed of each packet using Exponential Weighted Moving Average. The delay measurement also includes delay estimation at a node, which is determined by calculating the elapsed time between a node that sends a packet and a node that receives an ACK message from its neighbours as a response to a transmitted data packet. By looking at the delay estimation at a node, SPEED will select the nodes with delay values that meet the SPEEDs requirements.

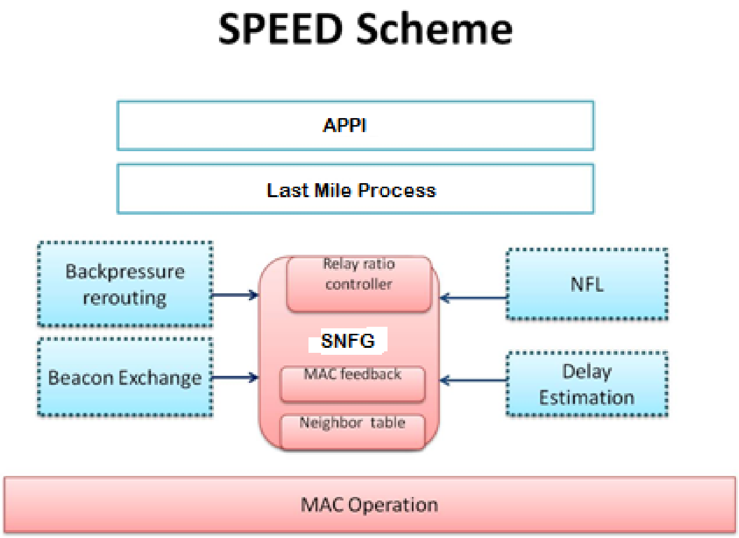

Figure 2.8: Components of SPEED Protocol
redrawn from (He et al., 2003)

However, when the nodes are congested and not able to meet the required data velocity, SPEED uses a backpressure re-routing mechanism to stop data transmission toward the congested nodes and finds other routes that can maintain the required speed. Hence, SPEED can provide congestion avoidance when potential congestion occurs as described in Figure 2.8. Therefore, SPEED performs better in terms of end-to-end delay performance compared with SAR. Moreover, in order to maintain the uniform delivery speed, this protocol exploits a neighbourhood feedback loop (NFL) method that calculates the relay ratio controller to provide the information about the miss ratio of the neighbours as described in Figure 2.9.

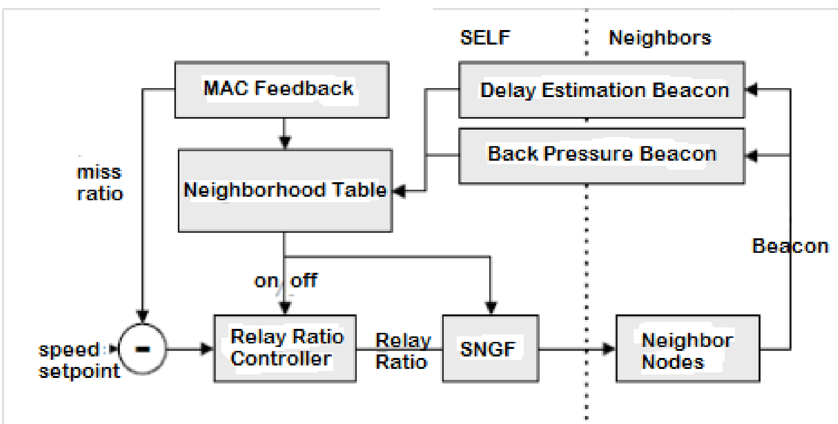

Figure 2.9: The neighbourhood Feedback Loop (NFL)
redrawn from (He et al., 2003)

According to the figure 2.9, the miss ratio is encountered when a packet delivered to a next node is lower than the required relay speed. The NFL aims to maintain the data delivery by selecting the nodes that have closer distance to the BS and minimum miss ratio for forwarding data packets. However, SPEED does not include energy metric in its routing decision, thus comparing SPEED with an energy-aware routing algorithm is needed.

RAP

RAP is a real-time communication protocol for large-scale sensor networks that provides high-level query and event services for distributed WSNs. RAP is supported by a network stack including a transport-layer Location-Addressed Protocol (LAP), a Geographic Forwarding (GF) routing protocol, a Velocity Monotonic (packet) Scheduling (VMS) layer, and a prioritised MAC (Lu et al., 2002). This light-weight and comprehensive network stack makes the RAP as a powerful and efficient routing protocol that provides a localised and efficient approach in reducing the end-to-end deadline miss ratio. The LAP in RAP is similar to UDP as a connectionless transport layer, but the packets are addressed based on the location of the nodes. This knowledge of the node's location is used by the GF routing algorithm to route a packet to a neighbour based on a greedy decision with respect to finding a next-hop node that has the closest distance to the destination. Further, the VMS as a novel packet scheduling proposed in RAP, takes into account for both time and distance constraints to provide a real-time monitoring for sensors communication scheduling in a large WSN. The VMS provides two policies according to the packet scheduling. They are Static Velocity Monotonic (SVM) and Dynamic Velocity Monotonic (DVM), depending on the requested velocity of a packet that will be transmitted. According to the requested velocity of a packet, the VSM assigns different priority for packet queuing when there are multiple outstanding packets.

MCF (Minimum Cost Forwarding)

This protocol attempts to find the minimum cost path for data forwarding to the destination node (Ye et al., 2001). The cost function is a cost for the combination function of the throughput, delay and energy usage to deliver a data packet from any node in the network to the destination (i.e. BS or sink). MCF is a simple algorithm but it can support data delivery for a large sensor network. In this protocol, the BS initiates the cost values of each node in the network and broadcasts this cost value across the network. Every node receiving the initiated cost from the BS will update the value by adding its estimate cost for using its channel wireless link for packet transmission. The new cost value will then be exchanged to find the optimal cost of all nodes to the BS. Currently, each node has its own cost value to the BS. When the nodes receive data from the source node, it evaluates the cost of the data packet and compare with the node's link cost to the BS. if the remaining cost is sufficient to reach the sink node, the packet will be forwarded to its neighbour node otherwise the packet will be discarded. This protocol aims to achieve optimal data forwarding with minimum number of overhead messages thereby it can save energy while considering latency.

Energy Aware routing

Akkaya and Younis in (Akkaya and Younis, 2003) propose an energy efficient QoS routing technique that supports the introduction of video and imaging sensors that pose additional challenges. This protocol aims to find a least-cost and energy-efficient path for real time data that meets certain end-to-end delay requirements, while maximizing the throughput of non-real time data by both adjusting r-value of bandwidth proportion and employing a class-based queuing model. The queuing model allows service sharing for real-time and non-real-time traffic. In addition, the link cost is defined as a function that includes the nodes energy reserve, transmission energy, error rate and other communication parameters. This protocol performs well with respect to QoS and energy notion. However, the protocol degrades its performance when the traffic increases and cannot find the optimum r-value for bandwidth sharing.

MMSPEED (Multi-Path Multi-Speed Protocol)

MMSPEED is the extended version of SPEED that provides multi path multi speed of packet relay for the support of QoS services in terms of reliability and latency (Felemban et al., 2006). The multi-path data delivery is performed to meet different reliability requirements while the multi-speed is provided in order for the packet to choose a certain speed based on the end to end deadlines. Hence, MMSPEED allows packet to choose the appropriate combination of path and transmission speed for data delivery depending on the reliability and latency requirement. As a result, MMSPEED can support for both periodic and non-periodic traffic flows generated from sudden events. Like SPEED, this extended version uses localised geographic forwarding to initiate routing that runs without doing path maintenance and end to end path discovery thus it reduces the complexity and energy consumption for larger sensor network with the compensation of lack accuracy for the successful packet transmission. Furthermore, MMSPSEED does not consider energy notion in its routing decision thus efficient power routing will be out of scope.

ReInForM

The aim of this protocol is to provide load balancing for packet transmission by forwarding packets and their copies of data to all the nodes randomly between the source nodes and the destination (BS). Thus the ReInForM attempts to provide even energy dissipation in order for the nodes, in the network to deplete their energy at the similar time while it also tries to provide a desired reliability by using multipath approach, network condition attribute and packet priority for each packet forwarding (Deb et al., 2003). The source sends multiple copies of each packet through multiple paths from source to sink depending on the local knowledge of channel error rate and the information of the network condition attribute that are inserted in the packet header. However, the information of the packet header will be updated as the condition of the paths between the source and the destination changes periodically. Moreover, in order to reduce power consumption, ReInForM does not allow the nodes to perform even data caching or data aggregation that against the nature of wireless sensor nodes, which have limited memory.

Mobicast

This protocol aims to provide accurate location tracking for a mobile object that always changes its position by using a flooding avoidance technique (Tsai et al., 2007). Thus, it can save the power consumption of the nodes and further extending the lifetime of the networks. In this protocol, the mobile object is called a target while the tracker is the source nodes that also moves from one place to another position in the network. This protocol uses geographic location information to perform the tracking by using face routing based on the concept of Gabriel Graph for getting accuracy. The method of tracking the location of the mobile target by the source nodes is performed by using a beacon message that is transmitted from the sensor nodes around the target that obtain and keep the track information of the target. By using this beacon message, the source node does not need to broadcast frequent request to obtain the location of the target that moves randomly in the network. In contrast, when the source reaches the target, the sensor also does not require to send the information of the present target location. The beacon message contains information about the current location, the velocity and the moving direction of the target. The experimental results show that Mobicast can locate the object accurately while considering energy efficiency compared to other flooding based protocols for mobile object tracking.

DAST (Directed Alternative Spanning Tree)

This protocol is a typical tree-based routing that considers three important QoS metrics such as energy efficiency, traffic flow and reliability to provide a more efficient packet forwarding using a Markov algorithm (Ji et al., 2008). The Markov based communication state predicting algorithm is exploited in the tree routing to estimate the parent and double parent for packet transmission by using different algorithm. DAST allows each node in the network to role itself with different functions such as data aggregator and data forwarder thus improving the effectiveness of energy usage.

MCMP (Multi Constrained QoS Multi-Path routing)

This protocol aims to use multiple paths in data delivery in order to meet different QoS requirements in terms of reliability and latency (Huang and Fang, 2008). For each path, the protocol attempts to find a shortest path with minimum number of hops while considering the energy cost to satisfy a certain QoS requirement. However, the process of broadcasting location update packet for the required QoS may lead to increase the overhead that wastes energy consumption in some cases. For example, in the case of end-to-end delay requirement, the protocol needs some computation of linear integer programming to formulate the optimisation problem.

MCBR (Message-Initiated Constrained-Based routing)

The MCBR (Zhang and Fromherz, 2004) protocol aims to use the meta routing strategy to route packets to the destinations based on some constraints. The constraints are according to constraint-based destinations, route constraint and QoS requirements for packets. The protocol use different route to deliver a data packet from the source to the destination via a route that satisfies specific QoS requirement. This can be performed by allowing the packet (data) to discover and learn its route during the process of data transmission to the destination. The meta routing strategy has two types of strategies. They are using a searching and controlled flooding strategy. However, based on the simulation results (Sumathi and Srinivas, 2012), the use of flooding-based strategy in MCBR may cause some problems i) additional overhead and ii) collision in the network thereby leading to the increased energy consumption even the energy cost for running controlled flooding is lower than that in MCMP (Huang and Fang, 2008) and MMSPEED (Felemban et al., 2006).

ECMP (Energy-Constrained Multi-Path routing

This protocol extends the MCMP to achieve energy efficiency for QoS routing in WSNs by selecting a path with minimum number of hops only when the path consumes minimum energy or a non-shortest path with minimum energy. ECMP formulates the routing problem as a QoS routing constraint according to reliability, delay and geo-spatial energy consumption (Bagula and Mazandu, 2008). It transforms the path-based model into the link-based model by using the zero-one optimisation framework in order to satisfy both the quality of path and the minimum energy cost.

EQSR (Energy Efficient and QoS aware Multipath based routing)

This protocol aims to provide service differentiation to support both real-time traffic and non real-time traffic (Ben-Othman and Yahya, 2010). The service differentiation provides more bandwidth allocation and assigns higher priority in its multi-queuing policy for real time packet transmission over the non real-time traffic. The path in this protocol is constructed by finding nodes depending on the residual energy, availability of buffer size and signal-to-noise ratio. For a large amount of information, the EQSR splits up the message information into a number of packets, inserts a Forward Error Correction (FEC) into the packet and transmit all the packets over multiple paths to ensure the packets reach the destination. Thus, this FEC code is used to provide fault recovery without incurring excessive delay against the need for performing packet retransmission. Hence a reliable transmission for both real-time and non real-time traffic can be provided efficiently by using the service differentiation. However, this service only differentiates for two types of data classes only (i.e. real-time and non real-time data) where it is not possible to cater the requirements of many applications in WSNs.

QuEST (QoS-based Energy Efficient Sensor Routing)

This protocol attempts to provide a multi-objective QoS routing by optimising multiple QoS parameters such as end-to-end delay, bandwidth requirements and energy consumption (Saxena et al., 2009). This Qos based protocol uses a Multi-Objective Genetic Algorithm (MOGA) to determine near optimal routes for certain QoS requirement even it has limited information about the network condition.

EE-SPEED (Energy Efficient SPEED)

This protocol extends the capability of SPEED by including energy metric in its routing decision. EE-SPEED provides energy-efficient paths that meet the requirement of the delay constrained in real-time traffic. The path is created based on a weight function as a combination metrics of three parameters (i.e. energy, delay and packet relay speed) (Kordafshari et al., 2009b). The weight function in EE-SPEED is formulated by using the model in (Sha, Du and Shi, 2006) where $f(weight) = \max(\alpha En + \beta Speed + \theta Delay)$. The nodes which has the highest value of the weight function will be selected as the next-hop data forwarder. By including energy aspect in its routing decision, this protocol aims to increase the lifetime of the network by routing data through nodes that have sufficient energy to the destination. However, the disadvantage of this new method is to find a next-hop node that meet the requirement of the weight for guaranteeing successful data delivery in situations of i) many nodes depleting their energy and ii) in a sparse or isolated network.

Beside the established QoS routing protocol mentioned above, some works have been done to emphasise the literature on QoS routing schemes. They only focus on limited number of aspects of QoS such as end-to-end delay and packet loss performances (Ennaji and Boulmalf, 2009), data reliability with multipath approaches (Liang et al., 2009; Zhang, Sivanand, Chin and Chung, 2008; Chen and Nasser, 2008) and adaptive QoS and energy aware routing schemes (Sen and Ukil, 2009; Peng et al., 2008). However, recent works described the development of QoS for wireless multimedia sensor networks (Almalkawi et al., 2010; Fapojuwo and Cano-Tinoco, 2009; Hamid et al., 2008) that deliver multimedia content due to the availability of inexpensive CMOS cameras and microphones combined with advanced digital signal processing in sensor nodes. This work aims to address the need for applying QoS in multimedia sensor environments that needs more efforts to balance between energy consumption and dedicated bandwidth.

From the above discussion about QoS routing protocols, Table 2.2 summarises the comparative analysis of the all QoS routing protocols described above. The protocols are compared using four important parameters. They are energy awareness, different services, the problem caused by the nature data communication in WSN and the flexibility for accommodating many applications. The following describes the metrics briefly.

Energy Awareness (En_aware)

is the most important constraint and performance metric for WSNs due to the limited energy resources of the sensor nodes. As the communication in WSNs is the most consumptive energy activity for the sensors, and hence the inclusion of energy notion in the routing decision for WSN communication protocols can play a significant role for improving the energy efficiency thus extending lifetime of the WSN (Al-Karaki and Kamal, 2004).

Service Differentiation (Service-diff)

is a metric to evaluate the adaptability of the routing protocol to enable traffic differentiation by differentiating QoS requirements according to the data type. This metric evaluates the capability of the protocol in facilitating route adaptation based on the type of data that may have different priorities and scheduling policies.

Multihop Routing Problem (Multihop prob)

is defined as the parameter for evaluating both the purpose and the capability of the routing protocols in addressing the occurring problem caused by the nature data communication in WSNs. As described in Figure 2.2 above, the routing in the sensor network is a typical many to one network that delivers data from many source nodes to a single destination (BS). It may lead to the occurrence of the energy hole problem: a situation when the nodes around the BS tend to deplete their energy faster than other sensor nodes in other regions because they have to relay more packets from remote nodes thus resulting in an early dysfunction of the entire network. The second problem is about the broadcast flooding or packet detouring due to the characteristic of link layer protocol and random node deployment of WSNs as shown in Figure 2.3.

Flexibility

is a metric to demonstrate the capability of the routing protocols in supporting multiple sensing capabilities that accommodate different application requirements in a single generic routing approach.

According to the above-mentioned metrics, the protocols are evaluated by highlighting a mark or description for each parameter according to the performance of the protocol. The result of the evaluation will be used to find a gap or new issues that highlight the challenging future research directions.

According to Table 2.2, we use the definition of "Yes", "Limited" and "No" to describe the performance of each protocol in regard with the corresponding parameter metric. The definition "Yes" means support, "Limited" means not really support and "No" means does not support. For example, with regard to energy awareness, 17 out of 23 protocols consider energy awareness on their QoS routing decisions. It shows that around 75 percent of all QoS routing protocols have been reviewed here, aim to satisfy energy efficiency in their QoS services. However, different performance has occurred to different metrics in regard of service differentiation and flexibility aspects.

The table shows that most protocols do not provide service differentiation in their routing approaches and only 20% have limited support for enabling the QoS differentiation according to real-time or multimedia data classes. Further, this figure affects the

No	QoS Protocols	En_ aware	Service-dif	Multihop prob	Flexibility
1	SAR	Yes	No	No	No
2	SPEED	No	No	Limited	No
3	RAP	No	No	No	Limited
4	MCF	Yes	No	No	No
5	Energy aware Routing	Yes	Limited	No	Limited
6	MM-SPEED	No	Limited	No	Limited
7	ReInForM	No	No	Limited	No
8	Mobicast	Yes	No	No	Limited
9	DAST	Yes	No	Limited	No
10	MCMP	Yes	Limited	No	No
11	MCBR	Yes	No	No	No
12	ECMP	Yes	Limited	No	No
13	EQSR	Yes	Yes	No	Limited
14	QuEST	Yes	Yes	Yes	No
15	EE-SPEED	Yes	No	Limited	No
16	MRL-CC (Liang et al., 2009)	No	No	No	No
17	QoS multipath routing (Chen and Nasser, 2008)	Yes	No	Limited	No
18	QoS scheme (Zang et al.,2008)	Yes	No	No	No
19	Adaptable QoS scheme (Sen and Ukil, 2009)	Yes	Yes	No	Yes
20	IACR (Peng et al., 2008)	Yes	No	Limited	No
21	Multimedia QoS (Hamid et al., 2008)	Yes	Yes	No	Yes
22	M-IAR (Rahman et al., 2008)	Yes	Limited	No	Limited
23	ASAR (Sun et al., 2008)	No	Yes	No	Yes

Table 2.2: Routing protocol comparison in WSNs

performance of the protocols to accommodate WSN's applications that have different type of data traffic. In the table, with respect to the flexibility metric (defined as the metric for evaluating the performance of the protocol in accommodating many application requirements in WSNs), the reviewed QoS protocols mainly do not provide generic routing approaches that can cater the specific application needs. More than 70% of the protocols are not able to satisfy different application requirements based on differentiating QoS requirements according to heterogeneous traffic. Finally, the table shows that not much has been done by the above-reviewed QoS protocols to address the occurring problem due to the nature of multi-hop communication in WSNs such as uneven energy depletion or broadcast flooding.

From the above discussion, it is found that proposing new routing schemes that are suitable for many applications is a challenge for further research. It is because the routing protocols in WSNs are usually built based on the purpose of each application. The result of the comparison described in the Table 2.2 revealed that designing a routing protocol that can cater the needs for many applications and fit the requirements of the applications based on differentiating QoS routing in a single routing framework is a challenge that drives the direction of this thesis. For this reason, this thesis proposes a new routing scheme that can satisfy multiple QoS objectives for different application requirements according to heterogeneous data traffic, and aims to develop efficient routing scheme that addresses the routing problem caused by the nature of multi-hop communication. In here, the problems are defined as 1) the 'energy hole problem' caused by the uneven energy depletion in WSNs, 2) the 'packet detour' or 'data flooding' due to link instability and an improper area of data routing and 3) the routing inefficiency as a result of frequent changes in WSNs.

2.9 Significance of the project

In the previous sections, different routing protocols in terms of energy efficiency and data quality are discussed and reviewed in detail based on different parameter metrics of WSN routing challenges. All protocols are compared against each other using these metrics. From the discussion of the above comparison, all the techniques in energy efficient based routing aim to save energy to extend the lifetime of the networks. Further, the comparison result in data quality-based routing shows that designing new routing schemes need to cope with tradeoffs between energy savings and data quality for targeting WSN's applications. This new QoS design has to adapt to the requirements of applications with energy sensitiveness in order to satisfy multiple QoS requirements across all traffic patterns. Moreover, the result shows that the new QoS scheme needs to address the problem of uneven energy depletion or broadcast flooding that commonly occur due to the nature of multi-hop many-to-one data communication in WSNs.

Before I describe a further description of the proposed project, this thesis first identifies and studies some protocols that aim to satisfy multi-objective QoS routing as follows.

2.9.1 The multi-objective QoS Routing protocol

The protocols that use service differentiation techniques are still limited in the literature of routing protocols in WSNs and hence this leads to highlight a potential project for this thesis works. They are Multiservice Adaptable Routing Protocol (MARP) (Sen and Ukil, 2009), Path-length-based Proportional Delay Differentiation (PPDD) (Hamid et al., 2008) and Localised Multi-objectives Routing (LOCALMOR) (Djenouri and Balasingham, 2011).

(Sen and Ukil, 2009) proposes a protocol, named MARP, to create a query-based routing protocol that caters different level of application requirements. Similar to this thesis, MARP is also designed to consider four different QoS classes: the highest QoS class, the low-latency class, the reliable class and normal class. In the MARP protocol, the reliability can be achieved by introducing either path or data reliability. Path reliability is guaranteed by creating multiple alternate paths from the source to the destination node while data reliability is accomplished by duplicating the same message into multiple copies, and sending the copied packets through the alternate paths. The transmission delay in the MARP is minimised by extending the radius of transmission using more power. For ensuring the applications with higher level tasks, which require both reliability and low-latency, the protocol finds a node that has minimum distance to the destination and lower queue waiting time for relaying the original packets, while the copied packets are transmitted through the alternate paths. Hence, the protocol is designed to provide fault tolerance when a node fails to deliver packets while considering timeliness for data packets having low-latency requirements.

(Hamid et al., 2008) provides a QoS aware routing protocol that promotes high data rate for wireless multimedia sensor networks WMSNs by guaranteeing bandwidth and end-to-end delay requirements of real-time data and throughput of non-real time data. The PPDD exploits multiple paths, multiple channels, and a packet scheduling technique to meet the bandwidth and delay requirements respectively. Figure 2.10 shows that the bandwidth and delay requirements are setup locally at each node based on the path-length and incoming traffic. A node, which is far away from the destination (in this case: BS) will adjust its bandwidth lower than a node having closer distance to BS. In contrast, the protocol allows higher delay for the node, which has smaller (hop) to the destination node. The reason is the data delivery from many source nodes in the network will converge to the single destination (BS). Thus the nodes, which have locations around the destination, have to relay more packets than the farther nodes.

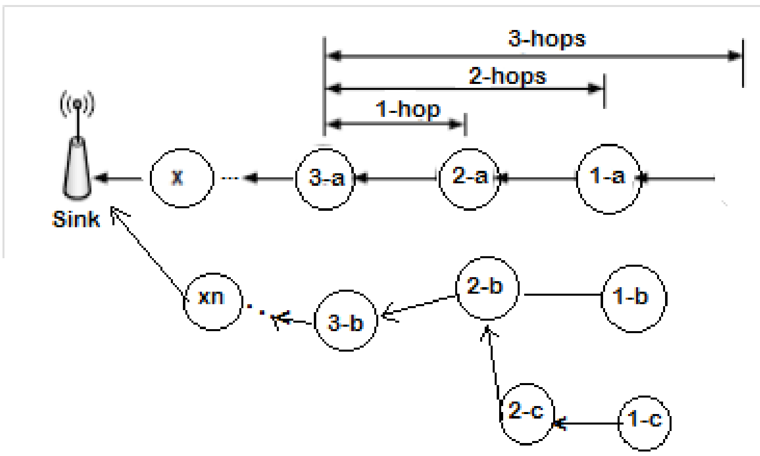

Figure 2.10: The adjustment of bandwidth and delay requirements in PPDD redrawn from (Hamid et al., 2008)

The nodes in PPDD are homogeneous multimedia sensor nodes, which are capable to perform different levels of tasks (video, audio, scalar data) and equipped with radio that has multi-channels. The node also uses multi queuing priority that assigns different priority for different type of packets as depicted in Figure 2.10 according to the delay and the bandwidth requirements for each type of traffic.

Localised Multi-objectives Routing (LOCALMOR) (Djenouri and Balasingham, 2011) designed a localised QoS routing that aims to cater the applications with respect to QoS requirements. LOCALMOR enables differentiating routing according to the traffic types, and attempts to fulfil the required data-related QoS metrics while considering energy sensitiveness. The protocol employs modules: energy module, reliability module, delay module, queuing module and neighbour manager modules that are responsible for ensuring the required QoS. The queuing module assigns different priorities for data transmission according to the required QoS of the type of packet. The energy module helps the protocol to find a next node, which has energy attribute satisfying the optimum power efficiency. The power efficiency is achieved by using the min-max approach with respect to both power transmission and residual energy. A best candidate for the next node is the node that has minimum deviation with respect to the two criteria as calculated as

$$\{Nextnode = min\ \frac{E_{vb}-E_{vj}}{E_{vb}}\ max\ \frac{\beta(dist_{vi-vj})-(dist_{vi-vt})}{\beta(dist_{vi-vt})}\ \}(2.1)$$

The following parameters are explained as follows. E_{Vb} is an energy of the neighbour node that has maximum residual energy, E_{Vj} is the residual energy of the candidate next node, vt is the neighbour node with minimum power transmission. The latency (delay) module estimates the packet velocity by defining two velocities of packet transmission. They are the required velocity, V_{req}, and the velocity offered by node, v_j denoted by V_{Vj}. The required velocity is proportional to the distance and the time remaining to the deadline.

The reliability module aims to increase the reliability by creating a single path to multi destinations. This multi-destinations single-path approach avoids data transmission using multipath to single destination in order for reducing traffic convergence around the destination node, and thus decreasing packet collisions. The protocol selects a candidate next node for each data transmission that offers the highest reliability, which is provided by the neighbour manager module.

The neighbour manager module is responsible for providing information about node' attributes regarding position, residual energy and estimated waiting time for each packet queue, and implementing estimation methods to measure the estimated latency, the packet delivery ratio and the required power transmission to and from the neighbour nodes. The measurement of reliability and latency estimation to the candidate next node is conducted using an Exponential Weighted Moving Average (Woo and Culler, 2003) estimation method, which is more suitable for WSNs and uses less resources in memory and computation compared to the other estimation methods. This module periodically exchanges information between nodes to provide relevant update about the information of the nodes for executing routing strategy according to the QoS packet class.

All MARP, PPDD and LOCALMOR suffer from the expense of data duplication and energy consumption for creating multi routes from the origin to the destination (BS). Even the multipath approach increases reliability, the transmission of the copied packets from the same messages through the alternate paths also increases the number of traffic injected in the network, and thus resulting in congestion that affects the performance of the delay of packet transmission. Moreover, this data duplication may increase the communication overheads and waste the energy consumption and hence this will reduce the lifetime of the network thereby degrading the network performance.

The other disadvantage of the PPDD and the LOCALMOR are: (i) The transmission of real-time packets with higher power for longer range of transmission to reduce delay, can deplete the energy of the nodes quickly, and thus it affects the network lifetime, (ii) the process for the PPDD to use asymmetric channel for both data transmission and data reception needs the PPDD to perform frequency switching between nodes and this incurs longer delay moreover when the network is large, (iii) both PPDD and LOCALMOR do not provide the information of the link quality of the neighbour nodes to the nodes that are closer to the BS and hence it reduces the accuracy for the estimation of end-to-end delay transmission.

2.10 The proposed scheme

This thesis argues the need to provide a customised multi-objective QoS routing that satisfies multiple QoS options based on differentiating QoS requirements according to the data type. The proposed routing scheme should enable a service differentiation mechanism to capture the requirements of heterogeneous traffic and to factor in different application requirements in a generic way. In doing so, there are some design challenges that need to be considered as follows. The proposed project should be able to (i) differentiate the required data-related to each type of traffic; (ii) incorporate the most important attributes of QoS at the node which include availability of energy, reliability, latency and other communication parameters for addressing the constraints in WSNs; and iii) enable adaptation for route differentiation based on the type of traffic by selecting a suitable node for a certain data classification. With regard to the four important parameters described above, the following assumptions is suggested to make the differences between the proposed scheme and the existing multi-objective QoS routing protocols discussed in Section 2.9.1.

First, unlike LOCALMOR (Djenouri and Balasingham, 2011), the proposed project does not employ a multi-sink single path approach to ensure reliability for certain applications because multipath routes can increase both energy consumption and communication overheads. In contrast, the proposed project use different approach to provide different QoS paths for different data requirements based on differentiation QoS requirements according to the data type. The approach may use a method that considers the quality of each neighbour matching the requirements of each data category. This quality represents the performance of each node to meet the requirements of different QoS levels according to reliability, latency, energy efficiency or hybrid. The new method evaluates the neighbour nodes and selects the appropriate nodes for data forwarding thus it can maintain the continuity of data transmission for each type of traffic. The new proposed approach aims to reduce i) information flow, ii) processing time and iii) queuing time as well as buffer storage. By avoiding data reception from multiple nodes, it decreases number of control packets thereby reducing the unexpected overheads.

Second, according to the flexibility and ease of use for the operation and the implementation of routing in WSNs, the proposed project uses a flat routing rather than hierarchical routing. Even flat routing presents collision overhead and unguaranteed fairness, it can offer an optimal routing with additional complexity (Al-Karaki and Kamal, 2004). Moreover, the flat routing does not need to perform synchronisation for the route setup and the maintenance of the network structure because the communication link for routing is formed on the fly during transmission. Thus, flat routing performs the best in terms of topology management compared to the hierarchical routing. Further, the flat routing can localise any node failure in the network thus it makes the rest of the network unaffected. When the topology of the network suddenly changes due to node failure, it can adapt quickly to find other paths by selecting new nodes that suit the requirements of data traffic.

From the discussion above, a flat routing is highly chosen for the proposed project. It is because despite its disadvantages, a flat routing has more chances to be improved in some aspects such as energy dissipation, latency, etc. Moreover, there are some provisions to make the flat routing performs even better.

To enhance the benefits of the proposed project, I develop the extended routing approaches to adapt with the changes of the network dynamics including the inherent characteristics of data transmission in WSNs. The first extended routing approach contributes to solve the uneven energy depletion due to the intrinsic many-to-one data communication model for WSNs by exploiting a non-uniform transmission range strategy. Hence, this approach aims to maintain the continuity of data delivery even the network is congested or not. The second extended approach is a new technique that can control the region boundary for packet routing in order to achieve route optimisation and reduce communication overheads caused by the undirected packet transmission or further data flooding storm. Further, to improve the benefit of the two extended approaches I propose a situation-aware adaptation routing approach that provides a gradual and an efficient adaptation of routing thereby providing a higher cost-efficient adaptation in WSN's data communication protocol.

From the discussion above, to justify the need for the proposed scheme to extend its performance in addressing the occurring problems due to the nature of multi-hop and many-to-one routing in WSNs, the following describes the protocols that aim to mitigate the problem of energy hole due to uneven energy depletion in WSNs.

2.10.1 Existing energy hole avoidance approach

This sub-section discusses the established approaches in addressing the occurrence of the energy hole problem around the BS as data transmission is converged at the BS. The existing schemes can be classified into two approaches for designing energy hole avoidance techniques. They are: i) Assistant Approach and ii) Non-uniform node distribution. The following sections describe the existing algorithms in each category.

Assistant approach

In this section, I discuss two energy conservation approaches such as deployment assistance and traffic compression and aggregation for alleviating the energy hole problem as explained previously. Examples of this type of algorithms are found in (Li and Mohapatra, 2007; Ye et al., 2003; Tilak et al., 2002; Ahmed et al., 2001).

(Li and Mohapatra, 2007) used deployment assistance in order to overcome the energy hole problem. In practise, (Li and Mohapatra, 2007) assumed that the network is divided into small sub-regions by creating a two-tier architecture as proposed in an existing literature for energy conservation (Ye et al., 2002). This two-tier architecture as depicted in Figure 2.11 includes one BS and two classes of nodes, i.e. sensing nodes (the white circle) and assisting mobile nodes (the red circle). The assisting nodes are assumed to have much higher energy and larger transmission range than the normal ones, which use limited battery capacities. An assisting sensor is deployed in order to help the sensing sensors perform data transmission. Data from the sensing nodes are gathered and aggregated at an assisting node before delivering them to the BS. Hence, the nodes that are located around the BS do not suffer from the responsibility for relaying higher traffic that converge to the BS. This protects the BS disconnected to the network because the nodes closest to the BS are kept alive.

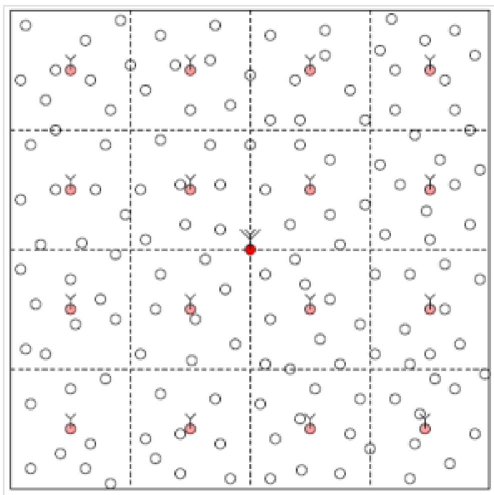

Figure 2.11: The assistant mobile sensors deployment in two-tier grid redrawn from (Mohapatra et al., 2003)

In this work, (Li and Mohapatra, 2007) also uses a wavelet based approach for time series compression and traffic aggregation to lessen the energy hole problem (Chen et al., 2004). The main advantage of this approach is to provide assisting mobile nodes, which have capability to move around and collect data from sensor nodes in a grid. The assisting nodes also perform an aggregation and compression technique for data redundancy to control on what and how sensing data is collected and communicated to the BS.

The other protocols that perform assistant approaches are (Ye et al., 2003; Tilak et al., 2002; Ahmed et al., 2001). (Ahmed et al., 2001) proposed a range extension network that deployed some mobile assisting gateways in order to provide better connectivity and facilitate scalability for the disjoint path in WSNs. Based on some linear optimisation problem, they suggested method for determining the optimal positions where the gateways should be placed. In a more recent work, (Ye et al., 2003) proposed an algorithm that controls the positions and trajectories of some reliable sensor nodes deployment in order to increase multiple paths from the source to destination for achieving better reliability in ad hoc routing protocols. Further, (Ye et al., 2003) explored the requirements for the infrastructure trade-off for improving the network optimisation that provides effective procedure for applications with respect to sensor network deployment.

Non-uniform node distribution approach

These types of energy hole avoidance algorithms consider the careful deployment of sensors and sinks and provide the non-uniform sensor node strategy approach for the improvement of energy efficiency and data capacity at the BS (Wu et al., 2008; Liu et al., 2007; Olariu and Stojmenovic, 2006; Lian et al., 2006; Jarry et al., 2006). The reason of the introduction of the non-uniform sensor distribution algorithm is based on the result of the established works describing around 90 percent of the total energy left unused. Thus, this situation potentially decreases the data capacity at the BS due to the static sink deployment and uniform sensor node distribution.

To increase the utilisation of energy resources in sensor networks, (Lian et al., 2006) provided a non-uniform sensor distribution strategy with mobile sinks in the network. This strategy allocates more energy to the sensors around the sink or deploying more nodes in the area that is closer to the sink. (Lian et al., 2006) partitioned the network into sub-areas and the sub-areas are further divided into sub-regions. The protocol put more sensors in i) the larger sub-area and ii) the sub-region around the sink. This strategy aims to ensure that most sensors will use their energy equally so they will be died almost at the same time. The simulation demonstrated that this non-uniform distribution approach can achieve much higher network data capacity compared to the uniform approach in a large and a dense network.

(Jarry et al., 2006) considered the mixed non-uniform strategy routing scheme in WSNs. The scheme aims at data distribution and adaptive data communication for power balancing in the sensor network. According to this scheme, each sensor node is allowed to either forward its data to one of its intermediate nodes or relays data packets directly to the sink by considering the remaining energy of nodes for deciding the transmission decision. They provided a simple linear programming analysis for data communication and proved that an energy-balanced mixed and non-uniform routing strategy scheme could achieve better performance compared to any other possible routing strategy. By proposing the gradient-based routing scheme and allowing the nodes to send data directly to the sink, it can reduce delay and maximise the lifetime of the network thereby improving the data capacity at the BS.

Most techniques for understanding and investigating energy hotspot problem is corona model, such partitioning disks into disjoint concentric sets as depicted in Figure 2.12. The corona model is used to analyse the relationship between the energy spent on routing, which affects the network lifetime and the width of each corona in concentric corona model. (Olariu and Stojmenovic, 2006) assume that the network is divided into concentric corona models, in which each corona consists of several nodes having certain radius of transmission range, R. The size of each corona can be designed either different or same as well as the radius of transmission range of nodes in each corona. The size of corona potentially affects the energy consumption of each node in transmitting data to the next node, which is located in other corona closest to the BS. If I set that each node can only deliver data from corona i to corona i-1, but not corona i-2 (closer to the BS), thus the nodes at corona i with locations closer to corona i-2 cannot directly transfer data to the next-nodes in the corona i-2 even their radius of transmission range can reach until the area of corona i-2. This wastes transmission energy and increases delay for data delivery because data has to be propagated with more hops to arrive in the BS.

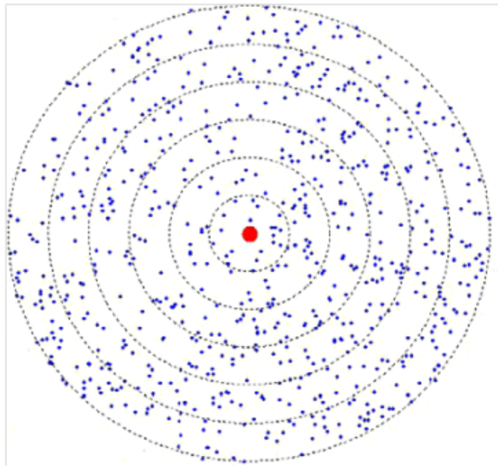

Figure 2.12: A Concentric Corona Model
redrawn from (Olariu and Stojmenovic, 2006)

(Liu et al., 2007) investigated the sink routing-hole problem as the results of the random and uniform node deployment in a circular corona model where the sink is placed at the center and the data transmission from the nodes are converged at the sink. Accordingly, they provided a non-uniform, power-aware deployment approach in order to optimise the continuous connectivity-coverage. They demonstrated the results that power-aware deployment can significantly lengthen the network lifetime and improve the data gathering. However, this scheme has challenged to consider different positions for the location of various number of sinks in different network scenarios. Thus, the stability of the algorithm in proposing the non-uniform power transmission can be evaluated in order to provide the prolonging network connectivity.

(Wu et al., 2008) used the non-uniform node distribution strategy to explore and solve the energy hole problem in WSNs. They modeled the network with a circular multi-hop sensor network like concentric corona with non-uniform node distribution in order to avoid the situation where most nodes in the inner coronas used up their energy quickly. Based on the observation, they increase the number of nodes to the coronas that are closer to the sink, except the outermost corona since the outermost corona only needs to forward data generated by themselves. This non-uniform node distribution in geometric progression of coronas aims to achieve sub-balanced energy depletion of the whole network because the protocol still be able to find nodes around the sink for relaying packets to the destination. To support the non-uniform distribution strategy, they presented a q-Switch Routing to find distributed the shortest path routing algorithm. The simulation resulted that the scheme achieved higher energy efficiency, and less than 15% energy is wasted thus improving the network lifetime, the residual energy ratio and the throughput.

Location-based routing for reducing packet flooding or detouring

The characteristic of WSNs in terms of the non-uniform node distribution and the link instability in WSN's data communication leads to packet detouring or further data flooding storm that causes a high possibility of congestion, a low packet delivery ratio, and high energy consumption. Most significant research efforts in WSN aim at energy management technique that use the geographic information of the nodes to route data by reducing the communication cost while achieving data quality (Al-Karaki and Kamal, 2004). Following several approaches have been described on how the information of location can be used (Stojmenovic, 2002).

The first approach is based on typical protocols that use the location information of source, neighbours and destination node to forward packets in greedy mode. These protocols are Greedy Perimeter Stateless Routing (GPSR) (Karp and Kung, 2000), Geographic and Energy Aware Routing (GEAR) (Yu et al., 2001) and Greedy Other Adaptive Face Routing (GOAFR) (Kuhn et al., 2003). GOAFR combines a greedy and a face routing technique to restrict routing discovery and to select a node which has the closest distance to the sender node. GOAFR is suitable to the high-density network but it takes more number of looping that causes longer time during routing process to route packets until destination. Similar to GOAFR, GPSR constructs a perimeter forwarding approach by transmitting packets through nodes at the edges of the graph around the destination when no closer neighbours are available. The extended mechanism to GPSR is GEAR (Yu et al., 2001). GEAR divides the routing region into four zones and it uses both the remaining energy and the geographic information of neighbouring nodes to estimate the link cost for a path. GEAR selects a node having closer distance to the destination and it uses the previous path if no closer neighbour to destination is available.

The second approach is based on the protocols that divide the routing region into smaller rectangle zone (grid) such as Geographic Adaptive Fidelity (GAF) (Xu et al., 2001) and Grid (Liao et al., 2001). Based on node's location and radius transmission, GAF forms a small virtual grid. The side of the grid is defined by $L = \frac{R}{\sqrt{5}}$,

Nodes inside the grid communicate with each other and periodically select a cluster head that actively monitors and reports to destination about any events while other nodes go to sleep. Hence, GAF may achieve energy conservation to lengthen the network lifetime. GRID uses grid shape and performs a grid leader to route packet in a grid-by-grid manner. GRID is suitable for a large and dense network as this protocol divides the network into smaller region (grid). GRID also can maintain a route alive when a destination node moves out of region by performing a 'handoff' operation similar to a cellular system. The last approach is the algorithm that uses the location-based information to limit the area of packet routing to achieve route optimisation. This protocol includes location-aided routing protocol (LAR) (Ko and Vaidya, 2000), DREAM (Basagni et al., 1998), and WSNHA-LBAR (Xiao Hui et al., 2011). LAR and DREAM try to reduce routing overhead by limiting the route discovery area in a smaller request zone. This zone can be improved by adapting the shape based on the speed of mobile node. They represent three shapes: rectangle, bar and fan. In (Xiao Hui et al., 2011), the protocol defines a cylindrical request zone for route discovery and maintenance as well as packet forwarding. A node can adaptively adjust the radius of cylindrical zone using a self-learning Bayesian's theorem according to the prior probability of successful and unsuccessful data transmission. The transmission fails when a source node does not receive a route reply packet from a destination node or otherwise. Each node creates a table containing both number of successful and fail packet transmission under different radius zone and chooses a radius which has the higher probability for packet forwarding.

2.11 Summary

In this chapter, different routing protocols in terms of energy efficiency and data quality (QoS based) are identified and discussed in detail. This chapter discusses different performance metrics of WSN routing, and uses the metrics to compare and evaluate the protocols against each other to find a gap that drives the direction of the thesis. The discussion argues that one of the challenges in WSNs is to design a QoS differentiating routing scheme that improves energy efficiency and satisfy multiple QoS demands for applications. And each application has various traffic types according to the requirements of different QoS levels. Moreover, to extend the performance of the proposed QoS differentiating routing for coping the routing challenges due to the typical many-to-one multi-hop data communication in WSNs, I propose some new approaches to leverage the full potential of routing adaptation and perform a holistic QoS routing framework for WSNs. In the next chapter, a new QoS differentiating routing is proposed.

Chapter 3

The Proposed Multi-Objective QoS Routing

In the preceding chapter, I presented a comprehensive review of the QoS routing for heterogeneous traffic classes, and discussed the issues and challenges. This chapter describes the Throughput-delay Guaranteed Routing for Reliability (TeGaR) algorithm that provides traffic differentiation for different data requirements. TeGaR routing fulfills multiple QoS demands for applications that generate traffic of various classes.

3.1 Introduction

In the previous chapters, it is argued that the need of the new QoS routing protocols for catering different application requirements is required in order to meet one of the challenges in Wireless Sensor Networks (WSNs). In designing a QoS routing scheme that improves energy efficiency and satisfies multiple QoS demands for applications that has various traffic classes. As described in Chapter 2, guaranteeing QoS in these networks is difficult and more challenging due to the fact that the available resources of sensors and the various applications running over these networks have different constraints in their nature and requirements (Djenouri and Balasingham, 2011). The previous chapter discussed that current studies on QoS routing scheme mostly focus on limited number of aspects of QoS (Al-Karaki and Kamal, 2004).

The aim of this chapter is to design and examine a new QoS differentiation routing based Fitness Function that supports heterogeneous traffic in terms of data-related QoS and meets the demands of each application in WSNs while it achieves cost-efficiency. According to the aim, the proposed routing scheme provides customised QoS metrics for each type of data traffic according to many applications in WSNs by differentiating QoS requirements. The proposed scheme encompasses multiple nodes attributes in a single QoS-based routing approach to enable route differentiation due to each type of traffic by using a, Fitness Function Method (Khanna et al., 2010). This Fitness evaluates the quality of nodes that suits the requirements of data traffic to find a best node for a certain data classification. The quality of nodes' include energy reserve, data delivery ratio, distance to the base station (BS), error rate in both channel access and buffer storage. Therefore, the scheme may fulfill the required data-related QoS metric(s) while being aware of energy efficiency. The scheme uses local neighbour information as well as geographical information. This eliminates the need for propagating route information and reduces both energy consumption and the time for creating the path. The scheme is modular following the traffic classification thereby the process of adaptation can be easily implemented. The proposed scheme tailors adaptation of route selection to suit application specific needs while considering reliability, latency, residual energy and error rate.

To facilitate route adaptation for differentiating QoS routing according to the data type, the proposed scheme has the following components:

1. traffic classification, which enables prioritisation of data traffic based on the importance of different application requirements.

2. fitness function based Genetic Algorithm that considers the quality of each node for route selection according to the data traffic and applications' requirements.

Section 3.2 identifies the requirements for improved multi-objectives QoS routing algorithms. Section 3.3 describes the service/traffic differentiation techniques which would be used for the proposed QoS routing scheme in supporting heterogeneous traffic in WSNs. In Section 3.3, I describe the proposed scheme in detail. Experimental results and analysis are presented in Section 3.4 to demonstrate the performance of the proposed scheme. The summary of this chapter is concluded in Section 3.5.

3.2 Requirements for Improved Multi-Objective QoS Routing

Reviewing the literature shows that there are a number of differentiating routing for multi-objective QoS (MQO) routing approaches that have been implemented for WSNs (Djenouri and Balasingham, 2011; Sen and Ukil, 2009; Peng et al., 2008). To satisfy QoS option in terms of reliability, Djenouri and Balasingham (2011) provide multiple paths to deliver packets to a destination thus ensuring both path and data reliability. The fault tolerance can be increased by maintaining multiple paths between the source and the destination at the expense of an increased energy consumption, traffic generation and overhead of maintaining the alternate paths. In addition, the work in (Hamid et al., 2008) provides a multi-channel access for packet transmission according to the data type in order to guarantee channel access for the data traffic requiring reliable transmission. This multi channel approach can eliminate collision when the traffic becomes more congested, but it incurs additional overhead for performing channel switching between nodes, and the delay takes longer when the number of nodes increases significantly.

The following lists the requirements for the MQO algorithms under the networks that pose resource constraints.

- With respect to energy constraint in WSNs, the MQO algorithm should be aiming to extend the lifetime of the network by (i) avoiding approaches that causes energy inefficiency even the approaches improve other QoS metrics such as reliability or latency, (ii) distribute energy consumption evenly in the network;

- To achieve multiple levels of QoS requirements and support heterogeneous traffic, the MQO approach should have compatibility to specific behaviours of applications. The approach should be able to differentiate the data-related requirement of each type of traffic and provide a method that enables an efficient route selection based on the data classification;

- The design of MQO algorithm has to be fully distributed in each node in order to avoid centralised algorithms that require additional global synchronisation overheads, and are not practicable to large sensor networks;

- The MQO algorithm should be able to perform self-configuration to maintain the system reliability so that it can adapt to any topological change in the constrained energy environments.

To address the above-mentioned issues and requirements, this thesis proposes a route differentiation based adaptation approach that takes into consideration the sensor's different attributes. The proposed scheme provides an efficient adaptation routing approach for differentiating QoS requirements based on the applications specific needs.

The following presents both the theoretical foundation and mathematical analysis of the proposed QoS differentiating routing based Fitness scheme, termed Throughput-delay Guaranteed Routing for Reliability (TeGaR) for addressing the need of MQO satisfaction.

3.3 The QoS Differentiating Routing based Fitness Scheme

A new localised QoS routing protocol for WSNs is proposed in this section. The proposed QoS differentiating routing based Fitness scheme targets WSNs' applications having different data requirements. It provides an optimal path for different QoS services to fulfil the required data-related to each type of traffic while considering latency, reliability, and energy efficiency.

To differentiate data based on the applications specific needs, a traffic classification model is used. The proposed traffic classification model attempts to provide a wide range of classes of traffic differentiation to support most of the application scenarios for WSNs as described in Chapter 2. Considering route adaptation, the proposed scheme finds a best node for a certain data classification by employing a Fitness Function (Khanna et al., 2010), a method for evaluating the quality of nodes. To ensure the required QoS for different data types, the proposed scheme provides a priority based delivery approach, a typical class-based queuing model that classifies packet based on its priority and importance. The following subsections discuss the proposed routing scheme and the operations of its components including traffic classification, route selection using Fitness Function and priority based delivery as the core contribution of this chapter.

3.3.1 An overview of the proposed scheme

As I mentioned earlier, differentiating QoS requirements needs to know the specific behaviours of application services (i.e. support heterogeneous sets of sensors purposes). For instance, in a military environment such as a battlefield, the sensors have to capture and send the imaging-sensory data of the detected target to the user under time-constraints. To take a proper action, both sensors and controllers should deal with real-time multimedia data, which requires the sensors to guarantee the reliable delivery with possible minimum delay. Further, for a continuous monitoring application such as habitat surveillance, sensors may not need to support real-time data transmission.

An effective adaptation approach for route differentiation should address the different data related requirements of application(s). It has to consider multiple attributes of sensors in order to improve energy lifetime, reduce delay to meet the demands of real time traffic while maintaining throughput for non real time applications, and optimise network utilisation.

The proposed scheme uses a Fitness Function (Khanna et al., 2010) that is derived from the genetic algorithm (Goldberg, 1989) to evaluate the qualifications of each chromosome (in this case a sensor node) and find an optimal solution for improving the adaptation according to data classification. This fitness function has the capability to evaluate the current condition of node's attributes, including resource levels. This enables the proposed scheme to provide a customised QoS path for each traffic category according to different applications' requirements while improving energy efficiency. Since the Fitness Function is derived from the genetic algorithm, it compares the fitness values that are the result of the sensor node evaluation, and only the node with a high fitness value (Kreinovich et al., 1993) matching the specific data type is selected. This node will be considered as the superior (suitable) data-forwarding node for that certain type of traffic.

To achieve the aforementioned requirements as discussed in Section 3.2, I have proposed four principal modules for the proposed QoS differentiating routing scheme. These modules include:

- Application Module
- Neighbour (Localised) Module
- Routing Module
- Queuing Module

Here I describe the purpose and functionalities of these modules. Figure 3.1 shows an overview of the TeGaR scheme and its main modules.

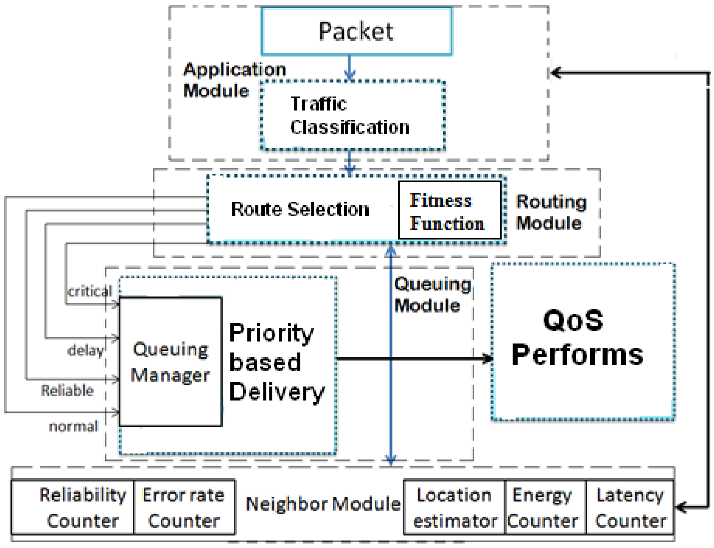

Figure 3.1: An overview of the TeGaR scheme

- Application Module. The application module is responsible for differentiating type of incoming packets into data packet or control packet. In this module, different classes of QoS requirements are determined including: 1) critical QoS, 2) real time QoS, 3) reliability QoS, and 4) normal/best effort QoS. These requirements can be viewed as application-related QoS metric that must be considered for all data types. With these QoS requirements, data traffic will be classified into several categories following the requirements. The detail description of different data classes will be described in the next subsection. In addition, this module also manages the operation of the control packet including Hello packet, Acknowledge packet and Sink packet. The control packet is broadcast to neighbour nodes in order for a node updating the states of its neighbours for local information of routing decision. However, if the delivered packet is a data packet, the application module mandates the routing module to provide a path based on the type of packet, and the queuing module to assign a priority to the packet before sending it out to the channel access. The aim of this module is to ensure the required QoS for each packet, this application module uses a traffic classification model that will be described further in the next subsection.

- Neighbour (Localised) Module. neighbour module is responsible for executing the HELLO packet that will be used to exchange information about the states/quality of nodes between neighbours. The HELLO packet here refers to the HELLO protocols that are used in these works (Zeng et al., 2009; Mahapatra et al., 2006; Felemban et al., 2006) adding some more information of nodes about number of error rate, reliability and latency as well as geographic information to the exchanged packet. At a critical condition when a node drops its energy toward the energy threshold, a critical message is broadcast to neighbours to inform them that they should not forward any packets to this node. This module is also managing a neighbour table and implementing methods for quantifying the quality of nodes that includes residual energy, success delivery ratio, nodes coordinate and history of nodes failure in relation to access channel and packet queuing, also current position of neighbours and other local parameters. The information of neighbours is stored in a table and is updated periodically upon each reception of a HELLO packet in a fined-period of time in order to provide relevant and refresh update of information without excessive energy consumption. Further, to provide better decisions, this neighbour module provides the other modules with the required information according to the type of packet. According to this module, the advantage of using a critical message and information of node's location aims to reduce energy consumption while initiating routing. The critical message avoids data forwarding transfers through the nodes which have less energy, while the location-based routing aims to use the geographic information of the nodes to route data by reducing the communication cost. Hence, this information can increase the durability of the network function.

- Routing Module. The routing module performs adaptation of route selection according to the neighbouring nodes attributes which reflects the quality of nodes. This module attempts to develop an application-centric QoS routing by determining the next node that fits to the requirements of specific application. The process of node selection is influenced by the weight resulted from the computation of the nodes attributes that represent their quality. A node with a higher weight is selected as a candidate node to deliver packets with higher importance and priority. In addition for routing to be effective, I assume that i) nodes should be stationary. This reduces to exchange Hello packets more frequently , which is time and energy consuming, and ii) the network should prevent dealing with void situations, a situation where nodes do not have neighbours in their vicinity.

- Queuing Module. This module classifies the priority of packets into real time and non real time queues. The incoming packets with delay-sensitive type are highly prioritised to be processed into the real time queue otherwise the other types of packets are processed into a non real time buffer. The consideration of using real time and non real time buffer attempts to achieve lower latency for the packets within a deadline. In addition, in both buffers the queue module employs a simple FIFO (first in first out) serve mechanism. The simple FIFO tries to avoid packets being queued up unexpectedly which results in longer delay. For a packet (m) arriving to the i^{th} queue (Q_i), the expected waiting time $E[W_m(Q_i)]$, depends on two factors, (i) the number of packets in the queue with the higher or same priority and (ii) the remaining time for sending out the current packet that is based on the operation of back-off mechanism in CSMA-CA MAC protocol.

The following subsection discusses the components of each module in order to explore the functionalities and the interactions between the modules. We first describe Traffic Classification and the way the application module differentiates each type of traffic.

3.3.2 Traffic Classification

The Traffic Classification is the component that is responsible for enabling traffic diversity that targets WSNs applications having different types of data traffic. This component is able to differentiate the required data-related to each type of traffic that is based on differentiating QoS requirements according to the data type. It provides customised QoS metrics for each data type due to the application's requirements. By implementing this traffic classification, each sensor enables to map each type of packet to certain category so that the algorithm can fulfill the required data-related QoS metrics to satisfy multiple QoS services. The traffic classification can directly affect the overall performance of the network (Iyer et al., 2001) as it ensures the required QoS for each packet. The result of the traffic classification is then be used by the Fitness Function (FF) method to enable the route selection based QoS objectives.

Unlike routing, traffic classification for improving QoS is not widely studied in WSNs (Park et al., 2006). The performance of WSNs highly depends on the lifetime of sensor nodes. For this reason, various QoS-based techniques in WSNs are mainly focused on developing energy efficient routing mechanisms that can lengthen the network lifetime. There are limited studies that consider the QoS requirements specific to each application (Al-Karaki and Kamal, 2004).

We introduce a traffic classification model based on QoS requirements as plotted in Figure 3.2. The figure shows the traffic classification process in a sensor node. The first step towards classification is to determine the class of each packet based on either the packet size or classification rules or the purpose of the built WSNs. The packet fields that are generally being monitored are the sensor's port number or the location of the node that correlates to certain types of sensing purposes.

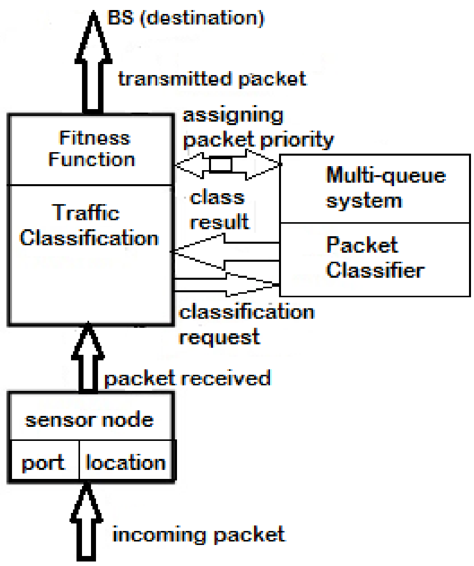

Figure 3.2: Traffic Classification Model

According to the wide range of WSN applications such as target tracking, habitat monitoring, intelligent transportation systems, and health-care applications (Ennaji and Boulmalf, 2009; Martinez et al., 2007), data traffics are classified into the following categories in order to cope the most typical application scenarios in WSNs as described in Chapter 2.

1. C1: Huge real-time traffic which requires both reliability and low delay services as it is related to the critical tasks;

2. C2: Scalar real-time traffic that delivers data packets within a given deadline for packet arrival at the BS;

3. C3: Scalar non real-time data traffic which requires delivery without specific deadline and which tolerates a reasonable loss of packets, such as continuous monitoring applications; and

4. C4: Best effort traffic that has no specific requirements.

Table 3.1 presents the traffic types that I propose for typical applications in WSNs as described above.

Parameters being Monitored	**Traffic Type**	**Location of Sensor**
Scalar data (humidity)	Non real-time	(X_i, Y_i)
Scalar data (heat/pressure)	Real-time	(X_n, Y_n)
Video/image data (object tracking)	Real-time	(X_m, Y_m)
Video continuous monitor	Non real-time	(X_z, Y_z)

Table 3.1: The type of traffic classification in WSNs

This data classification may address the need of many applications in WSNs due to the increasing growth of multiple sensing capabilities for multiple monitoring and data-gathering tasks (Gu et al., 2005; Dutta et al., 2005). These tasks have different requirements and priorities according to their QoS regarding reliability, timeliness, energy consumption and so on. For further description of the interactions amongst the modules, I describe the component for the neighbour module in the next subsection.

3.3.3 Localised Manager

The localised manager is the component that receives packets, updates existing entries of the neighbour table based on Hello packets and provides required information to the other modules for further analysis. This component manages the information of the following parameters: current position, residual energy, success delivery ratio and the history of node's failures.

- Energy. Energy is a crucial attribute of a node. Without sufficient energy, a node is not able to do activities i.e. idle, processing, receiving, and transmitting. The value of current energy will be updated every time the node changes its state of activities, and the amount of energy consumed by a node is supplied from the parameters of the MAC operation used;

- Success delivery ratio. This attribute is important to estimate the reliability of a node in terms of receiving or forwarding data to the BS via its neighbours. It is defined as the ratio of packets sent to packets received at a node (%). The maximum ratio indicates a node has higher probability to forward packet successfully;

- Nodes coordinate. The nodes coordinate is used to measure the distance from a node to the BS. By using this attribute, a node can determine the relative distance of its neighbours to the BS. The node with minimum distance is estimated to be able to reduce the delay of packet propagation to the BS. Thus, a node with minimum distance has a higher chance to forward C1 or C2 packets.

- The history of nodes failure. It explains about the frequency of failures at each node according to the channel access and packet queue based on the MAC operation. It is supposed to provide information about the probability fail of the node in the future.

The next subsection describes the route selection component for providing differentiation routing based on customised QoS requirements.

3.3.4 Route Selection using Fitness Function

The Route Selection (RS) is a core component of the proposed scheme that is responsible for differentiating routing based on QoS requirements using the Fitness Function, that is a scoring method for evaluating the quality of nodes. The RS constantly evaluates the nodes' quality with respect to QoS attributes (i.e. network layer parameters) which would be used to provide paths for different data classes to target different applications in WSNs. The following section provides a description of theoretical, though important, properties of Fitness Function.

Fitness Function: definition and properties

The Fitness Function has been applied to optimal solution problems in machine learning, robot engineering, and computing (Goldberg, 1989) by selecting fitter (better) individuals or objects as new genetic individuals for the solution. A Fitness Function is a scoring method in a genetic algorithm that evaluates the qualifications of each chromosome (in this case a sensor node) to find an optimised solution. This fitness determines the implementation of a solution in WSNs, and comprehends the principle elements for the performance of the system. The preferred solution yields another set of next generation chromosome as a fitter solution and result of the evolutionary process (Khanna et al., 2010), where only the superior chromosome affects the next generation.

Fitness Functions have also been used for solving many problems in WSNs, such as self-organisation for efficient autonomous control (Khanna et al., 2010), reliable transmission with fuzzy logic fitness function (Kim and Cho, 2007), energy-efficient clustering algorithm (Norouzi et al., 2011), and multi-objective optimisation design methods (Osadciw and Veeramachaneni, 2007).

In this research, the fitness function is used to evaluate the strategy of routing in a wider spectrum considering quality of service (QoS) of WSNs, functional distribution (coverage, scalability) as well as optimal resources allocations. Due to the limited resources of WSNs, this function aims to reduce computational resource consumption for targeting WSNs' applications having different types of data traffic based on differentiating QoS requirements. Like in (Park and Jung, 2010), I use the fitness function for evaluating the quality of the node, and use the concept of superior chromosome in the evolutionary process (Khanna et al., 2010) to define a suitable data forwarding node which has the quality matching the traffic type in order to avoid heavy traffic congestion. This enables the algorithm to perform efficient data transmission due to much less queue overflow and supports fair data transmission for all traffic types. Therefore, my proposed fitness function attempts to achieve reliable and distributive data transmission thus reducing the energy consumption across the network.

Figure 3.3 shows the structure of the table of neighbour that includes a fitness function for a specific node. Based on the information about neighbour sensor nodes, the fitness is calculated from these attributes such as node identification (ID), location information, data throughput rate, residual energy and error rate of channel access and buffer storage.

According to Figure 3.3, the value of fitness is a degree of fitness to the neighbour node, which is calculated by the Fitness Function. In my proposed scheme, the fitness

Fitness	OID (n)	NID (n)	N-att1 (n)	N-att2 (n)	....	N-att x (n)

Fitness : The result of Fitness Function Calculation

OID : Original/Source node ID (identification)

NID : Neighbor node's ID (identification)

NAtt: Neighbor node's attribute

Figure 3.3: Representation of a fitness function derived from sensor node's attributes

value is determined using the following formula:

$$(Fitness)^y = \sum_{A=0}^{n} C_i F_A = C_0 * (NAtt)_1^y + C_1 * (NAtt)_2^y + C_2 * (NAtt)_3^y + ... + C_i * (NAtt)_n^y \quad (3.1)$$

where $(\text{Fitness})^y$ denotes the fitness (quality) of a neighbour node Y, C_i is a constant due to node's attribute and $(\text{N Att})_i^y$ is the attribute (i) of a neighbour y.

By using the approach of the genetic algorithm, the values of fitness of each neighbour node are compared, and only a neighbour with a high fitness value is selected as the superior (suitable) data forwarding node for certain type of packet. One of the important properties of the fitness function is its awareness for categorising traffic classes (Park and Jung, 2010). For this reason, this chapter proposes to use the fitness function approach to provide multi-objective QoS routing based for applications that generate traffic of various classes.

The Proposed Fitness Function and Weight

Formula 3.1 describes the calculation of the fitness termed as FIT. Based on the FIT value, a best sensor node is selected for specific data type based on the above-mentioned traffic classification. The information about selected nodes for different data types is kept by each node. Based on this information, each node distributes all data types to a destination through different neighbours. A node has information about multiple neighbour nodes with different attribute values thus many solution sets can be formed. From these solution sets, a node having the highest fitness is chosen to forward packets for specific traffic category based on differentiating QoS requirements.

The method for selecting appropriate nodes (chromosomes) calculates each node's fitness regarding its quality divided by all neighbour nodes (population). The result is a value between 0 and 1. The quality of the nodes is determined based on a set of sensor attributes that can be used to identify the condition of nodes, such as residual energy, location, ratio of delivery, and so on. Route differentiation for heterogonous traffic is performed through using the Fitness Function (Rizal et al., 2012). Our proposed technique finds QoS paths according to each type of traffic requirement based on the following formula:

$$(Fitness)^y = \sum_{i=0}^{n} \omega_i f(QoS)^i \quad (3.2)$$

where ω_i presents the weight assigned to each QoS metric and QoSi denotes each metric that reflects the QoS performance. Weights are values between 0 and 1 and associated with the relative importance of each QoS metric due to each class of traffic. The Fitness Function considers five QoS attributes (metrics) as follows.

$$(Fitness)^y = \omega_0 f(E_y)+\omega_1 f(DR_y)+\omega_2 f(dist_{y-BS})+\omega_3 f(Err_y)+\omega_4 f(Load_y) \quad (3.3)$$

where f(E_y) is the factor that reflects the remaining energy based on the battery lifetime. Higher energy on a node makes the node more likely to be chosen. f(DR_y) is used for finding the ratio of packets successfully delivered compared to the received packets in a node. This function reflects the reliability of a node to route packets. This function favours nodes having a higher delivery ratio to be used for routing. f($dist_{y-BS}$) is the factor of distance between a node and BS that estimates end-to-end delay transmission. The closer a node is to the destination, the lower hop delay prediction is. This makes the node more attractive to forward real time packets. f(Err_y) is used to evaluate the link availability of a candidate next-node. Higher error rate is indicated by many channel access failures and it degrades the possibility of a node to be selected. f($load_y$) is a factor that reflects the number of buffer overflows that occur in a queue. This factor makes heavily-used nodes less attractive even if the node has a lot of energy. Lastly, ω_{0-4} are weighting factors.

Table 3.2 shows that each class of traffic is assigned different weight due to different levels of importance of each metric. The total weighting factors is 1.

Weight	**C1**	**C2**	**C3**	**C4**
$\omega_0 f(E_y)$	High	Moderate	High	Equal
$\omega_1 f(DR_y)$	High	Moderate	High	Equal
$\omega_2 f(dist_{y-BS})$	High	High	Low	Equal
$\omega_3 f(Err_y)$	Moderate	Low	Low	Equal
$\omega_4 f(Load_y)$	Moderate	Low	Moderate	Equal

Table 3.2: Weight and Importance Level

According to Table 3.2, I can define different weights to each metric to represent the contribution levels of the metrics due to the requirements of each type of traffic class. This enables the fitness approach to perform route differentiation that satisfies multiple QoS paths for all traffic classes.

Implementing Fitness Function in the routing algorithm

The proposed routing algorithm requires each node to maintain information about its neighbours and uses geographic information (nodes' locations) to find paths. Figure 3.5 presents the pseudo code for the proposed routing scheme.

According to Figure 3.4, each sensor is assumed to be static and knows its geographic (coordinate) information using some localisation methods. At a certain period, each node regularly exchanges information about the nodes' attributes with neighbour nodes and keeps the neighbour nodes' states in a table to create paths later, as described in line 4-10 in Figure 3.4.

At the beginning, based on the localised routing protocol, every node X does not have a path to a destination D. A node X only knows its neighbour nodes' attributes that are maintained in a table. When a source node S has data and wants to deliver data to a destination (in this case BS), the node S classifies the sensed data according to traffic classification as defined in Section 3.3.1, and initiates a route discovery process by sending a Route Request (RREQ) packet to the node N that has the quality that suits the required-data routing. Similarly, the node N, upon receiving the RREQ, will rebroadcast the RREQ again to the next-corresponding neighbour M if the node N does not know a route to destination BS. When the RREQ reaches a node that has a route to BS (which may be the destination BS itself), a Route Reply (RREP) packet is sent

Seq-line	**Programming Command**
1	Initialisation node v_i
2	Find neighbours by using sink message protocol
3	Determine the type of neighbours based on their location (hop away from BS): N $^{fwd}_{hop<hop\longleftrightarrow v_i}$; $N^{eq}_{hop=hop\longleftrightarrow v_i}$
4	If the time ≈ the exchange message period then
5	Compute the attributes of the node v_i
6	Sensing the channel (link) using CSMA-CA protocol
7	If channel (link) is free then
8	Run Hello exchange message and
9	Store the receiving Hello messages in the table of neighbour
10	Complete Hello exchange
11	else
12	Wait for backoff (t$_{BOT}$) until the channel is not busy
13	else
14	Run routing and go to line 12
15	Initialisation paths to BS.....
16	If no paths then
17	Initiate route discovery by exchanging RREQ-RREP messages
18	else
19	Initialisation: [buffered packet.type] ∈ type[traffic classification]
20	else
21	rejected the buffered packet
22	If the sending packet ($\acute{ə}_i$) ∈ C1,C2 then
23	Select next-node ($\grave{N}_i$) ∈ N $^{fwd}_{hop<hop\longleftrightarrow v_i}$ and
24	Set the weight of the sending packet according to the data type
25	Calculate the fitness of neighbour v_j, v_j ∈ N $^{fwd}_{hop<hop\longleftrightarrow v_i}$
26	If the number of neighbours of the N $^{fwd}_{hop<hop\longleftrightarrow v_i}$ > 1 then
27	Find the highest (Fitness value) for ($\grave{N}_i$)
28	break
29	else
30	If the number of neighbours of the N $^{fwd}_{hop<hop\longleftrightarrow v_i}$ ≤ 1 then
31	Choose v_j, $v_j \subset$ for ($\grave{N}_i$) for both C_1 and C_2 else
32	If the number of neighbours of the N $^{fwd}_{hop<hop\longleftrightarrow v_i}$=0 then
33	Find the highest fitness value from ($\grave{N}_i$),where ($\grave{N}_i$) $\in N^{fwd}_{hop=hop\longleftrightarrow v_i}$
34	break;
35	else
36	Select next-node ($\grave{N}_i$) for C_3 and C_4, ($\grave{N}_i$) $\in N^{eq}_{hop=hop\longleftrightarrow v_i}$
37	If the type of sending packet for C_3 then
38	Find the highest fitness value
39	else
40	the lower value is for C_4
41	End

Figure 3.4: Pseudo code for the proposed routing scheme

back to S. Currently S has a route to BS and uses this routing information to send data to BS through intermediate nodes that are suitable to the flowing traffic. This process is described in the lines between 15 and 17 in Figure 3.4, and reduces communication costs since a node S does not have to maintain a table containing all nodes which have paths to BS.

In WSNs, BS periodically updates its information about the locations of the nodes by broadcasting a sink message over the network. According to the line 2 in Figure 3.4, upon receiving the sink message from BS, a node X updates its relative location to BS as well as its hop value with the value of the sink plus 1 (+ 1) if its current hop value is lower than the hop value of the original sink message, otherwise X still keeps its current value. The new value (location and hop) is then sent back to BS in order to establish a distance between a node and the BS. Following this approach, every node becomes aware of its neighbours' locations (refer to line 3 in Figure 3.4).

The proposed routing scheme utilises a carrier sense multiple access collision avoidance (CSMA-CA) (Kleinrock and Tobagi, 1975) mechanism, which is used for sensing the communication link before initiating packet transmission as shown in line 7 in Figure 3.4. CSMA-CA limits the number of packet collisions when nearby devices transmit at the same time. To avoid packet collision and to minimise packet retransmission that costs energy, the MAC (Medium Access Control) protocol, uses collision avoidance rather than a collision detection technique, which senses the channel to ensure the channel availability for packet transmission. When a channel is busy (described in lines between 11 and 12 in Figure 3.4), a node performs a back-off approach (wait for a certain period) until a node finds an idle channel. Once the channel is free, the source/sender node X sends a Request to Send frame (RTS) to the intended destination node Y. If Y is ready to receive data, Y replies with a Clear to Send frame (CTS) to X. After receiving CTS, X sends data and Y responds with an acknowledgment frame (ACK) to indicate successful reception. If node X has more data (packets), X has to wait (back-off) a random time after each successful data transfer and competes with adjacent nodes to use the channel (medium).

During the RTS-CTS handshake mechanism, any node Z within the transmission range of either node X or Y can overhear the RTS or CTS frames. By hearing a RTS frame, node Z should halt sending any message until a CTS is received. At the same time when another node A overhears a CTS frame from Y, node A refrains from sending anything for a certain amount of time until the data and ACK frame are received. This mechanism helps to avoid hidden terminal problems in the MAC layer protocol (Kleinrock and Tobagi, 1975) and to estimate the time for data transmission as these frames include information about the length of data.

In this work, the protocol uses information about the time and energy required for every nodes activities based on some parameters (Kohvakka et al., 2006) to represent low-power and low-rate sensor nodes. When a packet arrives in the incoming queue at a node, the node checks the type of the incoming packet and sends the packet to the appropriate queue according to traffic classification as defined in Section 3.3.1. The node then finds the group of neighbours with respect to the type of packet as plotted in Figure 3.4 (lines between 22 and 27 as well as lines 36 and 37). The algorithm calculates the fitness for each node based on the Fitness Function defined in Section 3.3.2. Then, for each node the highest fitness for certain data class is determined by running the Fitness Function algorithm in line 27. In line 24, the appropriate weights according to packet type are determined for the next-node selection that suits the requirements of the QoS traffic. In addition, according to the availability of next-nodes to service the traffic class at different situations, the protocol defines rules that can be used for finding appropriate neighbours. For each packet that performs the highest-level tasks (C1) and traffic within a given deadline, the protocol selects the nodes that have a lower hop value compared to the hop value of the sender. This rule aims to achieve low latency for real-time data transmission. For C3 and C4 packets that are non real-time data, the protocol selects data forwarder from the nodes whose hops are equal with the sender node's hop. This selection may incur longer delay transmission for C3 and C4 packets since the number of hops to reach the BS increases compared to the corresponding traffics.

Since a node includes information about individual neighbour sensor nodes, it is possible for the proposed algorithm to build up many solution sets for different traffic requirements. The algorithm performs calculation to find optimum fitness values from the nodes to provide a specific route for certain data class transmission thus balancing the load in the network (lines 27, 33 and 38). This even data distribution decreases the possibility of data loss due to queue overflow and avoids the energy of the sensor nodes to be quickly exhausted thus improving the lifetime of the network. For best effort packet, the protocol avoids choosing the nodes used for other packet type transmission as shown in line 40, thus ensuring differentiating QoS routing. This differentiating QoS routing satisfies the demands of different application requirements as it finds out the best node among several ones by considering the above-mentioned criteria, without having to explore the all possible connections in the network. Each time, a node is selected as the next node of a certain QoS path, this is based on its contribution to the accumulated fitness value. Thus this process reduces possible energy consumption and communication overheads.

The following section presents the packet priority incorporating the queuing model that is used to detect congestion of each node.

Priority based Delivery

Sensed data have different levels of importance due to the traffic class. Priority based Delivery (PbD) assigns a priority value to each packet type that is sent to the communication link according to the type of the traffic. This component employs a multi-queue priority policy that allows the nodes to configure multiple priority queues for different traffic classes by specifying a different importance level for each of the traffic categories in a single service policy. According to the traffic category, each queuing module has a specific threshold to hold the arriving packets based on the priority queues thus this proposed PbD can manage the traffic congestion and further it improves the overall network performances.

According to data classification explained earlier, I define that different data types require different levels of quality and importance and accordingly might have different priorities in terms of data routing (Monowar et al., 2012). Some applications might need the receiving packets at the BS without loss, others might be tolerant to the lost packets problem and the rest could need packets delivery within a given deadline (delay-sensitive). To satisfy multiple QoS according to various data classes, each data category is assigned certain priority and each type of packet is treated differently based on the given priority to meet the demands of the application requirements (Hanzo and Tafazolli, 2007).

Since the traffic classification is used as a base for routing decisions in the proposed scheme for specific WSNs applications, the proposed scheme evaluates the queuing model, to understand the relationship between congestion and delays in a sensor node. Before I proceed further, it is essential to understand the concept of delay in a messaging system. The total delay experienced by packets can be classified into the following categories:

- processing delay: this refers to the delay from the time of reception of a packet in the receiving queue to the time that a packet is passed into the transmitting queue. This delay includes the time for processing a packet through the communication layers in the open system interconnection (OSI) protocol stack,

- queuing delay: this is the delay between the point of entry of a packet in the transmit queue and the actual time of transmission of the packet. The more data transmission on the communication link, the longer the waiting time of a packet,
- transmission delay: this delay depends on the speed of the communication link (nodes data rate) and the packet size, which calculates the time of the transmission of the first bit of a packet to the transmission of the last bit,
- propagation delay: this delay is usually neglected since it is so little, which depends on the physical characteristics of the communication link (the speed of the light).

From the explanation above, I understand that as a network gets congested, the total delay as explained above in the system increases, and this impacts its routing performances. We detect congestion by measuring the length of the queue (Hull et al., 2004), which rises when the packet inter-arrival time exceeds the packet inter-service time (Ee and Bajcsy, 2004). We employ a similar queuing model from the class-based queuing model (Floyd and Jacobson, 1995), which classifies the packets based on priority into real-time and non-real-time queues. The classifier will check the type of incoming packets, which are labelled accordingly in the upper layer and send them to the appropriate queue. The incoming packets of C1 or C2 with deadlines are given higher priority and put into a real-time queue, while C3 or C4 are processed into a non-real-time buffer. To determine the order of packets to be sent out of the queue according to the priority of queuing policy based on each type of data, a packet scheduler is implemented as depicted in Figure 3.5.

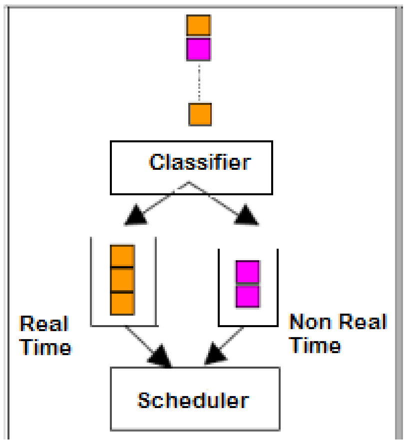

Figure 3.5: Traffic Classification Model

It is assumed that each queuing size has a threshold (Th) to hold the arriving packets. A queue that has the priority x for a node P may hold the number of packets with lower priority $(NQ_L)p$, same priority $(NQ_x)p$, and more critical packets $(NQ_c)p$. For a packet M arriving to the ith queue (Q_i) at node p, the expected waiting time T[Exp_m]p is determined by: (1) the remaining time for a packet in the current process/service of delivery, t[cu]$_p$, (2) the service time for all packets in the queue with higher or same priority, (NQ$_x^p$)+(NQ$_c^p$), and (3) the delivery service time for incoming higher priority packets during the packet M's waiting time, t(NQ$_z^p$). However, the number of packet arrivals always remains less than a threshold, Th. Thus, the total estimated waiting time for a packet M is formulated as:

$$T[Exp_m]^p = t[cu]^p + t\sum_{x,c,z=0}^{Th}(NQ_x^p),(NQ_c^p),(NQ_z^p) \quad (3.4)$$

The above equation explains that the classifier determines the packets transmission by following priority order. The highest priority is assigned to C1, then follows to C2, afterward C3 and the last priority is C4. As plotted in Figure 3.6, the queuing model provides certain queue size for each type of packet to be held, thus maintaining the throughput for each data class.

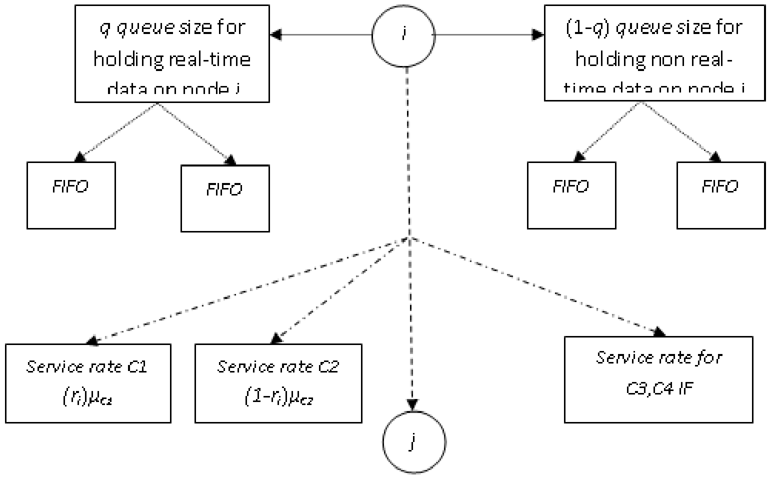

Figure 3.6: Queue Size and Service Rate on a node

To ensure the transmission of real-time data to meet specific deadline, the proposed protocol sets the bandwidth ratio r, which represents the amount of bandwidth to be dedicated both to C1 and C2. This r value is used to calculate the service rate of C1 while (1 - r) value represents the transmission rate for C2. Furthermore, both classes can borrow bandwidth from each other when one of the two types of traffic is non-existent or under the limits. Similar to the service rate policy, the queuing model allows a certain type of packet to adopt other buffer occupancy, which is dedicated to different traffic types when the specific buffer is not used or is below the thresholds. By using the equation (3.4) I can also estimate delay (latency) for a packet transmission from the source to the BS so that each type of traffic class is ensured to meet the demands of the application requirements.

Calculation of queuing and end-to-end delay

The calculation of the queuing buffer as well as end-to-end delay is required to find a QoS path that meets the requirements of each type of traffic. We calculate the total data load on a sensor node i based on the following formulas:

$$(load)^i = P_i S_i + \sum_{k=0}^{j} \binom{j}{k} (\mu n) \quad (3.5)$$

$$(\mu x)^i = (\frac{Ld}{R} + t_{BOT}) \quad (3.6)$$

$$(t_{BOT}) = r(t_{CCA}) + \sum_{a=0}^{a=r-1} t_{Bo}(aMinBE + a, aMaxBE) \quad (3.7)$$

where P_i is a packet interval rate, S_i is a source node, $\binom{j}{k}(\mu n)$ is the neighbours' service rate for relaying packets, L_d is length of data, R is data rate, t_{BOT} is a backoff time that depends on the load on the communication link (channel), t_{BO} is number of backoff attempts, r(t_{CCA}) is a clear channel access time, a is max backoff attempt before failure, aMinBE is 3 and aMaxBE is 5.

Equations (3.5 - 3.7) explain that data load on a node increases if a node receives data from a number of neighbours. The increasing load potentially causes congestion when the service rate for a node to transmit packets (based on random back-off period time) is lower than the interval rate of incoming packets. Congestion causes packet loss due to either channel access failure during a back-off time or when buffer occupancy exceeds a threshold. Since I know data load on a node, I can estimate total queuing delay on a node i as follows.

$$(tqueue)^i = \frac{(Load)^i}{(\mu x)^i} = \frac{PiSi+\sum_{k=0}^{j}\binom{j}{k}(\mu n)}{(\mu x)^i} \quad (3.8)$$

Equation 3.9 describes the queue length for a node. When number of packets increases in a queue and is more than threshold, then packets are dropped as described below:

$$(packetdrop) = \mathrm{P}_i S_i + \frac{\sum_{k=0}^{j}\binom{j}{k}(\mu n)}{(\mu x)^i} > (T_h) \quad (3.9)$$

T_h is a threshold for holding packets in a queue, $(\mu x)^i$ is a rate of real-time packet transmission on a node.

The proposed algorithm assigns higher service rate to real-time traffic classes $(\mu x)^i$ than non-real-time data (1- $\mu x)^i x$. Let us assume that the real-time service rate $(\mu x)^i$ is n(1- $\mu)^i x$ that the algorithm can reduce the rate of packet drop for the real-time packets; calculated as :

$$rate(drop) = \frac{PiSi+\sum_{k=0}^{j}\binom{j}{k}(\mu n)}{(\mu x)^i} < \frac{PiSi+\sum_{k=0}^{j}\binom{j}{k}(\mu n)}{(\mu x)^i} \quad (3.10)$$

In order to guarantee the delivery of real time packets from a source to BS, I present end-to-end queuing delay for a particular path i as:

$$(Tend2endQueue)^i = \sum_{i\epsilon path}^{\sum} hop * \frac{PiSi+\sum_{k=0}^{j}\binom{j}{k}(\mu n)}{(\mu x)^i} \quad (3.11)$$

$$\frac{(P_i)}{(\mu)} = \sum_{i\epsilon path}^{\sum} hop * \frac{Si+\sum_{k=0}^{j}\binom{j}{k}(n)}{(x)^i} \quad (3.12)$$

Based on the above equations (3.8 - 3.12), the distance or hop count between a node and BS can affect the performance of end-to-end delay. The closer a node to the BS, it will have lower hop delay and it will be a more favourable node to deliver real-time packets.

3.4 Metric and Model for Simulation

Before performing experimental works for examining my proposed scheme and the existing QoS routing algorithm (SPEED), I define system metrics and model of simulation to provide an insight in the simulation works. The following metrics will be used to quantify the effectiveness of the routing schemes to address some requirements in WSNs. We design the environment and the parameters of the simulation model for conducting experiments,where each parameter affects differently on the system metrics. The following describes the performance metrics as a basis to measure the performance level of the routing schemes.

3.4.1 Performance Metric

In order to establish a standard set of evaluation criteria based on the requirements discussed in Section 3.3.2, it is important to describe different metrics, which would be used to quantify and describe the requirements and capture the performance of different topologies. The set of evaluation criteria used and their definitions vary quite considerably. The following describe the metrics.

- *Average energy consumption of sensor network*: This gives a good measure of the energy consumed in the network. A routing algorithm with a higher number of nodes having sufficient energy to relay packet data, is desirable. The residual energy at each node is very important to determine, because this energy can be used for a routing scheme to select nodes to build a path. This metric also shows how efficient the algorithm is in terms of the ratio of node failure and the residual energy of the nodes.

- *Lifetime of the network*: The various network lifetime definitions are proposed and used in the literature, but there has not been any consensus on a standard quantitative lifetime definition. In this work, I define the lifetime of the network as the time to the first sensor node die as it provides a simple but effective description of the network lifetime.

- *Average delay per packet*: This is defined as the average time taken for packets traverse from the source nodes to a destination (BS). This is the end-to-end delay that includes queuing and transmission delay as mentioned previously. This metric reflects the response time of the routing schemes. For the applications that deal with real-time data are delay sensitive, this is an important metric. In this experiment, the delay performance is examined in relation to all traffic conditions to get optimal results.

- *Network throughput*: This is defined as the average rate of successful message delivery over a communication channel thus measuring total number of data packets received at the BS. The throughput for all traffic types is considered independently.

- *Network delivery ratio*: This refers to the ratio of the total data received by the BS and the total data sent by the source node. This reflects the performance of the routing technique in maintaining the network function for data delivery. Measuring packet delivery ratio can provide information about the reliability of the network to handle congestion, which affects the network performances with respect to packet loss, delay, throughput and energy consumption.

- *Network utilisation*: One aspect of measuring QoS in WSNs (Wireless Sensor Networks) is to evaluate the node usage in the network. The greater the number of nodes participating in data delivery, the better the routing algorithm in balancing traffic load, which in turn adversely affects the lifetime of the nodes. By using the above-mentioned metrics, I will conduct a series of experiments to measure the quality of the proposed schemes. The simulation environments is described in the following subsection.

3.4.2 A model for WSN Simulation

To assess the effectiveness of the routing schemes on the simulation performance, I design and develop from the beginning a modular WSNs-specific simulator written in object oriented programming language Java. The reason for the development of the tailor-made simulator is to provide and employ necessary modules that fit the simulation functionalities for the research objectives. Even, this tailor-made simulator is not superior to the existing simulators, this custom simulator is more appropriate and efficient as well as the ease of use compared with the existing ones. The tailor-made simulator aims to provide the building blocks of a general simulation model and architecture to deal with the objective of this chapter to apply a QoS differentiating routing protocol in WSNs.

The basic architecture in my simulator includes i) Radio Channel Architecture that employs transmission module, propagation module and reception module, ii) Physical environment architecture which consists of physical channel module, physical sensor module and event generator module. According to the event module, my simulator enables traffic differentiation, which is the main features of the proposed scheme that differentiate with other protocols. This module can classify data into several categories according to the required QoS and iii) Energy module architecture. The last architecture is iv) Energy model architecture that includes the power module for calculating the energy consumption of the different components and the battery module for formulating the battery discharge thereby the residual energy can be measured. Moreover, each node in my simulator equips with a) communication protocol component, b) physical node component that represents hardware characteristics, c) interface component for message passing and d) localised component for estimating the geographical information of nodes and recording the statistic of node's failure according to channel access, dropping packet due to failed buffering and energy depletion. Further, my specific simulator is developed using modular design following new approaches proposed in the thesis. To make the simulator easy to use, I designed a graphical editor to construct topology graph for either static or random node deployment for simulation in WSNs.

At first, according to support heterogeneous traffic based on differentiation QoS requirements, new modules were provided that are 1) a Fitness Function module for enabling route differentiation for different application requirements, 2) multi-priority queuing module that classifies the packets based on priority into real time and non real time queues.

To evaluate the performance of different routing schemes, I conducted various experiments to compare the following: i) adaptation performance in representing changes of the context due to network dynamic, ii) energy consumption for efficiency and network lifetime, iii) distribution of energy consumption due to the 'energy hole problem', iv) delivery ratio, v) heterogeneous traffic and vi) latency (end-to-end delay). The simulator is able to construct the network topology graph which consists of 360 nodes deployed randomly in the target field onto a 300 x 300 m2 grid. To test the robustness of the schemes in relation to the congestion and traffic uncertainty, multiple events are created from multiple source nodes with a random rate of traffic generation.

The JWSN simulator utilises the carrier sense multiple access collision avoidance (CSMA-CA) (Kleinrock and Tobagi, 1975) mechanism that limits the number of packet collisions when nearby devices transmit at the same time. The CSMA-CA uses a collision avoidance technique to ensure the channel availability (free channel) by sensing the channel first before initiating packet transmission. When a channel is busy, a node performs a back-off (wait for a certain period) until a node finds an idle channel. Once the channel is free, the sender node A sends a Request to Send frame (RTS) to the intended receiver node B. If B is ready to receive data, B replies Clear to Send frame (CTS) to A. After receiving CTS, node A sends data and B responds with an acknowledgment frame (ACK) to indicate successful reception. If node A has more data (packets), A has to wait (back-off) a random time after each successful data transfer and competes with adjacent nodes for using the channel.

During RTS-CTS handshake mechanism, any node C within the transmission range of either node B can overhear the RTS or CTS frames. By hearing a RTS frame, Node C should halt sending any message until a CTS is received. This mechanism helps to avoid "hidden node problem" as depicted in Figure 6.6. This figure shows that "hidden node problem" occurs when both node A and C start to send packet simultaneously to node B whereas node A is visible from node B, but not from node C which can communicate with node B.

The simulator exploits peer-to-peer communication where each node is capable of performing self-management and organisation to support multi-hop routing in an ad hoc network. To represent a low-power and low-rate sensor nodes, the simulator defines information about the time and the energy required for every nodes activities based on

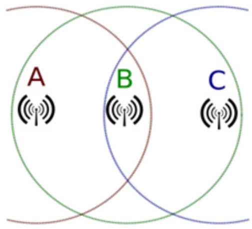

Figure 3.7: Hidden node problem

some parameters in (Kohvakka et al., 2006). The energy to run the transmitter amplifier is 100 pJ/bit/m2 in order to amplify the signal at an acceptable signal to noise ratio (SNR). In addition, the energy and the time required for running transmitter and receiver electronics, information processing and messaging in the protocol's layers and for every nodes activities are based on some parameters (Kohvakka et al., 2006) to represent low-power and low-rate sensor nodes. The bandwidth of the channel is set to 1 Mb/s (Kulik et al., 2002). In the experiments, each data message is assumed to be 1000 bits long and information-processing time in a node is taken between 10 to 20 milliseconds including the time for communication link access for data communication. The medium is assumed symmetric such that the energy required for transmitting a message from nodes A to B and from B to A are the same at a fixed SNR. Therefore, for free space propagation loss (Andersen et al., 1995) with 250 kbps data-rate, energy dissipation is certainly dominated by the long distance transmissions.

Parameter	**Value**
Propagation model	Free space
Node type	Stationary
Sensor Node	360 nodes
Traffic Rate	CBR
Data Rate	250 Kbps
Packet Size	1 Kb
MAC Type	CSMA-CA
Destination	Single BS
Source Type	Multiple/Random
Hello Cycle	8 Secs
Initial Energy	15 J
Transmission Range	30 m (fix)
Packet Deadline (C1, C2)	(300 ms)
Packet speed ($S_s etpoint$)	(700 m/s)

Table 3.3: System Properties

The effectiveness of the proposed routing algorithm is validated through several simulations. Each node has initial energy (15 J) with 30 m transmission range. The simulation is carried out in a scenario that has 360 nodes, which includes ten nodes to generate more traffic. All source nodes generate data with various traffic rates from 1 to 25-pps (packet per-second) and deliver data to a BS. The table 3.3 describes the parameters of the simulation in detail. The simulations are carried out five times and the final results are presented based on the average of the result sets. I set the default end-to-end delay requirement for real-time packet to 0.3 sec to accomplish the International Telecommunication Unit (ITU) G.114 standard, which recommends order value of 0.1 to 0.5 sec. The following section describes the results of the simulation.

3.4.3 Analysis and Comparative Evaluation of the results

Extensive simulation experiments were conducted to quantify the effectiveness of the proposed differentiating QoS routing based Fitness scheme. The experiments were performed to validate the proposed scheme in terms of reliability, latency (packet delay), energy efficiency, network utilisation and throughput using different routing schemes. The results of all experiments are described in this section.

The section presents the results based on two comparative evaluations: (i) weights comparison in TeGaR and (ii) schemes comparison. The comparison between weights in TeGaR is to highlight the strength of each weight in performing route selection according to each data requirement. We use the optimal weight for the proposed scheme for the comparison to another scheme. The second comparison is to evaluate the performance between TeGaR and SPEED with respect to the strength and weaknesses of both schemes in achieving multiple QoS objectives with the current availability of resources. The following sub-section describes further detail of the comparisons.

Comparative Evaluation based on Weight

This section investigates the impact of the usage of different weights that can be used to provide route selection for different packet types with required QoS performances. The weights that are assigned to each attribute of nodes represent relative importance of each attribute in relation to a certain data class.

The resource strategy performs adaptation of route selection based on the energy level of each node according to data types. In this approach, I set the level of energy attribute with a higher weight compared to the weights of other node's attributes since energy is the most critical aspect in WSNs. By increasing the weight level for the energy, the algorithm performs better in terms of cost-efficiency and preservation of resources. This strategy, known as "En_{weight}", is designed to prevent the nodes that have participated in data routing, from draining energy. The "En_{weight}", with higher weight level to energy will be implemented for all traffic types.

The other strategy is "Eq_{weight}". This strategy assigns equal weight (importance level) to all attributes of a node and does not consider the behavior of traffic for routing. This non-weight strategy is developed to evaluate the disadvantage of not applying weight strategy for route adaptation according to the data types. We will examine whether the strategy which does not consider traffic pattern may fail to prevent node energy drainage.

To incorporate resource availability information for route selection while considering the traffic requirements, I propose an "Opt_{weight}" strategy that combines the resources and the nature of the traffic into route selection to consider the applications' needs. The Opt_{weight} strategy factors in the information about resources and traffic-behavior, and computes the adjusted values of weights for the attributes of the node according to the availability of power resources and other communication parameters, which can affect the routing performance.

The objective of the comparative evaluation based on weights can be summarised as follows.

- To evaluate which weight adaptation strategy is better to perform route selection based on QoS objectives and traffic types by selecting a proper node as a data forwarding node with respect to the demands of the application's needs.

- To demonstrate the performance of weight selection strategies for non-critical and critical packet (in terms of energy-efficiency and QoS provision).

In order to achieve the above objectives, experiments were conducted considering best effort traffic delivery (Eq_{weight}), resource adaptation strategy (En_{weight}) and traffic-differentiation strategy (Opt_{weight}), and the results were analysed using following measurements:

- Network packet delivery ratio
- End-to-End delay
- Network utilisation
- Average energy consumption of each node

Effect on Delivery Ratio

In order to see how the weights of the fitness approach behave under different data delivery requirements, the experiments varied the weights of the nodes' attributes, and monitored how the changes impacted the throughput for both real-time (C1 and C2) and non real-time data (C3 and C4). The results are depicted in Figures 3.8 - 3.11. We used three different weight-strategies: (i) Eq_{weight}, (ii) En_{weight}, (iii) Opt_{weight} where the adjusted values of weights to each attribute is defined based on the parameters in Table 3.2.

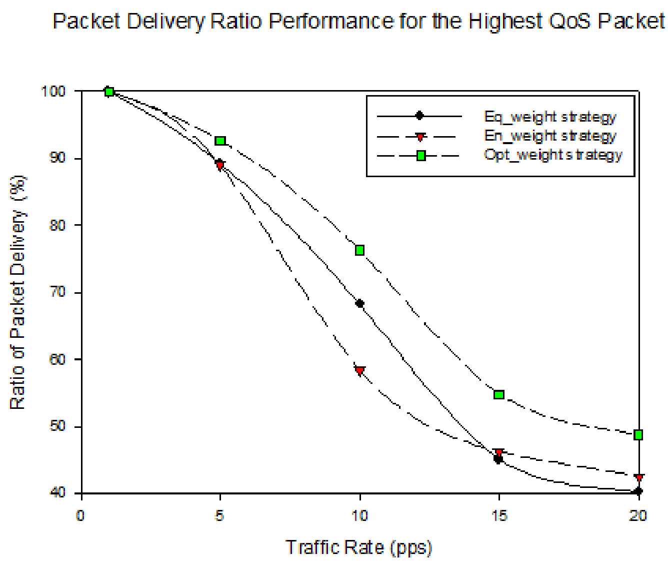

Figure 3.8: Different weights for the packet delivery ratio of C1 traffic

According to Figures 3.8 and 3.11, the data delivery ratio goes down while the packet generation rate increases. This increasing number of generated traffic can cause congestion, which results in many rejection or packet drops causing throughput for all traffic types to decrease. Hence, the ratio of packet delivery decreases significantly as more packets are injected into the network. However, the use of traffic classification can reduce the rate of congestion as it enables load balancing in the network by ensuring the required QoS for each type of packet. Thereby, this module directly affects the continuation of data delivery and improves success delivery ratio as shown in Figures 3.18 and 3.19. These figures show that the proposed scheme have a higher throughput and delivery ratio than SPEED.

As shown in Figures 3.8 and 3.9, since the Opt_{weight} strategy uses traffic-differentiation, it is able to assign appropriate weights that fit the requirements of the data classes. This algorithm performs better compared to the other two weight strategies and provide more effective route adaptation. It allows more packet reception than the other two strategies

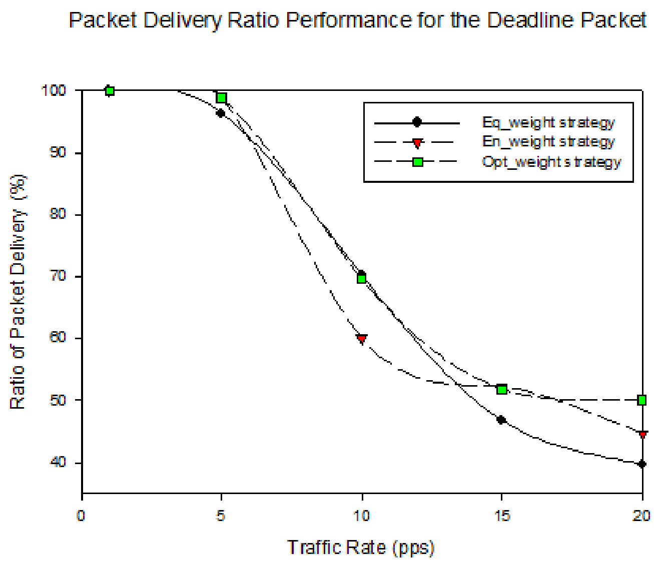

Figure 3.9: Different weights for the packet delivery ratio of C2 traffic

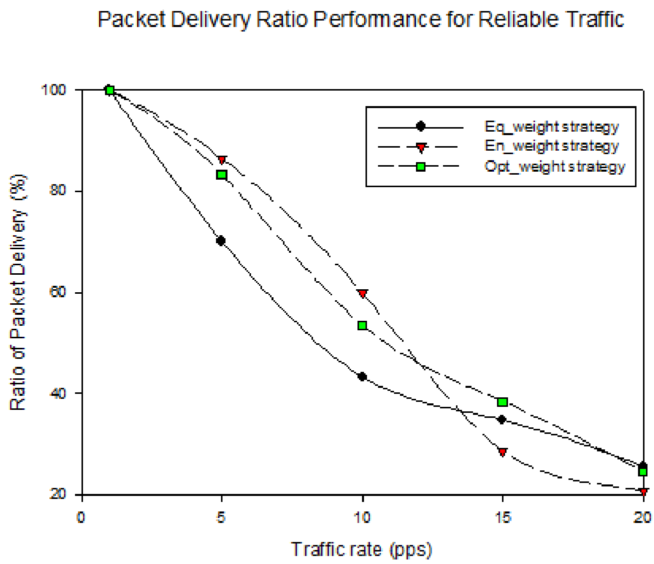

Figure 3.10: Different weights for the packet delivery ratio of C3 traffic

for real-time data which includes both C1 traffic (highest QoS data type) and C2 (real-time data but tolerant to considerable packet loss).

Figures 3.8 - 3.11 show that the non-weight strategy (Eq_{weight}) has the lowest ratio of packet delivery compared to the two strategies. Comparing the non-weight and the En_{weight} strategies, the simulation results demonstrated that the weight strategy with awareness of power resource (En_{weight}) provides a better packet delivery ratio under normal to mid-congestion traffic rates, except for a heavy congested situation This can be due to the fact that the En_{weight} strategy is able to maintain the route to the BS by finding and selecting the nodes which have sufficient energy.

In contrast, when the traffic becomes more congested, the energy consumption in the network increases significantly and this resource-aware adaptation strategy fails to find the nodes with enough power for maintaining the data delivery. While the non-weight strategy that does not consider the traffic type and the behavior of the application's services may remain having neighbour nodes with sufficient energy for routing packets, and this is also

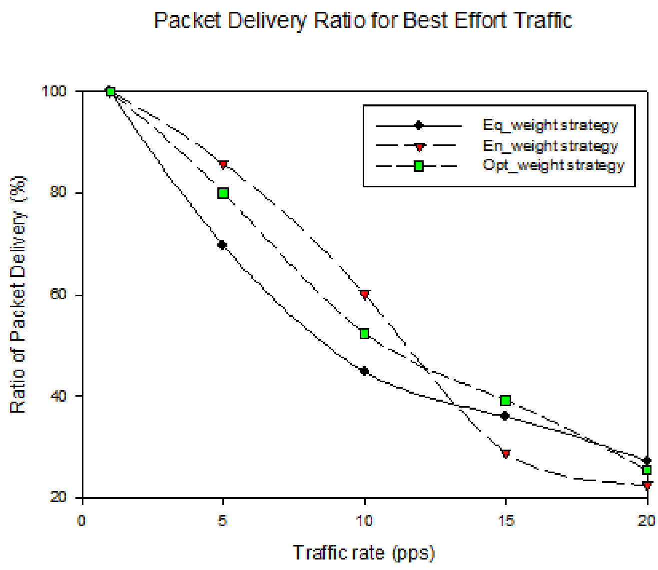

Figure 3.11: Different weights for the packet delivery ratio of C4 traffic

happening to the Opt_{weight} strategy.

Effect on end-to-end delay

Figures 3.12 and 3.13 show the effect of traffic rate on average delay per real-time packet (C1 and C2). The delay increases with the rate since packets incur more queuing delay and wait a longer time for a data transmission that uses a shared link access. Since the approach (Opt_{weight}) has more packets arriving to the BS, this increases delay because many packets travel a full-path from source to BS in the situation when the network is more congested. In contrast, other adaptation strategies, which encounter more lost packets as dropped packets take a small number of hops to their demise so this leads to less delay for the total communication sessions. However, the gap of the delay performance between the approaches is not high particularly for the C1 traffic.

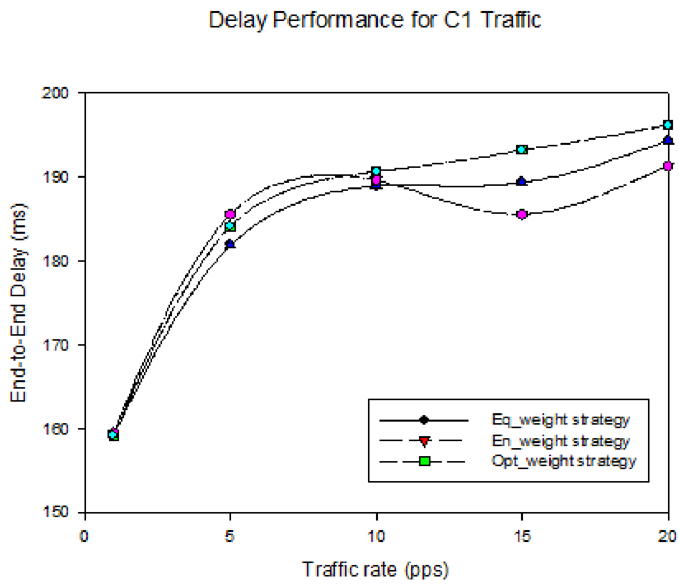

Figure 3.12: The end-to-end delay performance for different weight strategies for C1 traffic

Effect on network utilisation

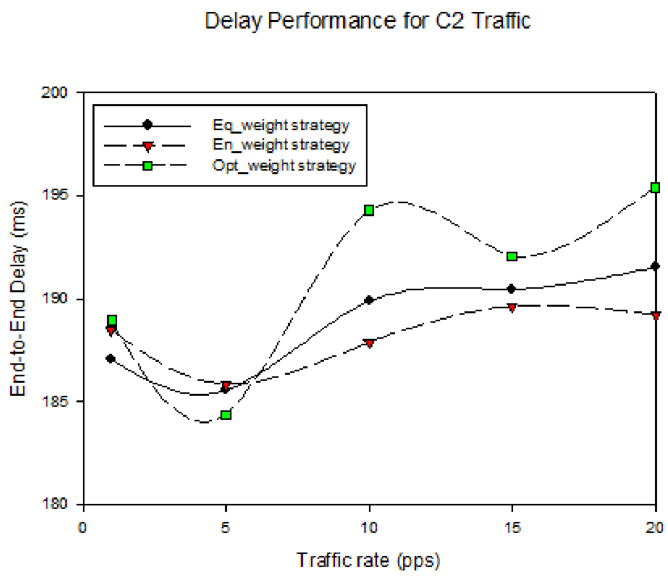

Figure 3.13: The end-to-end delay performance for different weight strategies for C2 traffic

One of the important aspects of QoS performance is evaluating the effectiveness of a routing scheme to increase the node usage in the network, which is defined as network utilisation. This is represented by the rate of node participation in data routing. The strategy (En_{weight}) simply assigns more weight for the energy metric, so it is not difficult to find nodes for relaying packets to destination (BS), thus increasing network usage (see Figure 3.14).

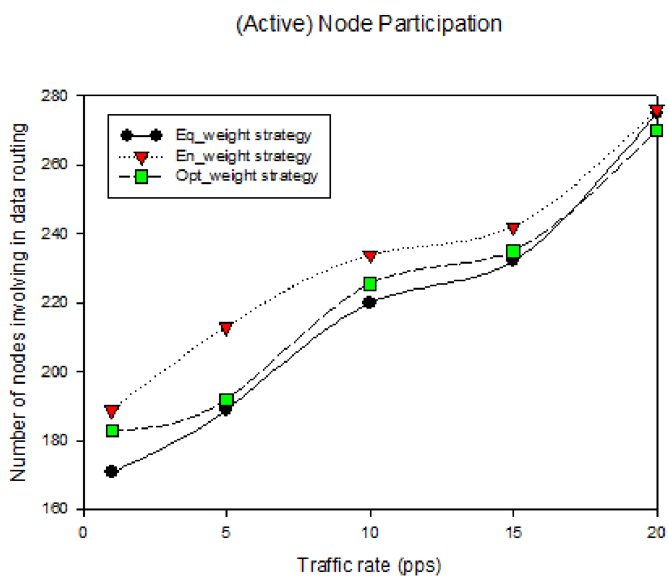

Figure 3.14: Network utilisation between different weight strategies

In contrast, the adaptation strategy that is aware of resources and traffic-behavior has difficulty in finding a node that satisfies the requirements of traffic thus decreasing the number of node participating in the network. However, the participation rate between the approach (En_{weight}) and the Opt_{weight} is not more than 5 percent for all traffic rates and the Opt_{weight} strategy involves a greater number of nodes than the route selection strategy with equal weight to attributes. It indicates that the Opt_{weight} scheme is also efficient to optimise the network.

Effect on energy consumption

One of the other important measurements to evaluate an adaptation strategy is studying its effect on energy consumption of a node. A higher participation rate of nodes will result in more nodes going into active mode and deplete energy more quickly. This often shortens the lifetime of nodes. Figure 3.15 shows the results of my experiment based on energy consumption.

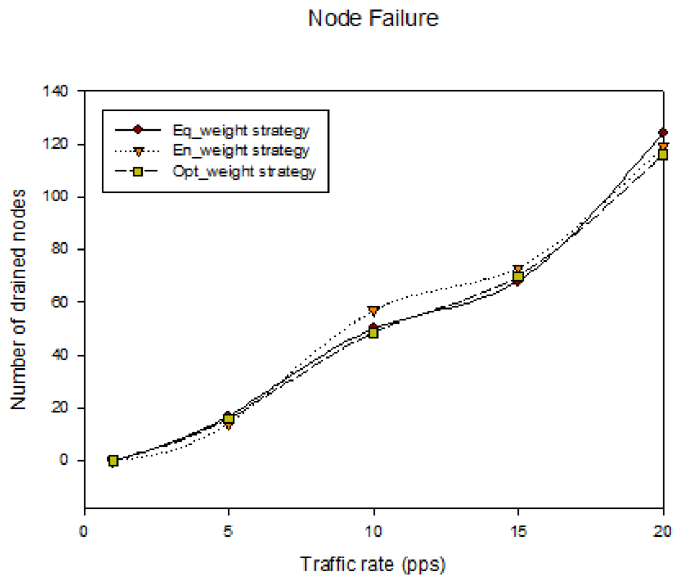

Figure 3.15: The node failure between different weight strategy

As depicted in Figure 3.15, the strategy of the Opt_{weight} demonstrates its strength in reducing node failures as this strategy considers both power resources and traffic behavior. The use of proper weights to attributes due to different types of traffic can maintain the consistency of the proposed scheme in reducing the number of nodes from draining their power resources even if the routing algorithm has to deliver more critical packets that require greater energy levels.

It can be concluded from the experiments as plotted in Figures 3.8 - 3.15 that this comparative evaluation with respect to different weight strategies demonstrates the benefit of the Opt_{weight} for routing adaptation. The performance of the Opt_{weight} provides better results for other QoS metrics. Although the Opt-weight causes a bit more delay compared to the other two weight strategies, it produces a higher level of the data delivery ratio and the energy-efficiency while providing similar rate of nodes to fail due to energy depletion. Hence, the evaluation results reveal that the Opt_{weight} strategy is the most appropriate approach for differentiating QoS paths for multi-objectives QoS routing. Therefore, this strategy is selected and used with the proposed Fitness Function for the evaluation of different routing schemes which is presented in the following section.

Comparative evaluation of different routing schemes

A key component and contribution of the proposed scheme is the Fitness Function method that enables adaptation for route selection based on differentiating QoS requirements according to the traffic type. To evaluate the effectiveness of this novel approach, I evaluate the proposed scheme against the existing QoS routing algorithm, SPEED (He et al., 2003). SPEED is chosen as a benchmark because it is a well-known QoS routing algorithm that has been widely used in WSNs, and applies an approach which is most similar to my proposed scheme. SPEED represents a geographic routing algorithm, uses localised information and considers QoS parameters with minimal control overheads.

Various experiments are performed to compare the sTReNsgths and advantages between these two schemes for satisfying multiple QoS levels for various applications in WSNs. The comparative evaluation applies the metrics such as energy efficiency, packet delivery ratio, network lifetime and average packet delay.

Evaluation based on packet reception rate.

The first comparative evaluation between my TeGaR scheme and SPEED is performed based on the packet reception rate. Figure 3.16 shows that BS (in the proposed scheme) receives more real-time packets (C1 and C2) than non real-time data (C3 and C4) and the number of packet receptions increases with the increase of traffic rate. This results in congestion and difficulty in satisfying QoS paths. Further, it can cause packet losses and decrease the packet delivery ratio (see Figure 3.21). When the network becomes heavily congested (with 20 to 25-pps), the throughput for C1 and C2 continues to increase, but not for C3 and C4. This is the effect of assigning different weights to maintain QoS paths for real-time data delivery. As a result, non-real-time packets (C3 and C4) have failed to be delivered. These results show TeGaR's ability to perform route differentiation that satisfies multiple QoS paths.

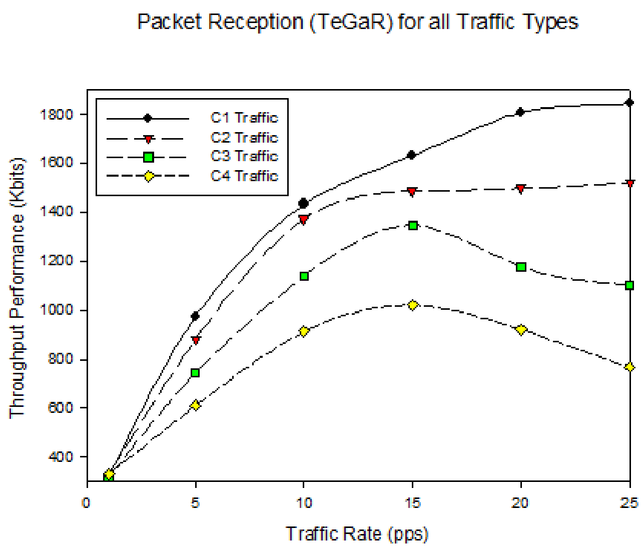

Figure 3.16: The trend of packet reception between the traffic types in TeGaR

In contrast, SPEED uses a nodes' distance and packet speed to route packets, it requires and maintains a uniform delivery speed across the network for each packet transmission to ensure the end-to-end soft real-time communication is achieved. This uniform packet speed is not suitable for forwarding different traffic categories that have different levels of importance and deadline priorities. SPEED is not designed to provide differentiated QoS options, thus resulting in packets treated similarly as depicted in Figure 3.17.

In addition, SPEED prefers to drop packets rather than holding (buffering) packets when the network is congested (see Figure 3.18). In contrast, the proposed TeGaR scheme attempts to maintain consistent throughput under different situations by treating each type of packet differently based on two approaches i.e. the Fitness Function incorporating appropriate weights to each node's attribute for given data types and the priority-based routing.

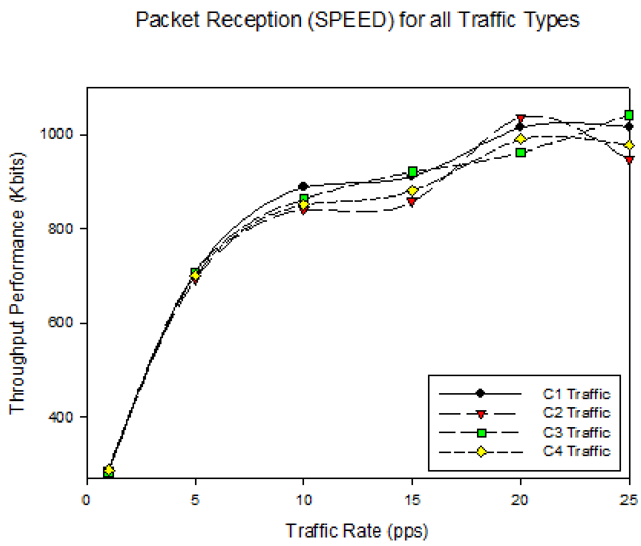

Figure 3.17: Similar Packet reception rate for all traffic types in SPEED

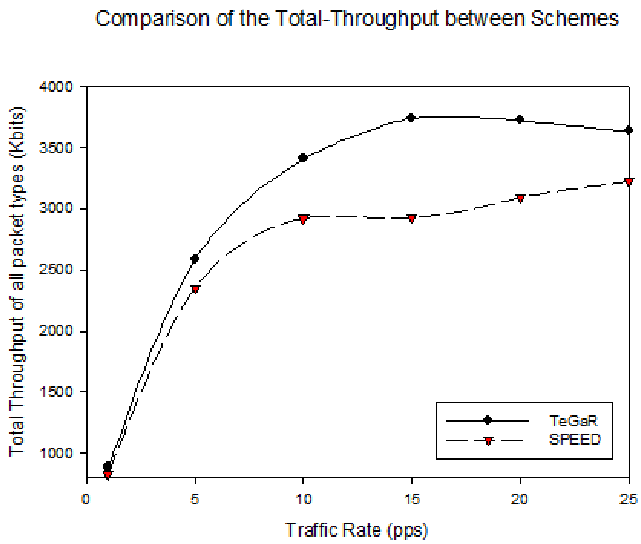

Figure 3.18: The comparison of packet reception between TeGaR and SPEED

Evaluation based on Packet delivery ratio

This section discusses the performance of the TeGaR and SPEED schemes in terms of the level of delivered data to the BS, which represents the effectiveness and accuracy of the schemes in selecting the best node to create an optimal path for data routing. The packet delivery ratio (%) is defined as the number of packets received by BS compared to total packets sent by source nodes. The evaluation results based on packet delivery ratio is shown in Figure 3.19. When the network becomes congested, both algorithms are unable to satisfy the QoS paths. The large number of dropped packets causes a noticeable decrease in the delivery ratio for both schemes. However, TeGaR has a higher ratio than SPEED since TeGaR enables route differentiation considering each type of traffic to maintain data delivery.

Unlike TeGaR, SPEED needs to reroute the congested packets rather than process the new generated packets, which reduces the delivery ratio even further. Moreover, when the network load is too heavy, it is not always possible for SPEED to find the neighbours (nodes) whose delivery speed is higher than the pre-specified packet speed (Set_{Speed}).

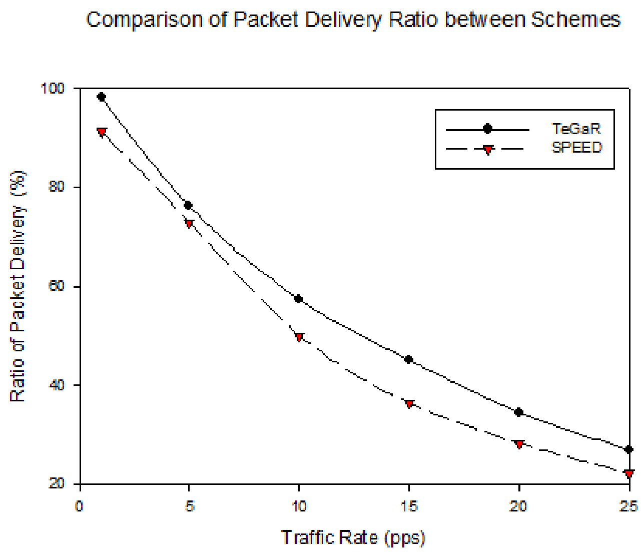

Figure 3.19: Packet delivery ratio between TeGaR and SPEED

Thus, the packet rerouting can fail, and packets will be probabilistically dropped to regulate the network load and find a path such that at least one neighbour can achieve the Set_{Speed}. In contrast, the TeGaR that uses the Fitness Function with the optimum weight strategy incorporated with a multi-priorities queuing system, can hold the number of real-time packets when congestion occurs, and it often finds a neighbour that has the quality to match the requirements of the data class. Therefore, it increases the number of packets received at the BS. The result of the packet delivery ratio for each routing technique can affect the delay transmission.

Evaluation based on average delay per-packet

Average delay per packet is defined as the average time taken for packets traversing from the source nodes to a BS. Figure 3.20 shows the results of comparative evaluation for average delay per packet. The results reveal an increasing delay related to the increase of the traffic rate because higher packet delivery incurs more queuing delay. As expected, the real time packets achieve a lower delay than the non real time packet types for most of the congestion rates. This is happening because the proposed scheme transmits the packets with given deadlines before the non real-time data. This demonstrates that the multi-queuing system of TeGaR is advantageous to the end-to-end delay for the real-time packets (C1 and C2).

However, when I look at a lighter congestion rate (i.e. 5 until 15 packets per second), the difference in delay between the real-time packets and the non real-time data (C3 and C4) is small, that is, about 5 ms. This can be explained because the number of non-real-time packets that arrive to the BS is lower than the real time ones (see Figure 3.16). For instance, for most of the rates, the throughput of C1 and C2 are 30% higher than C3 and C4 and nearly double when the network is heavily congested. This suggests that the packets that do not arrive to the gateway are most probably taking a small number of hops and thus incurring less delay. This figure also shows that at a highest packet generation rate (25 pps), C2 achieves the lowest delay compared to others. The is due to the fact that C2 only guarantees minimum latency without any consideration for reliability and this causes C2's throughput to be lower than C1 (see Figure 3.12).

The evaluation results show that SPEED has a similar trend for the average end-to-end delay for both critical and non-real time packets as shown in Figure 3.21 because SPEED cannot differentiate traffic and does not assign a packet scheduling for different traffic.

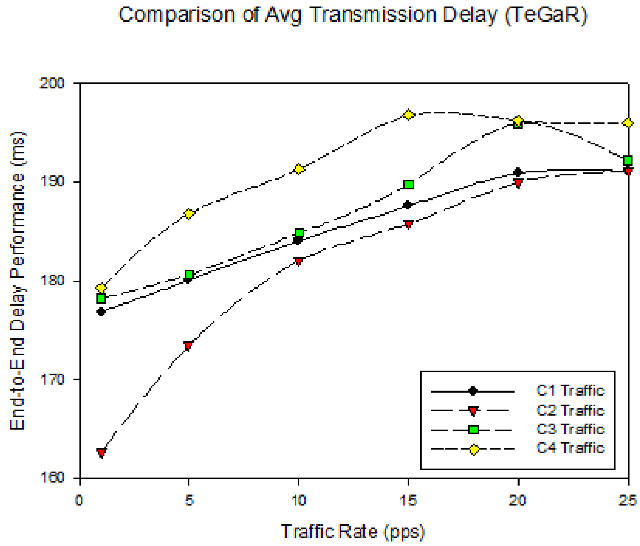

Figure 3.20: The average delay for all traffic types in TeGaR

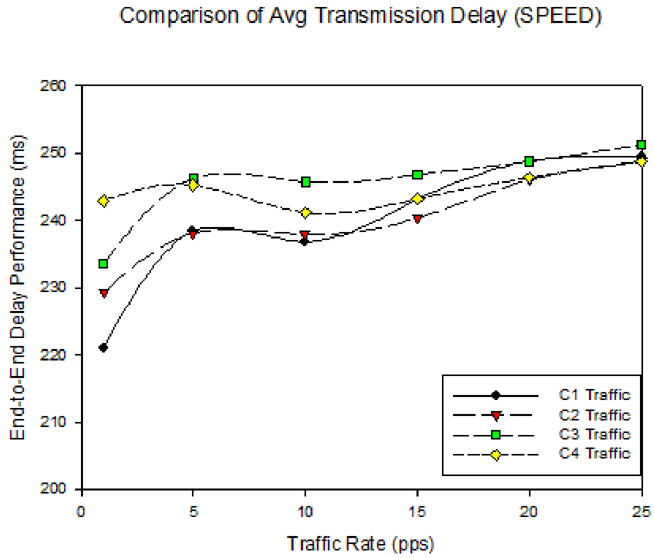

Figure 3.21: The average delay for all traffic types in SPEED

When the network is congested, each type of packet experiences higher delay (more than 250 ms). The delay rate decreases under moderate congestion at about 210 ms when the traffic becomes lower. The reason for the higher delay when the traffic is congested is that SPEED implements a backpressure mechanism to avoid congestion, which slows down packet transmission thus incurring high transmission latency.

With respect to the delay performance measurement, Figure 3.22. shows the ability of TeGaR in achieving a higher number of real-time packets arriving at the destination (BS) on-time. The reason is that TeGaR always finds the nodes that are closer to BS and have lower packet queuing time, to deliver real-time packets. The benefit of TeGaR 's queuing policy that assigns higher priority for real-time packets can reduce the transmission delay. Moreover, the fitness approach provides the best paths for real-time data delivery. Hence, TeGaR can maintain the number of packets arriving within the deadline especially when the traffic is not heavily congested. However, the results show that the SPEED scheme causes a higher number of packets to miss the deadlines. The components of SPEED such as the backpressure mechanism and the requirement for SPEED to find nodes that can maintain the desired hop-speed (Set_{Speed}) increases the packet delay transmission. The next experiment compares the energy distribution amongst nodes between TeGaR and SPEED.

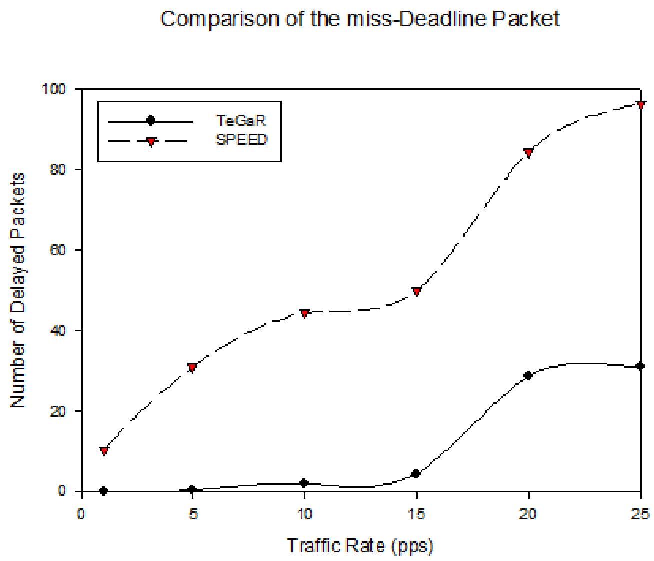

Figure 3.22: The delayed packet due to missing the deadline between TeGaR and SPEED

Evaluation based on energy consumption

In the WSNs routing area, energy consumption for data routing is a critical issue. Many researchers usually calculate the energy costs from the network lifetime, that is the time when the first node may fail or be blocked due to lack of power, physical damage, or environmental interference. However, in an adaptation routing technique, which can provide fault tolerance, the failure of sensor nodes should not affect the overall task of the sensor network. If many nodes fail, MAC and routing protocols must accommodate techniques for new routes for data transmission to the BS. In this experiment, energy consumption was compared by the total nodes that failed in the network during simulation.

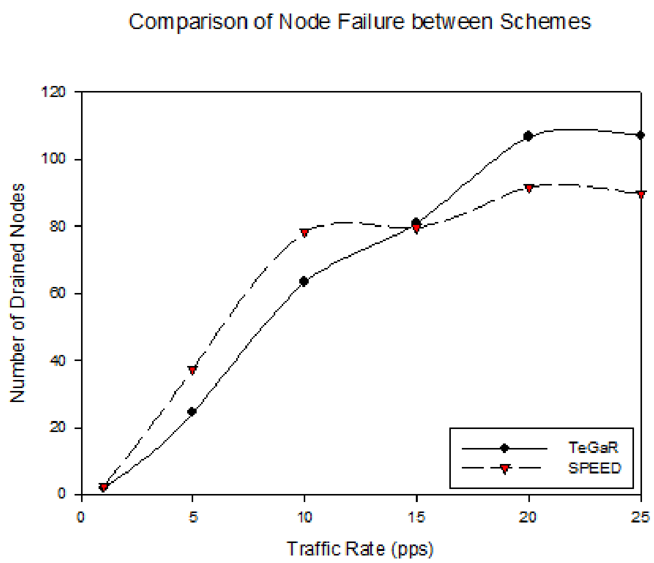

Figure 3.23: The average number of node failure between SPEED and TeGaR

Figure 3.23 shows the results of my comparative evaluation between TeGaR and SPEED based on energy consumption. The figure shows an increase of node failures with the traffic rate. Increasing throughput (TeGaR) causes higher energy consumption and therefore many nodes fail. Note that unlike SPEED, TeGaR uses a fitness approach and considers only the nodes that are estimated to ensure the required QoS. Thus, when the network is normal or not heavily congested, TeGaR is able to reduce energy consumption on more nodes compared to SPEED. However, when the congestion continues to increase, SPEED throughput decreases significantly due to lost packets (see Figures 3.18 and 3.19), which results in the inability of some packets to reach their BS. Thus, the nodes energy is conserved. In addition, the SPEEDˆs backpressure approach reduces the load (data transmission) in the network. The small packet-relaying saves energy for transmission and reception, therefore, it leads to a decrease in the number of node failures.

The last experiment is related to the effectiveness of the TeGaR in terms of routing. Figure 3.24 shows the comparison of workload between SPEED and the proposed algorithm, while Figure 3.25 demonstrates the strength of the schemes in balancing the load and optimising resources and connectivity.

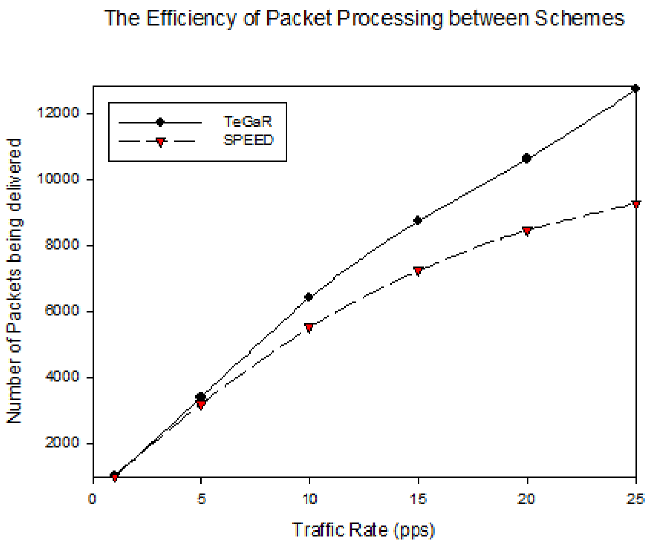

Figure 3.24: The data capacity/workload between TeGaR and SPEED

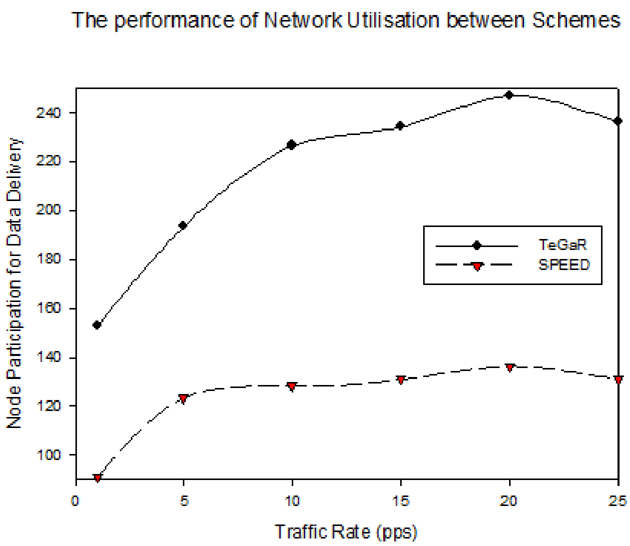

Figure 3.25: Network utilisation between TeGaR and SPEED

Figure 3.24 shows that TeGaR enables the routing algorithm to relay more packets than SPEED with respect to packet generation rate. This figure ensures that TeGaR provides a more effective and efficient routing scheme for applications that perform higher-level tasks and require huge data delivery, such as multimedia or real-time imaging sensory. The results in Figure 3.24 differ than the Figure 3.18; as Figure 3.24 presents the number of data being transferred in the network (workload performance) while Figure 3.18 shows the rate of packet reception at the BS.

Figure 3.25 shows the effectiveness of TeGaR in optimizing the QoS network by increasing the network utilisation. TeGaR, that provides different paths for different traffic classes, is able to distribute the load amongst the nodes, while SPEED uses only a path that can meet a certain hop speed; thus, decreasing the nodes usage. In addition, the increasing rate of node participation in data routing increases the energy usage in the network. However, the consumption is distributed amongst a higher number of nodes thereby prolonging the lifetime of the nodes as well as maximising the connectivity.

Figure 3.25 shows a continuous rise of node participation with regard to the increasing packet generation rate. However, as the traffic is generated randomly that affects the mechanism of node selection, in some cases when the packet generation rate at 15 pps, the trend of participation plots a bit lower rate.

3.5 Summary

In this chapter, a differentiated QoS paths based Fitness Function with an optimum weight strategy for multiple QoS routing is proposed. The proposed multi-objective QoS scheme has the following characteristics: i) it takes into account multiple node attributes in a single QoS-based routing approach to enable route differentiation for each type of traffic, thus incurring power balance, ii) it exploits traffic diversity, which is typical for most scenarios of applications, and provides a differentiated routing using different QoS metrics according to different data types, iii) the route differentiation for different data requirement is developed using a modular design following the traffic classification, iv) it considers energy notion for all types of packet according to its routing decision and all differentiated QoS links construction are restricted to use local neighbourhood information as well as geographical information to achieve routing efficiency, v) it uses multi-queuing with different priority according to the data type, giving more priority to both critical and delay-sensitive packets for achieving lower latency. This multi-queuing has been implemented in the network layer to avoid changes at the lower layers so the proposed protocol may not be intended for specific MAC protocol as long as the protocol employs an acknowledgement mechanism, and vi) each path is created using the Fitness Function approach that evaluates the qualifications of each node regarding QoS aspects.

According to the aferomentioned features, the proposed scheme outperforms SPEED, a prominent QoS algorithm that uses geographical information for routing by offering better QoS for each data classification while considering power efficiency. The simulation results demonstrated that the proposed scheme is applicable to traffic differentiation for supporting the demands of different application requirements.

However, there are some areas where the performances of the proposed Multi-objective QoS scheme can be enhanced further. First, to address the problem of 'energy hole problem' due to imbalance energy consumption in the network according to the nature of many-to-one routing in WSNs, the proposed scheme proposes a non-uniform transmission range strategy that allows the nodes to extends their transmission ranges to a certain level when their number of forwarding neighbours decrease below a certain degree. This extended approach is able to distribute energy consumption evenly, thus to avoid an early network dysfunction.

Second, due to the characteristic of the non-uniform node deployment and link instability of the design of the MAC protocol in WSNs, data transmission in this network leads to a high possibility of broadcast flooding and unguided packet transmission that wastes energy consumption and degrades a data delivery ratio. To address this issue, I proposed and developed a routing region approach that aims to control the area (region) of routing for individual communication sessions between nodes. In this way, packet detouring or broadcast flooding can be reduced, thus decreasing routing overhead and time taken for data delivery further saving more energy.

Third, energy conservation without losing accuracy is another issue that should be addressed. As the characteristic of WSNs have restrictions on energy and device capabilities, it causes sudden and frequent changes in the network. Knowledge of occurring situations in the network is able to provide situation-awareness for adaptation to represent minor changes of network dynamics. The adaptation approach aims to improve energy management further without compromising accuracy for applications such as routing regions in the proposed differentiating QoS routing scheme.

The research highlights these three challenges and propose and design three enhancements to the existing approach in the next three chapters.

Chapter 4

Non-Uniform Transmission Range Algorithm

The previous chapter described my proposed TeGaR framework for differentiating QoS requirements according to data types using the Fitness Function approach, and also pin-pointed issues and challenges of 'Energy Hole Problem'. This chapter introduces the first of the three proposed enhancements to the TeGaR scheme for addressing the energy hole problem with a non-uniform transmission range strategy. The proposed approach is named Transmission Range Extension based on the Neighbourhood Size (TReNs).

4.1 Introduction

In the previous chapter, the QoS differentiating routing based Fitness Function scheme, named TeGaR was introduced. This chapter discusses the first enhancement scheme, which is a transmission range extension based on the neighbourhood size routing. The second proposed scheme is derived from non-uniform and adjustable transmission range strategy that have been extensively used in WSNs (Wu et al., 2008; Li and Mohapatra, 2007; Jarry et al., 2006). With the adjustable transmission range approach, the algorithms are able to distribute power consumption equally (evenly) in all parts of the networks, thus the energy depletion can be nearly balanced, and the WSNs can perform the tasks for a longer duration of time without any part of the networks dying prematurely. The motivations for adjustable transmission range extension algorithms are discussed below.

Designing non-uniform transmission range strategy that allows nodes to adjust or extend their radius of transmission range is required to balance power consumption in the entire network. This approach aims to mitigate the creation of the 'energy hole problem' due to uneven energy depletion according to the characteristic of many-to-one traffic pattern in WSNs. This problem firstly occurs whenever huge amount of data from multiple sensor nodes are relayed to the BS thus resulting in nodes locating around the BS need to relay more traffic compared to other nodes in other regions of the networks. Hence, the nodes close to the BS suffer much faster energy dissipation rate thus have shorter expected lifetime, and the function of the network will end prematurely even the total energy in the network is left unused. As a result, no more packets can be delivered to the BS even if the nodes from other regions still have sufficient energy. Furthermore, the next packet transmissions are lost, and these reduces the throughput. In addition, this 'energy hole problem' is detected when all sensors generating data, communicate directly to the BS (single hop transmission) as these nodes, which may be farther away from the BS would deplete their energy faster (Madkour et al., 2013). In this chapter, I will discuss the 'energy hole problem' according to the first case.

This chapter aims to propose a new adjustable transmission range extension scheme for distributing power consumption in the network so that many sensors in the network may have similar lifetime (Li and Mohapatra, 2007). This approach addresses one of the main objectives of WSNs, which is lengthening the network lifetime in energy scarce networks. The first enhancement approach termed Transmission Range Extension based on the neighbourhood Size (TReNs) increases the range of node's transmission when the neighbourhood size decreases under a certain degree. The part of this chapter has been published in (Rizal et al., 2012). TReNs aims to maintain a number of neighbouring nodes that are closer to the BS for data forwarder, so that the data delivery can be maintained resulting in an improved data capacity at the BS (Jarry et al., 2006). The process of the proposed approach is depicted in Figures 4.1 and 4.2. Figure 4.1 shows that all neighbours died due to energy depletion resulting in the data cannot be delivered at all.

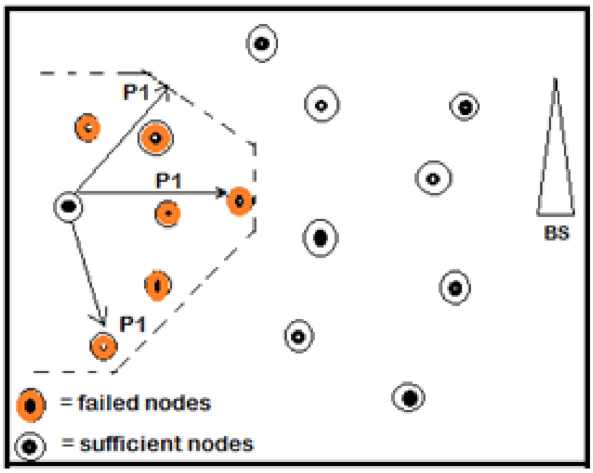

Figure 4.1: Data delivery by using uniform transmission range

In contrast, Figure 4.2 describes that the node is able to find new neighbours that are closer to the destination by extending its transmission range thus the data delivery is maintained and the delay of transmission is reduced while saving the energy of the previous neighbours to improve the residual energy of the networks.

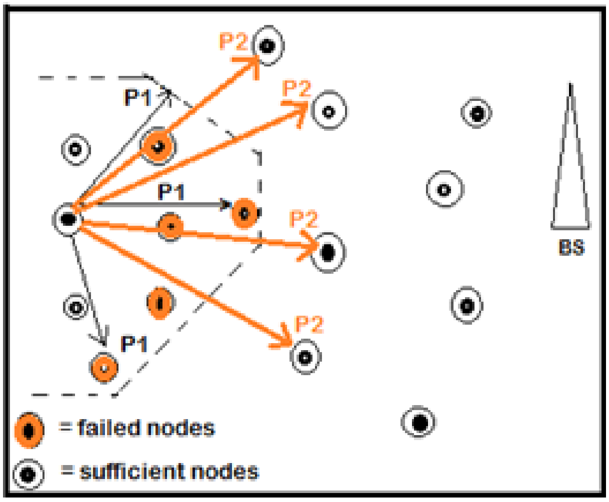

Figure 4.2: Data delivery by using a non-uniform transmission range

Section 4.2 describes the concept of the 'energy hole problem' to introduce some background knowledge in sensor network model and the energy model, and to discuss some related assumptions to facilitate the analysis (mathematical formula) for the proposed approach. The basic concepts of the adjustable transmission range are discussed in Section 4.3. In this section, I also describe the operation and interactions between the modules of the proposed scheme as well as the integration of the first enhancement approach to TeGaR. Section 4.4 details my experiments (simulations) to demonstrate the performance of the proposed scheme and the analysis based on the simulation results. Finally Section 4.5 summarises the chapter.

4.2 The concept of the energy hole problem

In this section, I examine the concept of the 'energy hole problem' as a situation if some critical parts of the network run out of energy while other parts of the network still have a lot of residual energy leading to premature network dysfunction. With respect to this problem, I describe the model of the network and the energy in WSNs to illustrate the characteristic of the 'energy hole problem', and identify key factors that contribute to the uneven energy consumption rates. Both models are then used to analyse the problem of 'energy hole' mathematically, and to validate my proposed scheme through simulation. The following sub-sections describe both the network model and the energy model as well as related assumptions in more detail.

4.2.1 The illustration of the energy hole problem in the network model

In this sub-section, there is a need to describe a network model that would enable evaluation of the creation of 'energy hole problem'. This model can be used to study the functional behavior of uneven energy depletion due to the intrinsic communication pattern of WSNs. According to this network model, the typical communication patterns that are classified into (a) one-to-many; (b) many-to-many and (c) many-to-one, can be used to investigate and characterise the occurrence of the 'energy hole problem'. However, in this case, I are more concerned to analyse the severe problem caused by imbalance energy consumption in many-to-one sensor networks. The following details highlight assumptions of the networks.

In the proposed WSNs model, most of the sensor nodes are deployed randomly. Each node knows its coordinates which is obtained from any triangular measurement or global positioning satellite (GPS). Initially, these nodes communicate with each other by using the same transmission power range (P_{range}) in a circular region with a radius of R (Keeney and Cahill, 2003). Any node within the range of the transmission is called as a neighbour. If a node is Ni, then the set of neighbours of N_i is N_{Nj}: dist $N_{Ni} \leq P_{range}$. In this model, I consider three types of neighbours including: i) the advance neighbours, whose locations are a hop from the BS, ii) the close neighbours, where their hop positions to BS are equal to Ni, and iii) the non-forwarding neighbours that have farther locations from the BS as compared to N_i.

The network is designed such that each sensor has more than one neighbour. All the sensors are homogeneous and each of them has a unique identity (ID) number. Data transmission is triggered by an event detection mechanism which is generated constantly also termed as CBR (Constant Bit Rate). We assume that the sensed data is sent from multiple source nodes to a single BS, which is located at the center of the region (see Figure 4.2). The sources use varying rate of event generation in order to create traffic uncertainty and congestion situation. Thus, I can evaluate the performance of the proposed scheme when the situation becomes so congested.

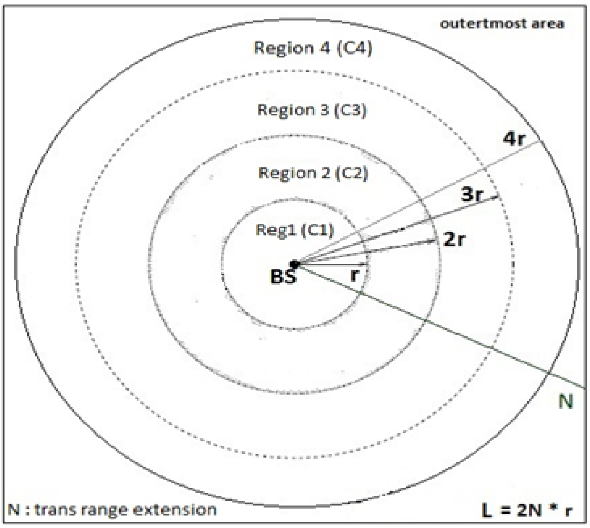

Figure 4.3: Circular regions in the partitioned network

At first, the BS can only communicate directly with the nodes in its radius range, R; then a node in BS's range exchanges information with farther nodes based on radius R, and this communication continues to the nodes in the edge of the network. Thus, the network will be partitioned into several circular regions with radius R. Let us assume that the first circular region close to the BS is denoted as region 1 (circular 1) and the ith circular region is region i or C_i. Hence, the region C_i is composed of nodes whose distances to the BS are between regions (i -1) and i. Therefore, nodes in the region C_i will forward data generated by themselves and also nodes in the outmost regions of C_i, where outmost regions of C_i are $C_i + n$ | i ≤ n ≤ L; where n denotes a constant, which represents the number of transmission range extensions and L is a network width. In this work, the assumption of data aggregation at any forwarding nodes is neglected. The following discusses the energy for processing and communication of sensor nodes in WSNs.

4.2.2 The illustration of the energy hole problem in the energy model

In this sub-section, I describe the 'energy hole problem' from an energy perspective. We attempt to formulate the energy consumption for the activities of nodes that include sensing, processing and transmitting to illustrate the creation of the imbalance energy depletion in WSNs. We intend to use the formula of energy consumption model to evaluate whether the simulation results match with the analytical model. Further, this formula can be used to verify my proposed scheme in the simulation.

In here, each node is supposed to be capable of performing self-management and organisation to support multi-hop routing in WSNs. The information of parameters about the time and the energy that are required for every nodes activities such as sensing, computation and communication are supplied by the Medium Access Contention (MAC) protocol in (Kohvakka et al., 2006).

According to the network model as plotted in Figure 4.3, the energy model is then simply described as follows. Every node has initial energy (e_i; e_i > 0), while the BS is equipped with unlimited power resource. The model determines d_t unit of energy for a node to send a packet (1Kbyte) and costs d_r unit of energy for receiving a packet data, where $d_t = d_r > 0$. Further, the energy depletion for both transmission and reception activities increases along with the increasing radius τ (τ>R) the sensors use as shown in Figure 4.3.

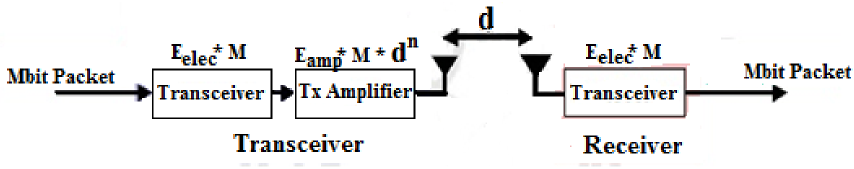

Figure 4.4: The radio energy model for both transmission and reception activities

According to Figure 4.3, the values of d_t and d_r can be represented as explained in (He et al., 2004) as follows:

$$= \; d_t = M_{bits} * (e_{elect} + (e_{amp} * d^n)) \; (4.1)$$

$$= \; \delta_t = M_{bits} * e_{elect} \; (4.2)$$

Referring to the above equations (4.1 and 4.2), M is the length of the packet data (bits), e_{elect} is the energy for radio electronics, e_{amp} is the energy for amplifier to do a communication activity, d is a distance between two corresponding nodes and n is a path loss, where according to Rappaport (2005), the typical value for n is 2 or 4. In this model, I use n = 2.

4.2.3 Analysis of the 'energy hole problem' using both the network and the energy model

This subsection provides an analytical model for calculating the energy consumption of nodes in the network using the network and energy models described previously to investigate the 'energy hole problem'. This analysis is used as a basis to study and provide such techniques that can be implemented in the proposed scheme for mitigating the 'energy hole problem' due to uneven energy consumption in WSNs.

According to the network model illustrated in Figure 4.3, the sensors are randomly deployed onto L x L m^2 grid, where L is the width of network with the length of L = N*r meters. In here, the packet traverses using multi-hop communications across the circular regions until it reaches the BS. The amount of traffic, the sensors have to send will increase dramatically if the distances to the BS become smaller (for the nodes in region 1 (C1) as depicted in Figure 4.3. Thus, these sensors around the BS tend to fail early in terms of energy, causing the energy hole problem and network to become disjointed. This will result in an early network dysfunction where no sensor will be active.

According to a random node deployment in the proposed network model, the node density of each circular region may not be uniform. The node density is defined as the total number of nodes in the network (circular region) divided by the region of the sensor network (circular region), $\rho = \frac{\sum_1^n node}{A_i}$, A_i is the width of the ith circular region. Referring to Figures 4.1 and 4.2, if the width (radius) of each circular region is r_i, then the width of the sensor network for all regions is ∀r = 2Nr, where N represents the number of transmission range extensions. The number of packets flowing in the sensor network is ∀packet = M (see Figure 4.3). Thus, the traffic load that the nodes in the circular region 1 have to relay is:

$$(Load)^{region1} = \frac{Trafficloadinthesensornetwork(\forall circularregion)}{numberofnodesincircularregion1} \; (4.3)$$

$$= \frac{\rho_n(\forall r).(\forall r).\forall packet_n}{\rho i(\pi r^2)} \; (4.4)$$

$$= \frac{\rho_n\{(2Nr).(2Nr)\}M}{\rho_1(\pi r^2)} = \frac{4N^2r^2M_n}{(\pi r^2)} \; (4.5)$$

$= \frac{4N^2M}{(\pi)}$; M_n is constanta (bytes data) (4.6)

To investigate the difference of traffic load between nodes that are closer to the BS and those that are farther, similarly I can obtain the traffic load in the circular region 2 as follows.

$Load_{region2} = \frac{Trafficloadoutsidecircularregion1}{numberofnodesincircularregion2}$ (4.7)

$= \frac{\rho_{n-1}*\{(Area_n)-(Area_{reg1})\}*\forall packet_{n-1}}{\rho_i.Area_{reg2}}$ (4.8)

$= \frac{\rho_{n-1}.\{(2Nr)^2-\pi r^2\}M}{\rho_2(\pi(2r)^2-\pi(r)^2)} = \frac{(4N^2-\pi)r^2M_{n-1}}{(4\pi-\pi)r^2}$ (4.9)

$= \frac{(4N^2-\pi)M_{n-1}}{(3\pi)} = \frac{(4N^2-1)M_{n-1}}{(3\pi)}$ (4.10)

$= \frac{1}{3}\ \frac{(4N^2-1)M_{n-1}}{(\pi)}$, M_{n-1} is constanta (bytes data) (4.11)

If I compare between the equation 4.6 and equation 4.11, it indicates that each node in the region 1 (around the BS) has to relay about three times higher number of packets rather than the average node in the region 2, which is farther to the BS. The difference (gap) in terms of traffic load can increase significantly if the comparative analysis is performed for the outermost region (for example the circular region 5). The calculation is shown here:

$Load_{region4} = \frac{Trafficloadoutsidecircularregion3}{numberofnodesincircularregion4}$ (4.12)

$= \frac{\rho_{n-3}*\{(Area_n)-(Area_{reg1-4})\}*\forall packet_{n-3}}{\rho_i.Area_{reg4}}$ (4.13)

$= \frac{\rho_{n-3}.\{(2Nr)^2-\pi r^2\ -\pi(2r)^2\ -\pi(3r)^2\}M_{n-3}}{\rho_4(\pi(4r)^2-\pi(3r)^2} = \frac{(4N^2-\pi)r^2M_{n-1}}{(4\pi-\pi)r^2}$ (4.14)

$= \frac{(4N^2-14\pi)r^2M_{n-3}}{(9\pi)r^2}$ (4.15)

$= \frac{(\frac{4}{3})N^2-4\frac{2}{3})M_{n-3}}{(3\pi)}$ (4.16)

$= \frac{1}{9}\ \frac{(\frac{4}{9}N^2-\frac{14}{9})M_{n-3}}{(\pi)}$, M_{n-3} is constanta (bytes data) (4.17)

As shown in the equation 4.17, the traffic load per node in the area (region) around the BS and the outer region 5 increase nearly nine times; and is even greater based on both the node density and the total number of source nodes. The ratio becomes much higher if it is compared to the outermost region. This is because the nodes in the innermost remote region do not have to relay packets from nodes in other parts of the sensor network. Furthermore, I integrated the equations (4.3 - 4.17) into the equation formulas 4.1 and 4.2 discussed earlier, which results in the model of energy consumed per node in each circular region. The following formula calculates the estimation of the energy costs of a node in the BS's inner circle.

$= \ e_{cons} = d_s + d_t + d_r$ (4.18)

where d_s is the energy for sensing events in the region 1, d_r is the energy for receiving traffic from the outer side of the region 1, while d_t is the energy for transmitting load in the region 1. The energy d_s is defined as follows.

$(RelayTraffic)_{region1} = \frac{Trafficloadoutsidecircularregion1}{numberofnodesincircularregion1}$ (4.19)

$$= \frac{\rho_{n-1}(Area_n)-(Area_{reg1}).\forall packet_{n-1}}{\rho i(Area_{reg1})} \quad (4.20)$$

$$= \frac{\rho_{n-1}\{(2Nr).(2Nr)^2-(\pi r)^2 M}{\rho_1\pi(r^2)} = \frac{4(N^2-\pi)r^2 M_{n-1}}{(\pi r^2)} \quad (4.21)$$

$$= \frac{4(N^2-\pi)M_{n-1}}{(\pi)} = \frac{4(N^2-1)(M_{n-1}}{(\pi)} \quad (4.22)$$

$$= \frac{(4N^2-1)M}{(\pi)} \; ; M_{n-1} \text{ is a constant (bytes data) } (4.23)$$

Hence the total energy costs per node in relation with formula 4.18 is:

$$=\text{M}e_{sens} + \text{M}(\tfrac{4N^2}{\pi}\text{-1}).(e_{elec}) + \text{M}\tfrac{(4N^2)}{(\pi)}\ (e_{amp}\ .\ d^n)(4.24)$$

According to formulas 4.18 - 4.24, I can also define the energy consumption of each node in the region 2 as:

$$=\text{M}e_{sens} + \text{M}\ \frac{\rho_{n-1}.\{(2Nr)^2-\pi(2r)^2}{\rho_2(\pi(2r)^2-\pi(r)^2}.e_{elec} + \text{M}\tfrac{1}{3}\ \frac{(4N^2-1)}{(\pi)}.(e_{amp}.d^n)(4.25)$$

$$=\text{M}e_{sens} + M\tfrac{1}{3}\ (\tfrac{4N^2}{\pi}\text{-4}).\varepsilon_{elec} + \text{M}\tfrac{1}{3}\ \frac{(4N^2-1)}{(\pi)}.(e_{amp}.d^n)(4.26)$$

According to the above analysis based on the mathematical model, different energy consumption between the nodes in different regions is justified by the formulas 4.25 and 4.26, and these differences become larger if I compare to the nodes in the outermost region in the sensor network. This gap has been experimentally demonstrated in Section 4.3 as shown in Figure 4.5. This figure describes that most nodes around the BS tend to die quickly as they have to relay heavier traffic than other nodes in different regions. This leads to the creation of the 'energy hole'. In order to address the 'energy hole' problem, the following section proposes a novel routing technique that aims to equally distribute energy consumption in sensor networks.

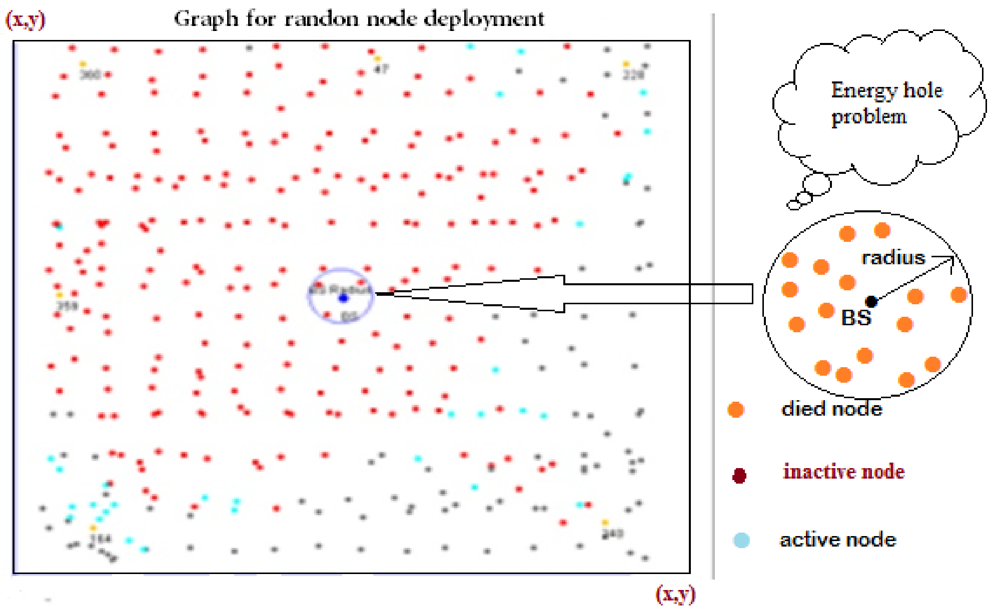

Figure 4.5: The existence of the 'energy hole problem' around the BS

4.3 Transmission Range Extension based on neighbourhood Size (TReNs)

In this section, based on the description and the results of the analytical model explained above, I present a novel adjustable transmission range extension based on the neighbourhood size (Rizal et al., 2012), termed TReNs, which aims to balance energy consumption amongst the sensors for mitigating the problem of energy hole in many-to-one sensor networks. The following sections discuss the proposed scheme in detail.

4.3.1 An overview of the proposed TReNs

The proposed novel scheme is developed based on the analysis derived in the analytical model. The proposed scheme implements an adjustable transmission range routing based on sensor node's position information and the neighbourhood size/degree of nodes to balance energy consumption in the network to reduce the occurrence of the 'energy hole' while improving the lifetime of the networks that leads to increased data capacity at the BS. The neighbourhood size is defined as the number of forwarding neighbours that can assure the network connectivity. This approach is considered to achieve efficient routing as it avoids the nodes to send the information directly to the BS that consumes large amounts of energy as well as reduces the creation of 'energy hole' caused by multihop routing, especially at the area (region) around the BS. The following figure describes the first enhancement approach of TeGaR that includes a forwarding neighbourhood size approach and a transmission range extension approach.

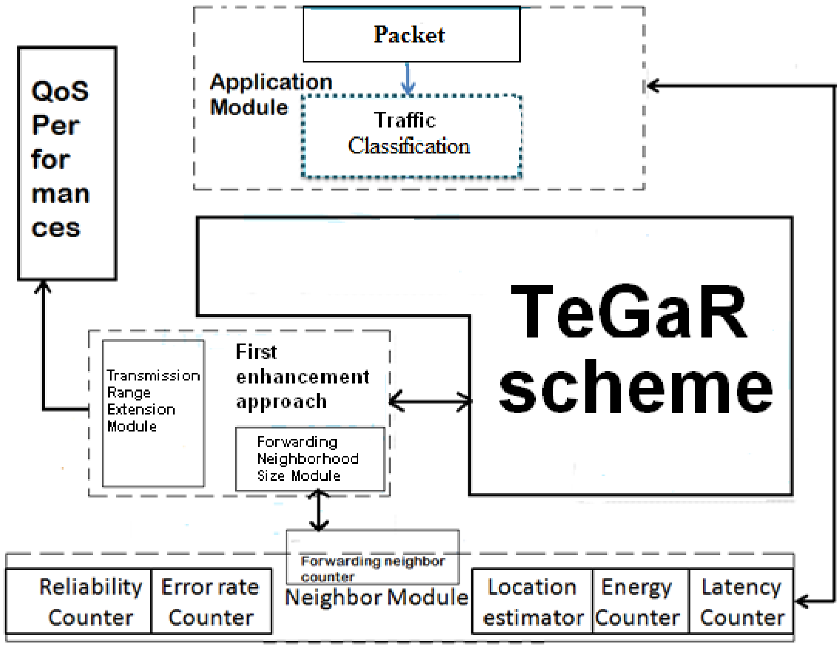

Figure 4.6: The proposed first enhancement scheme: TReNs

The radius of transmission range often affects the network performances in terms of throughput, delay and energy consumption (Wu et al., 2008), and the range for communication between nodes can directly affect the degree of node connectivity (Dallas and Hanlen, 2009). A long transmission range can include more nodes closer to the BS to deliver information to the BS thus the capacity of data received by the BS is increased while the number of hops that packets traverse to the BS can be decreased. However, the increase of transmission radius may cause far more interference that impacts loss of packets. We may control the interference simply by reducing the radius of transmission range but, this implies the increasing delay for packets to BS as well as creating more overheads and internal traffics at intermediate nodes, which affects the overall performance of the routing algorithm. Therefore, it is highly important to address this trade-off between the adjustment/extension of the transmission radius and the QoS performances.

In addition, in a sparse network, a small transmission range causes network disjoints where many nodes will be isolated and the probability for network connectivity decreases. In contrast, when the transmission power is amplified, the degree of connectivity increases with more connections, and thus the amount of traffic load increases that may cause the nodes overhear data transmission even the data are not intended to them. However, in this work, I assume that the MAC protocol is working well to reduce the possibility of overhearing node that wastes energy. Therefore, developing an effective adaptation for adjusting the transmission range is a key challenge to achieve energy saving and QoS performances.

The following subsections describe the enhancement approach that includes Forwarding neighbourhood Size module and Transmission Range Extension module in more details.

4.3.2 Forwarding Neighbourhood Size (FWS) module

This module is responsible to define the number of neighbours having distance closer to the BS that can be used to deliver information to the BS (a destination) as described in Figure 4.7. According to data routing, if this module finds the number of forwarding neighbours (i.e. neighbour nodes having hop counts smaller than the sending node) decreases under a certain threshold, this module notifies the transmission range expansion module to extend the transmission range of the sending node to include new neighbours in its vicinity (Rizal et al., 2012).

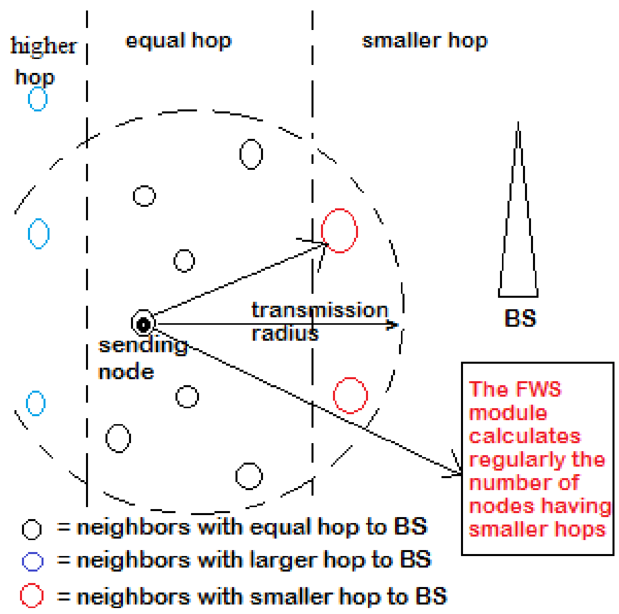

Figure 4.7: The initial procedure of the Forwarding neighbourhood Size module

Based on the discussion in the previous subsection, let N_i, N_{eq}, N_h and N_l respectively denote the type of neighbours of the sending node j as the nodes having equal hop, the nodes with higher hop and the nodes that have lower hops to the BS. Thus the number(size) of $N_i > N_{eq} > N_h$ or N_l. If it is assumed that the distance from the sending node j to any node in N_h is $\frac{1}{2}$C (i.e. C is a constant), to any node in N_{eq} is C and to any node in N_l is 2C, and the distance between the BS and nodes in N_h is LC, distance to nodes in N_{eq} is (L-1)*C and distance to nodes in N_l is (L-2)*C, then the time taken to traverse packets from the sending node through any neighbour from different nodes is as follows:

$t_{(N_h-BS)} = \frac{L.C}{(\frac{1}{2}C)} =$ 2L (4.27)

$t_{(N_{eq}-BS)} = \frac{(L-1)*C}{(C)} =$ L-1 (4.28)

$t_{(N_l-BS)} = \frac{(L-2)*C}{(2C)} = \frac{L-2}{2}$ (4.29)

From the equations (4.27 - 4.29), it is proven that the theorem of relaying packets through nodes that are closer to the destination can shorten the transmission latency. However, including higher node participation can cause the network more dense thus encountering the higher possibility of overhearing node. Suppose that the likelihood to encounter an overhearing for any node in a certain region (N_{eq}) with the traffic load (M_n) bytes data is $\triangle$. Thus, with the help of equations (4.6 and 4.11), I can determine the likelihood of overhearing as follows.

Likelihood of overhearing (any node in N_h) = $\frac{1}{3}\triangle M_n$ (4.30)

Likelihood of overhearing (any node in N_l) = $3\triangle M_n$ (4.32)

From the above equation(s), it is justified that the likelihood of nodes to encounter overhearing will increase amongst the typical nodes with lower/smaller hop to the BS (N_l). Thus maintaining a certain number of the neighbouring nodes having lower hop for data forwarder is required to address the inefficiency of routing due to excessive amount of node overhearing. This certain number should be simulated in order to find the optimum results with respect to the QoS performances while considering energy efficiency. The next subsection explains the technique for the proposed adjustable transmission range extension approach.

4.3.3 Adjustable transmission range extension (ATRE) module

In the network model described earlier, the sensor network consists of homogeneous sensors i.e. having equal role and capacity in terms of computation, communication, and power. The nodes are deployed in a non-deterministic manner in order to support many applications in WSNs, which commonly are developed in unattended operations.

As described previously, an expansion of neighbour (connectivity) degree can balance energy distribution by reducing transmission hop-count, and the transmission range can control the neighbourhood size. Let N be a set of nodes in a 2D Euclidian Space, and N_l: dist$N_l \leq R_i$, where N_j is the number of neighbours of node j within the initial transmission range, dist represents the distance between two nodes i.e. node 'j' and any neighbour ('k'). The dist is defined as a real number that represents the radius of two corresponding coordinate points (x_1, y_1) and (x_2, y_2). Thus,

$dist_{jk}$ is always > 0 (4.32)

$dist_{jk} \equiv dist_{kj}$ (4.33)

$dist_{jk} + dist_{kl} \equiv dist_{jl}$ (4.34)

A node k is determined as a neighbour of node j, if $dist_{jk}$ satisfies this formula:

$dist_{jk} \leq R_i$ (4.35)

According to formula 4.31, I can define the neighbour degree of node j, ($D_{neigh-j}$) as:

$\forall$ neighbour $\in D_{neigh-j}$ and $dist_{j-neighbours} \leq R_i$ (4.36)

In order to define the set of nodes 'F_j', termed as the forwarding nodes of N_l, whose task is to relay packets to the BS from a node j, I formulate the $dist_{neighBS}$ as follows.

$\forall$ neighbour $\in F_j$, $dist_{j-neighbours} \leq R_i$ (4.37)

$dist_{j-BS} > dist_{neighbours-BS}$ (4.38)

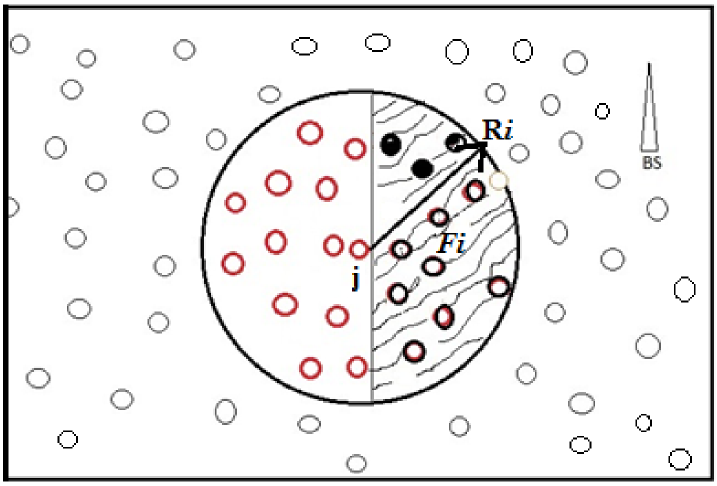

Figure 4.8: A node j with its set of forwarding neighbour, (F_j) and a set of non forwarding neighbour

Figure 4.8 plots the set of forwarding nodes of N_l. Every node inside the circle is an N_l' s neighbour where the distances between nodes in the circle are always below than the transmission range, R_i. Nodes in the shaded area of the circle are the forwarding nodes of N_l as their distances to the BS are lower than the distance between node j and the BS. In relation with the scenario in Figure 4.8, set of nodes in F_j can be used as the forwarding nodes thus, node j can select one of them for relaying packets to the BS according to the used routing algorithm.

In case of a routing algorithm that is based on differentiating QoS requirements, the selection of a forwarder node take into account the specific requirement for the type of data traffic as explained in Chapter 3. For all nodes in F_j, ($\forall$ neighbour $\in F_j$, $dist_{j-neighbours} \leq R_i$), the distances between the nodes in F_j and sensor j is determined as:

$dist_{j-BS} < dist_{neighbours-BS} = \varsigma$ (4.39)

$Max(\varsigma)$ = (largest) $dist_{neighbours-BS}$ (4.40)

$Min(\varsigma)$ = (closest) $dist_{neighbours-BS}$ (4.41)

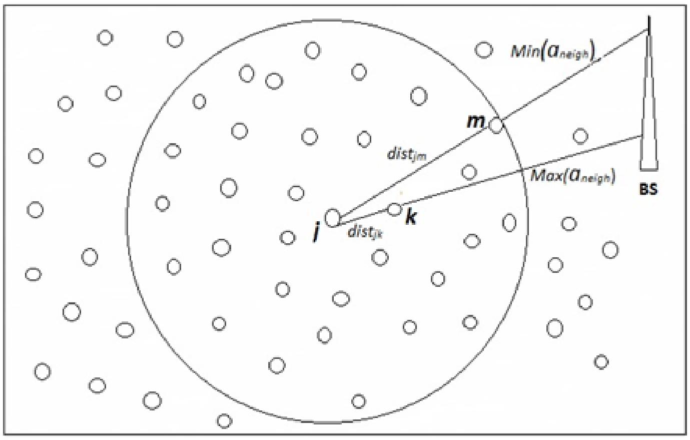

Figure 4.9: The distance calculation of the neighbour using a MaxMin Constanta

According to equations 4.40 and 4.41, when a real-time data has to be transmitted, node j attempts to find a neighbour in F_j, which has the closest distance to the BS in order to meet the given deadline for the real-time traffic as depicted in Figure 4.9. Thus, the node m will be selected as the next hop node for relaying bounded-delay packets as node m has $\text{Min}(\varsigma)_{neighbours}$. On the other hand, to deliver non real-time data, which does not have specific requirement in terms of transmission delay, the routing algorithm may pick the node k having $\text{Max}(\varsigma)_{neighbours}$ so that the algorithm can maximise the participation of each node in F_j for delivering a certain type of traffic. The following sub-section describes the process of the route setup of the proposed adjustable transmission range.

4.3.4 The routing setup for the proposed scheme

After a random node deployment, each sensor is assumed to be static and know its geographic (coordinate) information using some localisation methods. To initiate the network configuration, a Base Station (BS) broadcasts a sink message through the whole network in order to obtain the locations (hop counts) of all nodes. The locations of nodes according to the BS are then determined, and changed regularly to get the updated information about the nodes in term of their positions (i.e. hops and distances to the BS). Each sensor node knows about the neighbours in its vicinity.

The next step of the routing setup is to determine a threshold value, termed as 'T_{neigh}' for each sensor node according to its neighbourhood size for maintaining the data delivery. The calculation of the threshold is as follows.

If $S_{neigh} < T_{neigh}$, then an increase transmission range process begins by incrementing the transmission range every 10 meters (Xiao Hui et al., 2011) until $S_{neigh} \geq T_{neigh}$,

If $S_{neigh} \geq T_{neigh}$, then the initial transmission range is maintained.

Varying the transmission range according to S_{neigh} of a node (x), affects both the radio coverage and energy consumption of each node. A high transmission range certainly costs high power consumed per node thus, the increase of radius should consider the energy of nodes as well as network density and neighbourhood size. These factors are important criteria for representing the effectiveness of setting up the T_{neigh}. Considering energy efficiency, the threshold T_{neigh} is defined as the number of forwarding neighbours, whose locations are one-hop less to the BS. This means that the increasing transmission range continues until the sending node finds a certain number of neighbours that meet the T_{neigh}.

This aims to improve energy saving in the nodes, which suffer from frequent changes of the transmission radius according to the S_{neigh} of a node (x).

Let us assume, N_l : dist$N_l \leq R_i$, N_l is the number of neighbours of node l within the current transmission range.

$$N_l = \sum_{x=1}^{x=T_{neigh}} F_x^l \quad (4.42)$$

whereas x = class services, F_x^l is the forwarding node with different traffic classes for the sender node l. T_{neigh} is the threshold of forwarding neighbour membership, if $N_l < T_{neigh}$, the algorithm then increases the transmission range until finding optimum neighbours, i.e. $M_l \geq T_{neigh}$, M_l : dist$M_l \leq P^i$range, whereas M_l is the number of new neighbours after adjusting the transmission power, and P^irange is the new transmission range.

To alleviate excessive energy consumption due to the increase of power transmission, the proposed scheme selects neighbours having equal hop to forward non critical packets such as C3 and C4. F_x^l is the previous forwarding nodes that have lower or equal hop count to the sender and P_x^l is the new forwarding node with lower hop count (closer to the BS). Hence the equation formula changes as follows.

$$(N_l) = \sum_{x=1}^{x=n}(F_x^l) + \sum_{x=n+1}^{x=T_{neigh}}(P_x^l) \quad (4.43)$$

When the node finds its number of neighbours, below the threshold, the node increases its range of transmission power and broadcasts a Hello packet over the same link (using a bi-directional link) to update its neighbourhood information. Each node within the new transmission radius will receive the Hello message, update its table entry with new neighbour's information, and sends back a message including its own information including energy, hop delay, distance and history of channel access and ACK (acknowledgement message) to the original node that broadcasts the Hello packet. Before transmitting these messages, the neighbour checks the availability of the channel. If the channel is clear, then the packets are sent, otherwise the neighbour node waits for a randomly chosen period of time, and then checks again to see if the channel is idle. This period of time is called the backoff factor, and is counted down by a backoff counter based on CSMA-CA MAC protocol that the algorithm uses.

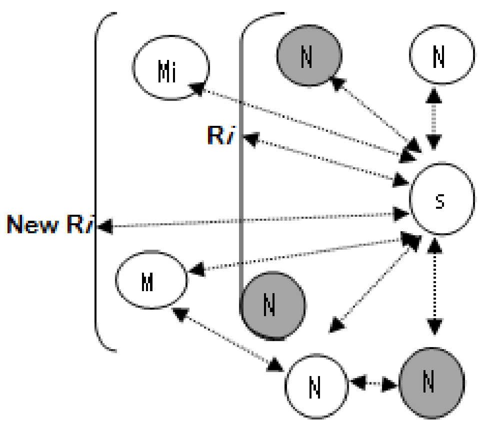

Figure 4.10: A new transmission range for a new neighbourhood size

After the Hello exchange between two neighbours is completed, the process continues to other neighbours until all neighbours update the new neighbourhood information. Thus, the size of neighbourhood increases to include enough number of forwarding neighbours.

Figure 4.10 shows a scenario. Every node can adjust its transmission range and update its new neighbourhood information when the number of forwarding neighbours is below the threshold. Nodes with grey color are failed nodes due to energy depletions and S is a sender node. The benefit using the proposed approach is to reduce the existence of the energy hole around the BS. Every node in the network attempts to maintain the size of neighbourhood by extending its transmission range when its node degree decreases under a threshold. Therefore, the data transmission can be continued through new neighbours that have closer distances (lower hops) to the BS. Nodes with enough forwarding neighbours will maintain their initial range, and relay packets using more-hopping to reach the BS as the initial range is shorter than the adjustable range. The flowchart of the scheme is shown in Figure 4.10.

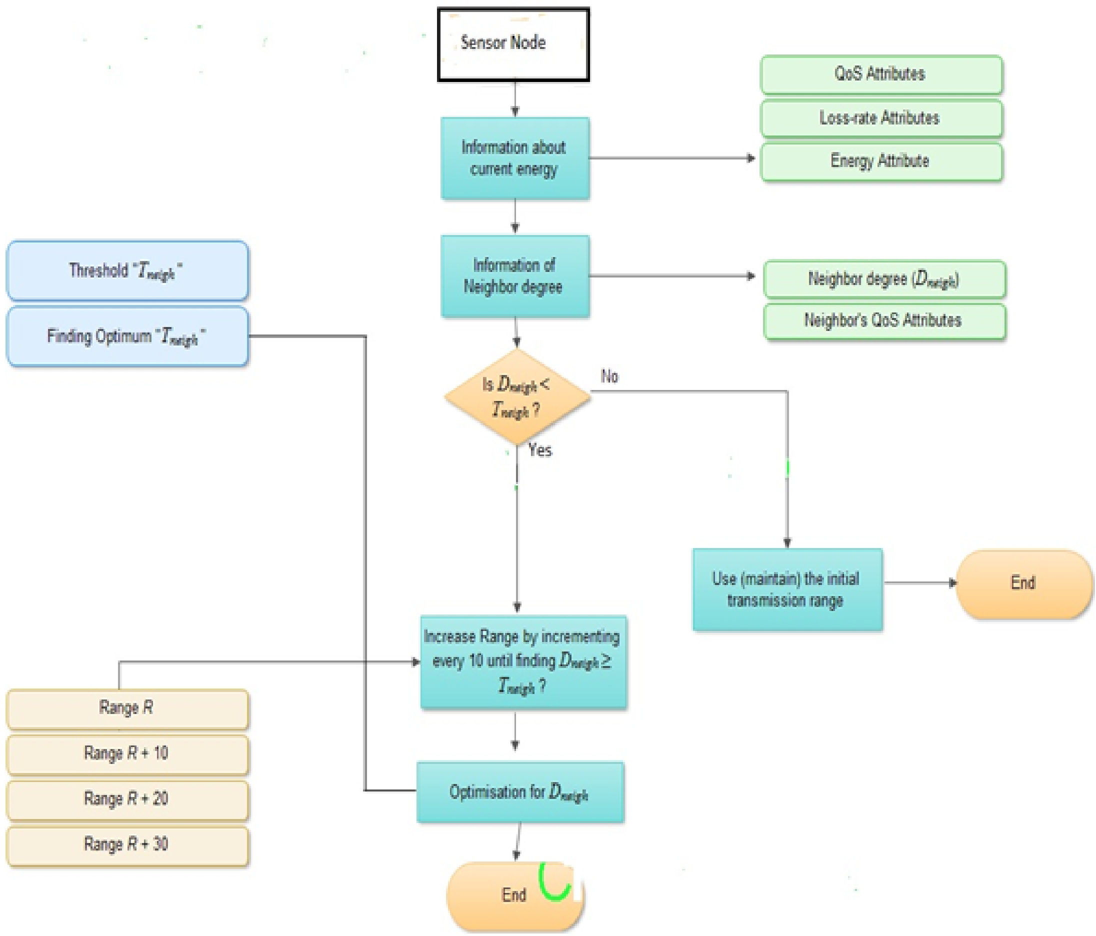

Figure 4.11: The flowchart of the ATRE module

Once I obtain the proper radius of transmission, which suits the the neighbourhood size limit then I use the routing algorithm as described in Chapter 3 by integrating the fitness routing approach into the proposed approach as described in the figure 4.6. The following section examines comparative performance between the original scheme and the enhancement approach.

4.4 Performance Evaluation

In order to evaluate the functional behavior and the performance of the routing schemes, I have developed the simulation model in order to ensure the schemes can be simulated in the experimental works. The aim of this section is to find the advantages of the implementation of non-uniform and adjustable transmission range strategy based on neighbourhood size in the original QoS differentiating routing (TeGaR) by comparing the performance of TReNs with TeGaR and SPEED (Tian et al., 2003), a prominent QoS routing algorithm that exploits geographical information routing and QoS notion We first established a set of evaluation metrics that would be used as a benchmark to the quantitative analysis for the comparative evaluation of the schemes. The proposed scheme (TReNs) is the first enhancement to the original scheme (TeGaR), which did not have support to a non-uniform transmission range extension/adjustable approach. The details of the evaluation metrics and the simulation environment and parameters are described below.

4.4.1 Evaluation Metrics

To measure the quality of the schemes according to some criteria that are required in WSNs, I define some metrics for quantifying the performances of the QoS routing schemes. These metrics can characterise the properties of an entire WSNs and measure the effectiveness of the routing schemes to certain application specific needs.

- *Average energy consumption* of sensor network: this metric gives a good measure of energy consumed in the network. This metric also is able to represent the unused energy of the network when the first node fails due to energy depletion. To evaluate the routing algorithms according to energy efficiency, I examine the number of alive sensors and the unused energy during simulations.

- *Network Throughput*: this metric is defined as the average rate of successful packet transmission from a pair of source-destination. This metric is defined as the total data capacity (number of data packets received) at the BS.

- *Network Delivery Ratio*: this metric is defined as the ratio of the total data received by the BS and the total data sent by the source node. This reflects both the effectiveness and the reliability of the routing schemes in maintaining the network function.

- *Average delay per packet*: this metric is defined as the average time taken to transmit packets from the source to the destination (BS). It represents the response time for the routing schemes in handling the required data-related to each type of traffic.

4.4.2 Simulation Environment

To characterise the variables and constants that affect the performance of the routing schemes, I determine the parameters and environments that can be used in the simulation works for implementing the set of evaluation criteria described above. The results of the simulation will be denoted as the function of these parameters that perform comparative evaluation between the proposed enhancement scheme and the established routing algorithms (TeGaR and SPEED).

To implement the proposed TReNs, I have built two modules. They are a) forwarding neighbourhood size module and b) transmission range expansion module. These two modules are added into the current tailor-made simulator as described in Chapter 3. The first module introduces new features that if the algorithm records nodes having number of forwarding neighbours decreases under a certain threshold, this module notifies the transmission range expansion module to increase their radius of transmission. Furthermore, by receiving this notification message, the module of transmission range expansion extend the node's transmission range until it has a number of neighbours that meets the defined threshold.

The environments of simulation are developed in two dimensional Cartesian coordinates. Total 300 nodes are deployed randomly in the target field onto 300 x 300 m^2 grid, and random multiple events from 10 source nodes are created and exchanged with various rates. As described in Chapter 3, the simulator exploits carrier sense multiple access collision avoidance (CSMA-CA) (Kleinrock and Tobagi, 1975) for sensing the channel access with a bandwidth 1 Mb/s (Kulik et al., 2002). The simulator uses a set of parameters that were introduced in (Kohvakka et al., 2006) according to sensors' activities, and all parameters have been established for the first simulation in Chapter 3. In addition, the simulations are carried out based on the International Telecommunication Unit (ITU) G.114 standard, the following table lists the environments of simulation.

4.5 Analysis of comparative simulation results

Extensive simulation experiments were conducted to quantify the effectiveness of the schemes, and to demonstrate the protocol reliability, latency, energy efficiency, and throughput. The results of all experiments are described in the following sections.

These sections present two comparative evaluations: (i) the evaluation based on different neighbourhood size degree that is used in the first enhancement approach and, (ii) the performance evaluation of all schemes in terms of QoS requirements. The first comparison is to evaluate the optimum degree of neighbourhood size that can assure energy efficiency. While the second comparison is to evaluate the performance between the schemes with respect to QoS satisfactions. The following sub-section provides further detail of the comparisons.

4.5.1 Neighbourhood Size Comparison

This section investigates the impact of the usage of different neighbourhood sizes that can be used to achieve route efficiency. The evaluation of neighbourhood sizes aims to achieve maximum data capacity in the network proportional to energy efficiency. The evaluation expresses the tradeoff between different transmission radii i.e. small transmission range (many hops) and large transmission radii (low hops but less interference). Varying transmission radii have an impact on the average degree of network connectivity, and further affects the network performances. A lower node degree causes a drastic reduction in data capacity. On the other hand, exceeding an optimum degree of network results in the gradual degradation of network throughput caused by interferences.

In order to achieve energy efficiency while satisfying QoS performances, the following figures validate the effect of different neighbourhood sizes through simulations.

Figure 4.12 shows the different performances of network data capacity under heavy congestion (25 packet per second) where varying level of forwarding node degree can be used as a threshold for the proposed scheme in increasing its transmission range. The best data capacity is achieved when each node can maintain four neighbours, whose hops are smaller than the node. This number represents the optimum degree of forwarding node for high level of connection. Because a larger forwarding node degree (five nodes as plotted in Figure 4.12) causes more interference and packet collisions (transmission overhearing), resulting in wasting energy. But, a smaller degree tends to reduce the possibility for nodes being connected with each other. Therefore, Figure 4.12 justifies that a high packet reception can be achieved by the three or four of forwarding node degree.

The next experiment measures energy consumption of nodes under different forwarding node degrees according to unused energy, rate of node failure and network lifetime. Figure 4.13 shows the network lifetime, which is defined as the time by the simulation when the first node dies with the five forwarding node degrees. We can see that when a node still has

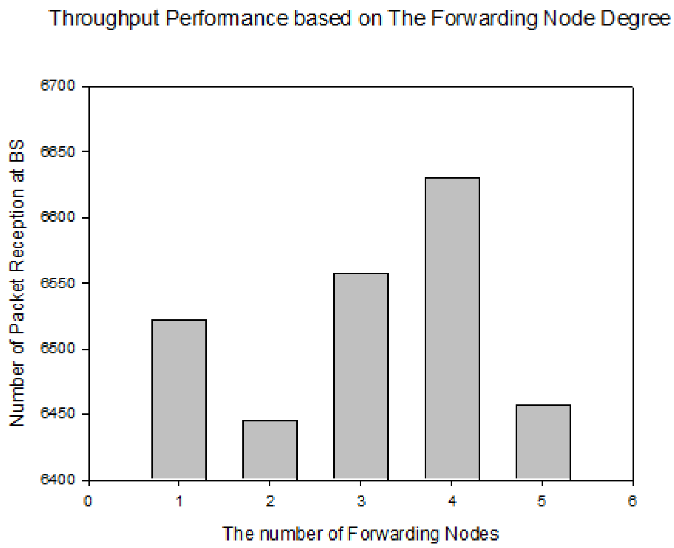

Figure 4.12: The impact of varying forwarding node degree to the network data capacity

three or four forwarding neighbours in its vicinity, a balanced data distribution is achieved, and the longer time for the first node to leave the network due to energy depletion. The lifetime of network decreases when the number of the forwarding node is reduced. This is because fewer forwarding nodes will depend on one neighbour to relay the excessive amount of traffic (high load) thus, the neighbour will deplete its energy soon. In contrast, when the forwarding node degree is too high which involves a larger transmission range, the neighbours are engaged with coping with the interference, which in turn affects the lifetime of nodes.

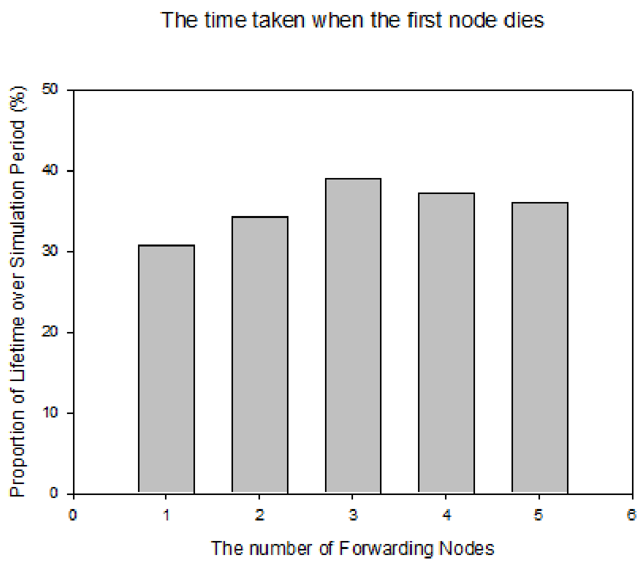

Figure 4.13: The network lifetime under varying forwarding node degrees

Note that the neighbourhood sizes from two to four forwarding nodes perform better in terms of energy efficiency than those with other node degrees. The average unused energy is defined as the left over energy and it remains till the end of the simulation, when the routing algorithm fails to use those nodes for data delivery. Figure 4.13 depicts while the size of forwarding node degree is decreasing the energy that is left in the network is increasing because data is relayed to fewer neighbours and the other nodes become inactive. This reduces the utilisation of the network. We also observe that the energy of nodes being unused tends to decrease as the degree of connectivity increases. By increasing the range of transmission, the expansion of the forwarding node (neighbour) degree includes a higher number of nodes being connected for routing packets. Figure 4.14 again demonstrates that the four forwarding node degree provides an ideal low rate of the unused energy. This number can be used to represent the effectiveness of the proposed enhancement approach in terms of energy efficiency.

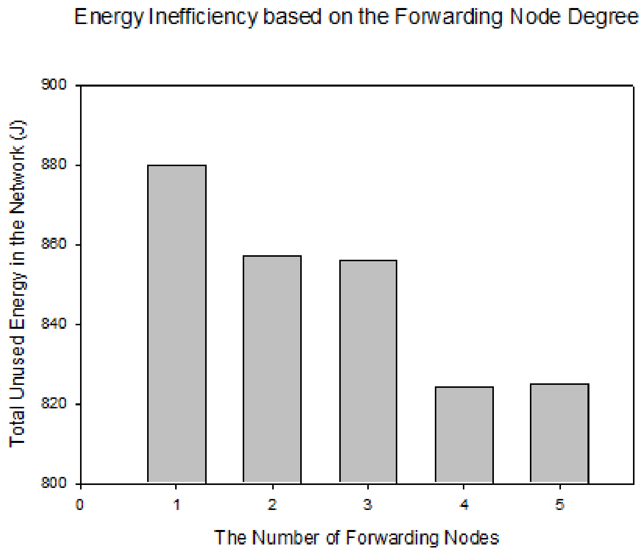

Figure 4.14: The rate of unused energy (J) under various forwarding node degrees

Based on the experimental results of the above figures, it can be concluded that the optimum forwarding node degree i.e. the four node degree shows the effectiveness of the proposed scheme in mitigating the problem of imbalance energy consumption rates in different parts of a sensor network, and thus achieve consistent lifetime across the network.

The simulation results verify that the maximised data capacity at the BS is achieved when a node can maintain its forwarding nodes to four sensors. By using this degree, the proposed scheme provides some benefits according to QoS metrics such as network throughput, energy efficiency and network lifetime as described in Figures 4.12 - 4.14. Thus, the number of four forwarding nodes presents a proper value for the threshold of the adjustable transmission range mechanism, and used by the proposed first enhancement to the TeGaR scheme to compare with the original scheme of TeGaR and SPEED as described in the following sections.

4.5.2 Comparative Evaluation of the Schemes

In order to examine the effectiveness of the extended proposed approach that offers a transmission range extension based on forwarding node degree, extensive simulation experiments were conducted to compare the performance of the scheme with the original scheme (TeGaR) and SPEED. The reason why these two protocols were chosen is that (i) to verify whether the proposed enhancement approach provides better performance than the original one in mitigating the 'energy hole' problem caused by an uneven load distribution, (ii) to compare to a well known QoS routing scheme SPEED, which exploits a backpressure mechanism and a neighbour feedback loop (NFL) approach (He et al., 2003) for congestion avoidance and balanced load distribution. The comparison was performed in terms of the following aspects.

1. Network throughput is defined as the total data capacity at the BS

2. Number of drained nodes over time period. A drained node is a node, which fails maintaining its energy above the threshold due to energy depletion.

3. The network lifetime is measured based on the time when the first node fails and starts to leave the network

4. The residual energy is represented by the energy level that successfully being maintained by nodes in the sensor network over the simulation time

5. Latency is defined as the time needed for a packet to travel from a source to a destination node

Packet reception evaluation

In order to see the reliability of the algorithms under different traffic conditions, I monitored how the changes of congestion rates affect the network throughput (data capacity at the BS). As overall, the first enhancement approach has a higher throughput than the two corresponding schemes. The data capacity at the BS, identified as the network throughput, increases when the load injected into the network also increases.

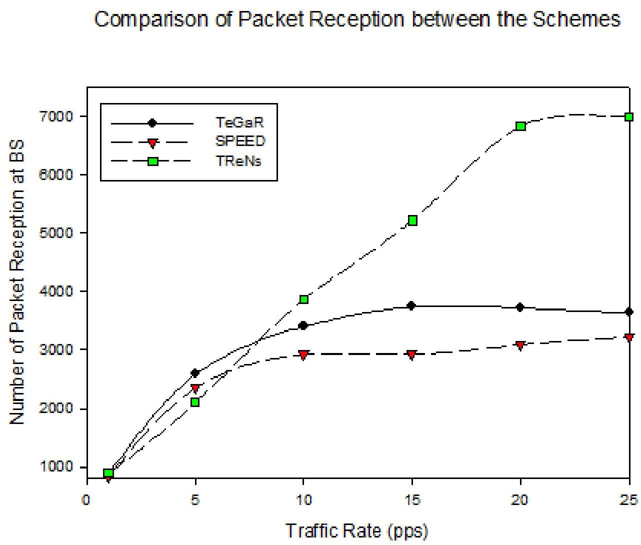

Figure 4.15: The performance comparison of the throughput between all schemes (K$_{bits}$)

The rate of packet reception increases significantly until reaching a peak at around 6998 Kb for the adaptation scheme, 3150 Kb for the original scheme (TeGaR) and 3050 for SPEED (see Figure 4.15). At the highest traffic rate (25 pps., i.e. packet per-second), the TReNs scheme can maintain the ratio of packet delivery more than 60% of the total packets sent from the source nodes, while the other schemes drop the data capacity under 30% of their throughput. The decline of the average throughput is the reflection of the network dysfunction caused by the large number of nodes around the BS that increase energy consumption (see Figures 4.16, 4.17 and 4.18). All remaining transmitted packets fail to reach the BS thus resulting in high loss of packets. Results in Figure 3.18 and Figure 4.15 show a similar rate/trend of throughput for TeGaR and SPEED.

Figure 4.16 demonstrate that the first enhancement approach (TReNs) deals with the problem of the 'energy hole' better and more efficiently compared to other two schemes. The 'energy hole problem' occurs with a higher degree of failing nodes when the TeGaR scheme is used because TeGaR does not perform a balanced load distribution technique as described in Figure 4.17. While, SPEED that exploits a backpressure mechanism is able

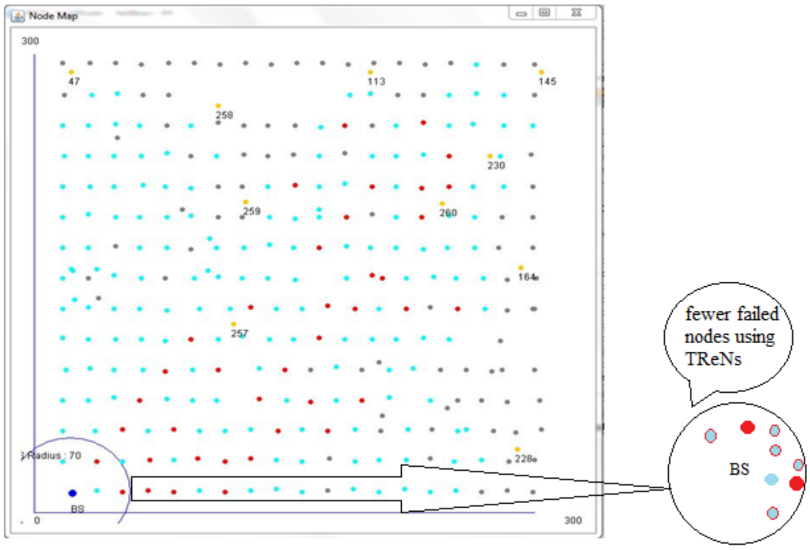

Figure 4.16: A fewer drained nodes in TReNs can avoid the 'energy hole problem'

to reduce the existence of the 'energy hole' as SPEED can decrease its transmission rate when a congestion occurs.

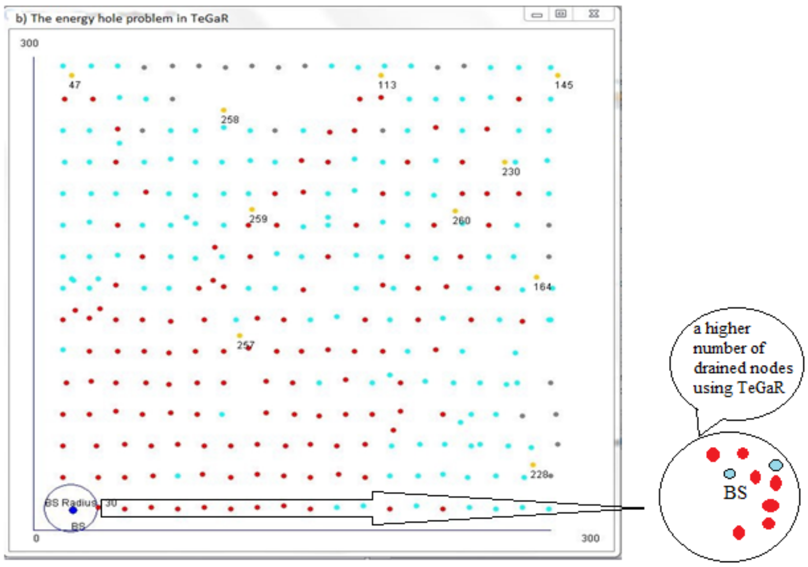

Figure 4.17: A higher number of drained nodes in TeGaR resulting in the 'energy hole'

In addition, it is also clear that nodes (represented in grey dots) in the edge of the network (outermost region from the BS) have a higher level of remaining energy, whereas all the other nodes (blue dots) nearly use up their energy. This is in accordance with the previous analysis that the nodes farther from the BS just relay packets that are generated from themselves. In contrast, nodes, whose locations are closer to the BS have tasks to relay packets from nodes in outer region.

Evaluation of energy consumption per node

In this section, I discuss experiments that evaluate the energy consumed per node by determining the total number of nodes being failed due to energy depletion over the simulation time.

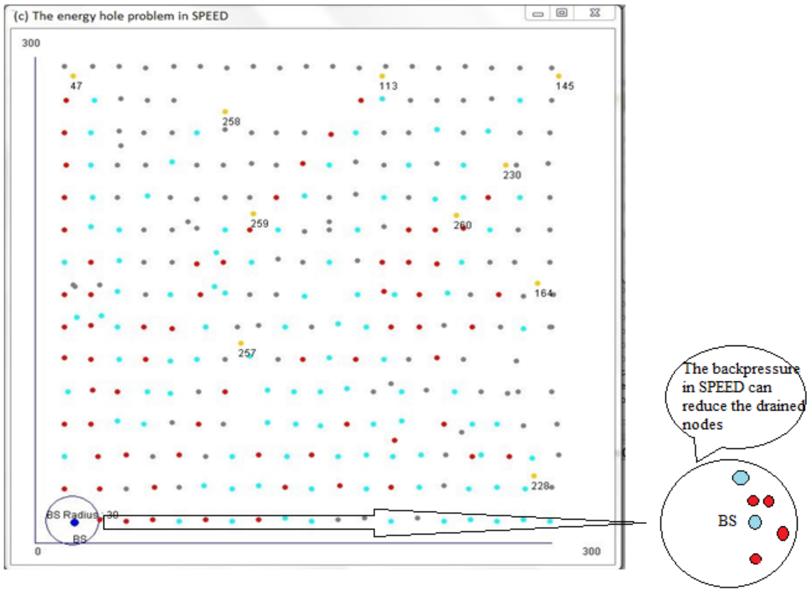

Figure 4.18: The failed nodes in SPEED results in the 'energy hole problem'

Figure 4.19 shows the increase of node failures with the traffic rate. The increase of throughput often affects the energy consumed per node, which causes the node to fail due to high consumption of energy. Note that the enhancement approach (TReNs) achieves the lowest rate for failing nodes even when its data capacity increases twice compared to TeGaR and SPEED (see Figure 4.19). For example, under heavy congestion (25 pps), the proportion of dead nodes is less than 15% of all nodes in the network, while SPEED provides about 25% and the TeGaR scheme results in nearly half of the nodes failing in maintaining the network connectivity. This means that TReNs is more efficient in using the energy for all the operations including sensing, computing or communicating. When the forwarding node degree decreases, a node reacts by increasing its transmission range properly so that the nodes being involved can be used effectively to relay packets. Thus, the data capacity is increased, while the energy expenditure is saved.

On the contrary, since the original scheme (TeGaR) maintains a higher throughput than SPEED by involving a higher number of nodes participating in data delivery, especially when the network becomes congested, it fails to reduce inactive nodes due to energy depletion. The decrease for maintaining the active nodes can result in some packets not being able to reach their BS (see Figure 4.15). On the other hand, at a similar situation when a congestion occurs, the SPEED's backpressure approach that aims to reduce its transmission rate, can decrease the load of traffic. Thus, a lower level of packet-relay saves energy for transmission and reception, therefore, the SPEED scheme decreases the number of node failures as well as the throughput (see Figure 4.19).

As shown in Figure 4.20, the energy consumed per node using the TReNs is less than energy consumption level using the other two schemes. Thus, the time taken for the first node to die is longer compared to results of the other two schemes as depicted in Figure 4.20.

The adjustable transmission range approach is able to be scalable enough to respond to any changes according to neighbourhood size (topology) thus it can prolong the lifetime of nodes. Figure 4.20 shows the network lifetime that is defined based on the time when the first node running out of its energy is double compared to the two corresponding algorithms, and more than three times when the traffic becomes heavily congested. Under the light congestion, on SPEED and TeGaR, the network lifetime is to half (approximately) to the whole simulation time, and the lifetime decreases to about 4% for SPEED and about

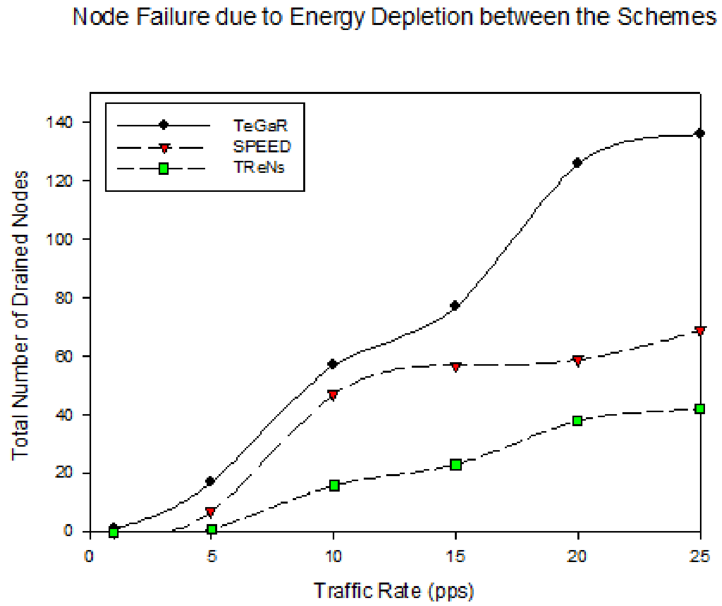

Figure 4.19: Average number of failed nodes between all schemes

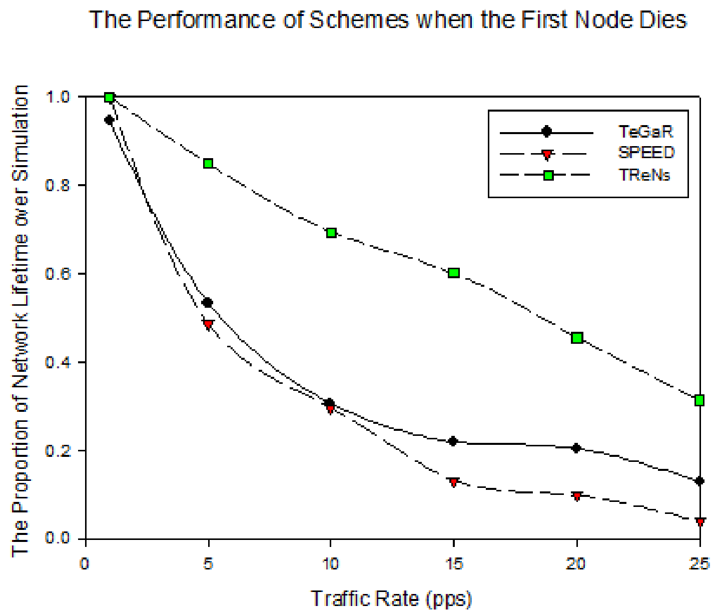

Figure 4.20: Average time taken when the first node of the schemes is broken

13% for TeGaR under the highest congestion rate. In contrast, using the first enhancement approach, when the network is not too congested, the first node dies after the simulation runs for about 85% and more than 30% under heavy congestion. However, this lifetime decreases significantly for all the schemes when the traffic rate increases. The longest time is achieved by the first enhancement approach (TReNs), then followed by TeGaR then by SPEED.

According to Figures 4.15 and 4.20, these figures show that the longer lifetime can be achieved by the scheme, which can affect significantly to throughput (packet reception). Figure 4.14 demonstrates that the TReNs, which has longer lifetime can perform better in maximising its throughput compared to the other two schemes.

We also examined the residual energy of all nodes with different traffic rates in Figure 4.21. The figure shows that the total residual energy in the network, decreases when the traffic rate increases, and a rapid decrease occurs for nodes in the original scheme (TeGaR). Figure 4.21 shows that the first enhancement approach (TReNs) performs the best in terms of energy efficiency as it provides a balanced power distribution by increasing its transmission range when the forwarding node degree drops to the limit (threshold). Thus, the nodes around the BS do not have to relay an excessive amount of traffic thus, the nodes are able to maintain their energy until the simulation ends.

However, when the network traffic becomes congested, some nodes die (fail) due to energy depletion, but those nodes are distributed over the network, and are not uniformly located at the BS's vicinity as shown in Figure 4.16. Nonetheless, other active nodes are capable of maintaining their energy exceeding the threshold level so that the data capacity can be improved compared with TeGaR and SPEED (see Figure 4.15).

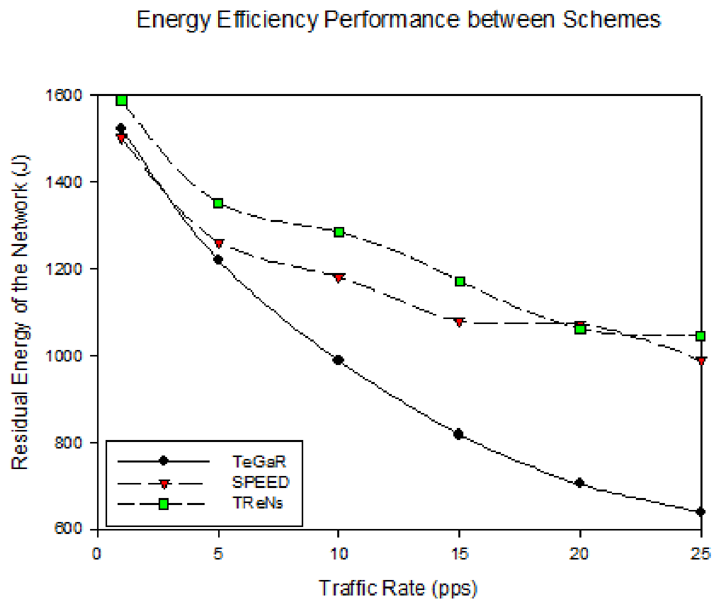

Figure 4.21: Total remaining energy of all nodes in the network when the simulation ends

The residual energy ratio of network using the TReNs is nearly twice compared to the original scheme, and relatively higher than SPEED traffic rates ar (i.e. 20 pps and 25 pps). The difference ratio becomes smaller when the traffic rates decrease (see Figure 4.21). This demonstrates the effectiveness of the first enhancement (TReNs) strategy in mitigating the 'energy hole problem' compared to TeGaR and SPEED, which uses a backpressure mechanism for a congestion avoidance strategy.

Packet delay evaluation

In this experiment, I present the performance results of end-to-end delay obtained by the simulation for all the schemes (TeGaR, TReNs and SPEED). Figure 4.22 shows the performance of the schemes using average end-to-end delay of packets. The average delay tends to increase when the traffic rate is increasing because more packets being forwarded will incur longer delay for packet queuing at intermediate nodes.

As the TeGaR uses a Fitness Function combined with a Priority based Delivery mechanism, TeGaR provides shorter delay than SPEED, which may reduce its transmission rate when a congestion occur. However, the best performance according to the average transmission delay is achieved by the TReNs as this approach has an ability to increase its transmission range for certain circumstances thus shortening en-route data to the BS.

Figure 4.22: Average delay transmission under different traffic rates

4.6 Summary

This chapter investigated analysis of the uneven energy depletion phenomenon noticed in many to one data communication networks, and introduced the ways to address the problem of 'uneven energy depletion'. The TReNs is the first enhancement approach to the original scheme (TeGaR) that introduces a complex routing to create a new proposal, which combines an adjustable transmission range extension based on the neighbourhood size with the fitness function that considers node's quality to enable different QoS paths for different QoS requirements

The proposed enhancement approach (TReNs) aims to reduce the possibility that the 'energy hole problem' exists, especially in the part of network, which is close to the BS. The strength of the new proposed approach is to provide a balanced power distribution technique that can reduce the 'energy hole' problem around the BS. The algorithm is designed to increase its transmission range when its forwarding neighbourhood size decreases below a predefined limit (threshold). By doing this, at a certain circumstance, using a new increased transmission radii, nodes in the farther part of the network may transmit packets either directly to the BS or to the nodes that are one-hop away from the destination. Thus, nodes, whose locations are close to the BS do not have to relay an excessive amount of traffic in order to reduce the occurrence of 'energy hole' problem.

The simulations demonstrated that the proposed adaptation scheme is able to decrease the existence of 'energy hole' problem around the BS, and further avoid an early network dysfunction. Thus, the data capacity can be improved while the energy of nodes are efficiently used. This chapter described that both the network and energy models, which identify the key factors contributing to the energy hole problem is well validated through simulations by using the enhancement approach. The new proposal of a adjustable transmission range strategy based on a forwarding neighbourhood size is able to evenly distribute energy consumption in the network, thus, many nodes can maintain their energy better than the original scheme (TeGaR), which uses a uniform transmission radii. The experimental evaluation have also shown that the proposed enhancement approach performs the best compared to TeGaR and the well known QoS scheme named SPEED in terms of some essential key factors in WSNs such as throughput, end-to-end delay, energy efficiency and network life time under different congestion levels and scenarios. Hence, the evaluation results reveal that the first enhancement approach (TReNs) can address the issue of 'energy hole problem' that has been highlighted in Chapter 3.

Based on the experiments, even the original scheme (TeGaR) can improve the network utilisation by involving a high number of nodes to be participated in data delivery, both the maximised throughput and energy efficiency cannot be improved. The TeGaR may fail deliver packets effectively, and encounter packet detouring as a result of an improper area for packet routing. The next chapter discusses and proposes the second enhancement approach that aims to address this issue of broadcast flooding by controlling the area (region) of routing for individual communication sessions.

Chapter 5

Self-Adaptive Routing Region Algorithm

The previous chapter described the first adaptation scheme that aims to reduce the existence of the ‘energy hole’ problem in the network using a transmission range expansion based on the neighbourhood size approach. This chapter introduces the next enhancement to the TeGAR scheme for decreasing the occurrence of packet detouring or unguided transmission as a result of an improper area for packet routing.

5.1 Introduction

The preceding chapters presented the TeGaR scheme for WSN's applications having different types of data traffic based on differentiating QoS requirements, and the first enhancement approach that is a non-uniform transmission range extension based on the neighbourhood size.

This chapter discusses the second enhancement of the proposed differentiating QoS routing algorithm, and this enhancement aims to improve route optimisation and restrict unguided relaying/transmission due to improper area of routings, which wastes energy and increases transmission delay in WSNs. The notion of proposing certain areas of routing termed routing region is primarily motivated by the idea that, if the nodes fail to handle route discovery flooding effectively in a certain zone, they encounter higher routing overhead as well as packet loss thereby wasting the energy of nodes in the network. Adding location-awareness in guiding the routing, leads to improved route optimisation that would affect the throughput, and further reduces energy consumption by decreasing unnecessary retransmissions caused by packet looping that induces packet loss. In addition, guiding packet routing with the help of request zone techniques also eliminates an unnecessarily long transmission and thus decreases the time taken for packets to arrive at the BS. In order to ensure these advantages, I propose a scheme to achieve effective route discovery for data routing using request zone approach based location-aided routing techniques.

In WSNs, different location-aided routing algorithms have been used (Xiao Hui et al., 2011; Ko and Vaidya, 2000; Basagni et al., 1998). They commonly use the location information of nodes for creating data paths and limit the area of routing to a geographic area around the destination. Location based routing algorithms, are less complicated to their practical design and implementation compared to hierarchical routing protocols that requires additional complexity for performing clustering techniques and cluster head selections. Further, the routing algorithm using the location awareness is more efficient from the perspective of energy consumption than flat-based routing protocols as the location based techniques are able to save energy by forcing as many as nodes go to sleep mode while they do not have any activity. Thus, the information of nodes‘ location is used to confine route discovery as well as data routing and maintenance in a specific zone, instead of transmitting blindly for a route in the entire network. Unlike the existing location-aided routing protocols, the proposed scheme provides a centralised routing region adaptation using a self-learning process to make transmission more directed toward the destination in an efficient manner. This approach avoids overhead costs at a node for maintaining a routing table to destination, eliminates unnecessary frequent adaptation that waste energy. Further computation of the restoration region is more easily handled.

In doing so, this chapter proposes to exploit such request zone techniques based on location-aided routing to create routing regions for individual communication sessions by optimally re-positioning the region boundaries. The aims of this chapter is to design a novel scheme, named Rule-based Learning Adaptation Approach for Routing Regions (RuLeARR) i) to define a routing region that limits data routing in a specific zone to achieve both energy efficiency and network performances; (2) to perform self-adaptation to the routing region due to frequent changes in WSNs; and (3) to integrate this second enhancement approach to the proposed differentiating routing algorithm (TeGaR) to support heterogeneous traffic.

The rest of this chapter is organised as follows. Section 5.2 describes the routing region techniques which would be used for the proposed self-adaptive routing region scheme. Section 5.3 discusses the routing algorithm of the proposed routing region (RuLeARR) that includes the realisation of self-adaptation using a rule-based learning adaptation. The performance of the proposed routing region scheme and other schemes is evaluated by simulation in Section 5.4 and Section 5.5. Further, Section 5.5 presents analysis of the simulation results. Finally, section 5.6 draws the conclusions.

5.2 The routing region for route discovery

In this chapter, I propose a location-aided routing technique that determines a zone or region for controlling the ares of routing based on the location information of nodes. The following provides basic information about the importance of routing regions and their applicability in creating energy-efficient and optimised routing.

A routing region in some location-aided routing algorithms is developed to limit the route discovery flooding as well as data packet transmission between the sending nodes and the BS. Since, various applications in WSNs have different rules according to node deployment, the shape or the calculation of the routing region should be appropriate to the task. For instance, the shape of a routing region for a military application or bush fire monitoring will be different from home automation applications. In geographic routing approaches, the routing region is not used to route packets to subsequent nodes, which have closer distances to the destination (i.e. BS) to reduce both communication costs and transmission delays. But, the location information of nodes is used to setup 'space searching' for both route discovery and packet relay thus restricting data flooding in the entire network that potentially will waste resources and cause data redundancy.

By using the information of nodes' locations, the data route can therefore be determined. After route setup, when the BS and all nodes in the network know the positions (whether hops or coordinates) of each other, the area of routing is then established. Based on the positional information, both BS and a source node initiate the creation of the routing zone. At first, BS creates a line to connect the BS and the source (S) then, BS defines an initial radius for each pair of adjacent sides of the region which are perpendicular to the line of S-BS. These two sides are the boundaries for the routing region itself, as shown in figure 5.1. The establishment of a routing region could be varied. Individual nodes could decide the routing region independently, or it could be done by the base station. A region needs to be able to self adjust in order to achieve efficient routing It is not applicable to define a static radius for every routing region if I want to achieve optimum QoS performance. We should design the proposed scheme that adapts to the changes in the network by proposing a self-adaptive routing region based on the value of QoS parameters instead. At a particular situation, a small routing region can probably save the energy consumption of the network for data routing as a smaller zone only involves fewer nodes being used for relaying packets. Thus, other nodes outside the routing region will go to idle mode. However, when the network becomes congested, and many nodes in the zone fail due to energy depletion, the data delivery cannot be maintained thus decreasing the data delivery capacity at the BS. Therefore, defining a self-adaptive radius for routing region is an important issue that will be discussed further in the following sections. For this reason, by using the routing region techniques, this chapter proposes to use self-adaptive routing region techniques to control the searching area for both route discovery and packet relay.

5.3 Routing Algorithm for the proposed routing region

The proposed routing regions approach, RuLeARR is a localised-routing that uses the information of the neighbour nodes' locations to initiate the route discovery. We assume that each sensor knows its geographic (location) information using established localisation methods; further this geographic information can be used to setup the routing region for data delivery as described in Table 5.1.

Sequential	**Program Command**
1	S sends a ROUTE message
2	If the node X received ROUTE before then discard ROUTE;
3	Else
4	If ROUTE. destination = X then respond by replying with RREP message using the reverse link;
5	Else
6	If ROUTE. destination $\neq$ X then forward ROUTE to X's neighbours until reaching ROUTE. destination;
7	Destination (BS) updates RREP message
8	RREP is sent by BS to a S through intermediate nodes
9	If an intermediate node X receives RREP message, then
10	Calculates its location whether inside or outside the predefined region
11	Else
12	If RREP message already received before then discards RREP mesg.
13	Else
14	Forward RREP to a source node S
15	End
16	End
17	End
18	End
19	Every node can locate its position to the region
20	Nodes inside the region are forwarding data packets otherwise are not relaying packets
21	End

Table 5.1: The setup process for the routing region between a source node and BS

When a source node wants to deliver data to a destination, the source sends a route-request (ROUTE) message to the destination via intermediate nodes (neighbours). If the ROUTE message has been already received by an intermediate node X then it discards ROUTE. However, if the ROUTE is the first message arrived at node X and the node X is not the destination node, this intermediate node will rebroadcast the ROUTE message using multi-hop communication until the ROUTE message reaches the destination (in this case is BS). Further, if the intermediate node X is a destination of ROUTE message, X then responds back by sending a route-reply (RREP) message to the source using a reverse path. Otherwise, the intermediate node will forward the ROUTE message to the destination node (BS) through its neighbours in a hop-by-hop manner. The RREP message sent by the destination includes the routing region information: locations of both the BS and the source node, the initial range of routing region and the calculation of source-BS pairing vector. Each intermediate node X receiving the RREP message will execute routing region discovery to determine the location of node X whether inside or outside the region as well as forward the RREP message to the source node as described in algorithm 1. Nodes in the region area are responsible to forward data while nodes out of the region are not selected to reroute data.

This proposed enhancement approach exploits the information of nodes locations to track the route discovering zone to a specific routing region instead of blindly relaying data packets in the entire network. The following sub-section describes the route discovery based routing region for data routing.

5.3.1 Location-based Route Discovery

To achieve efficient routing region adaptation, the BS in the proposed scheme adjusts the region using the information about the region that includes delivery ratio, average transmission delay and the actual energy of the current region. Hence, the adaptation considers the entire situation of the region.

According to Table 5.1, at the beginning, BS defines the routing region and broadcasts a RREP message containing all information that corresponds to the routing region. The intermediate nodes receiving RREP message will use the information in RREP to determine their locations, based on the defined region for routing. A proper choice for defining a radius to establish the routing region directly affects the performance of the network in terms of energy consumption and message-overheads. It is the motivation to avoid use of static routing region; as the non-dynamic region, at a certain circumstances, cannot include enough nodes in the region and results in failure to do data routing. Further, a 'packet detour' occurs when the region consists of high number of nodes. In the following paragraphs description of how to determine a radius for routing region is introduced.

We suppose the routing region is an area shown as dotted line in figure 5.1, where it is assumed that the coordinate of node S, BS, X and Y are (X_S , Y_S), (X_{BS} , Y_{BS}), (X_X,Y_X) and (X_Y,Y_Y) respectively.

The distance between node X and the line (S-BS) is $\bar{x}$ and the distance from node Y to line (S-BS) is $\bar{y}$. X is considered within the routing region if its distance is $0 \leq \bar{x} \leq$ R, otherwise is out of the region like Y. As the line (S-BS) and the RR-1 and RR-2 are perpendicular, I then calculate the angle between the center line (S-BS) and the line that connects S to any node as drawn in figure 5.2. We suppose the line (S-BS) is a vector $\vec{a}$ and the line (S - node X) is a vector $\vec{b}$. The result of the cross-product of vectors $\vec{a}$ and $\vec{b}$ is the vector $\vec{a}b$. Using the cosine law, I derive the following equation:

$$(\vec{a}.\vec{b}) = \|\vec{a}\|.\|\vec{b}\|Cos\phi(5.1)$$

Then I can find the angle θ,

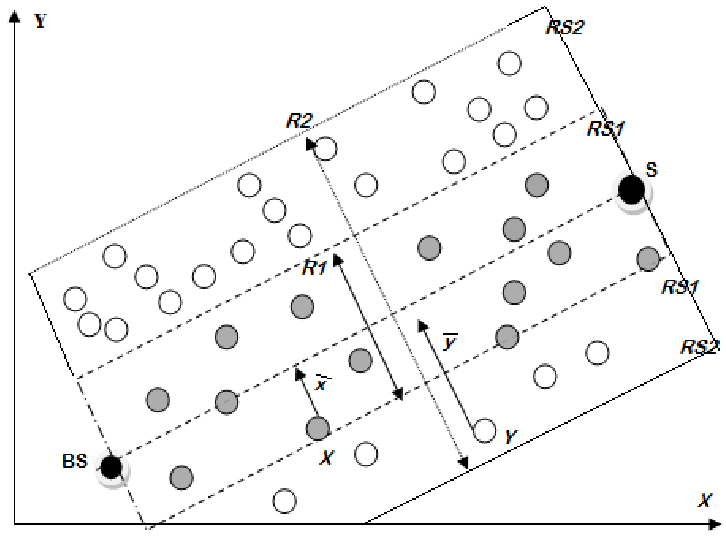

Figure 5.1: The routing region mechanism

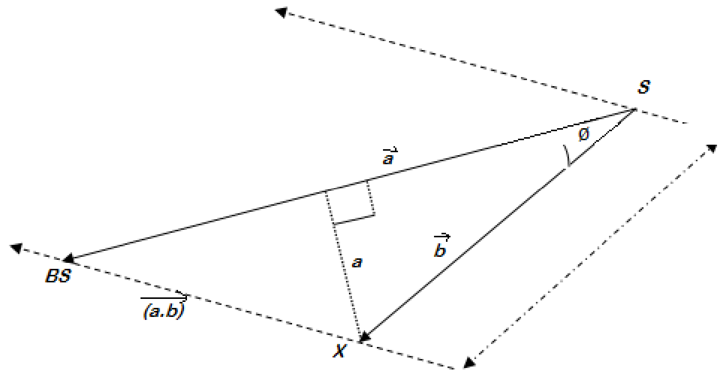

Figure 5.2: Distance between a node to the line (S-BS)

$$\phi = \cos^{-1} \frac{\vec{a}.\vec{b}}{\|\vec{a}\|.\|\vec{b}\|} \quad (5.2)$$

$$\phi = \cos^{-1} \frac{\sqrt{((Xbs-Xs)*(Xx-Xs))+((Ybs-Ys)*(Yx-Ys))}}{\sqrt{((Xbs-Xs)^2+(Ybs-Ys)^2)*((Xx-Xs)^2+(Yx-Ys)^2)}} \quad (5.3)$$

The following equations calculate the distance between a node and the line (S-BS) to determine the location of a node whether it is inside or outside the region.

$$\alpha = \|\vec{b}\|. \sin\phi = \sqrt{[(Xx - Xs)^2 + (Yx - Ys)^2}. \sin\phi \quad (5.4)$$

$$\alpha = \sqrt{[(Xx - Xs)^2 + (Yx - Ys)^2}.sin[\cos^{-1} \frac{\sqrt{((Xbs-Xs)*(Xx-Xs))+((Ybs-Ys)*(Yx-Ys))}}{\sqrt{((Xbs-Xs)^2+(Ybs-Ys)^2)*((Xx-Xs)^2+(Yx-Ys)^2)}}]$$
(5.5)

If $\bar{x}$ is less than R so that the location of a node is in the region and then node is likely to be selected as the next-data forwarding node for packet routing. In contrast, all nodes outside the region will not be chosen for routing data packets.

To initiate data routing, the nodes periodically exchange information with neighbours about the nodes parameters: energy level, distance to the BS, error rates according to both channel access failure and queuing buffer; and store the information of the neighbours that can be used as a knowledge for routing decision. All the nodes are neighbours within a node's transmission range either inside or outside the region. Each node maintain information about the neighbours current state in its routing table and applies a fitness approach in (Rizal et al., 2012) to select a next-node in the region which has the quality (node states) that suits the traffic requirements. Hence, the nodes inside the region are used for data routing only. In contrast, other nodes outside the region will save their energy by deactivating the transmitter activities e.g. transmitting or receiving packets. Thus, the energy consumed in the network can be reduced significantly, which can improve the lifetime of the network.

5.3.2 Self-adaptation of the routing region

We adopt the approaches (Xiao Hui et al., 2011; Ko and Vaidya, 2000; Basagni et al., 1998) in my routing protocols and unlike WSNHA-LBAR (Xiao Hui et al., 2011), the adaptation in the proposed algorithms is not only because of the decision for successful packet transmission, which is represented by the reception of the reply message (RREP) by the source node. In WSNHA-LBAR if the source node does not receive the RREP message from the BS, the source node will change the region as many times as the number of the occurrence of the unsuccessful transmission message. Thus, it causes reduced efficiency for the routing region adjustments, and leads to energy wastage, even though this approach is likely to be more adaptive to network conditions (such as link instability or heavy collision in the MAC layer).

In the proposed scheme, the range of the routing region is determined based on the region attribute values defined by a set of network layer parameters which includes delay, delivery ratio and nodes' energy in the region. Hence, the adjustment of the routing region for packet routing, directly affects the performance of the network. For instance, if the routing region is too narrow, it may limit the number of nodes participating in data delivery. This path leads to no-route or limited routes, resulting in creating 'routing holes'. If a path has a hole, packets will be dropped instead of being delivered and the routing setup needs to be done multiple times to prevent 'routing hole problem'. Thus, the route setup consumes excessive energy and causes reduced network performance. In contrast, expanding the region area much wider will result in joining many nodes where some nodes may not be used for packet routing. In addition, the expansion of area of routing improperly may cause packet detouring (broadcast storm) that degrades the network performances, and the benefits of the algorithm is lost.

In order to reduce the routing region overheads, and energy consumption, the adaptation of the region in the proposed scheme is done periodically using the time when the network is updated with the use of regular hello packets. Hence, the adjustment of the routing region in the proposed scheme may not result in success all the time but, it can achieve energy saving as shown in figure 5.3.

According to the scenario in figure 5.3, if the time for transmitting a data is t_d, the time for routing region adjustment is t_{adj} and the time from the current Hello packet to next Hello packet is t_{hello}, N is the number of data and α is the number of routing region adjustments, then traffic load during the hello cycle period is:

$$t_{hello} = N(t_d) + \alpha(t_{adj}) \quad (5.6)$$

If no routing region adjustment is required during Hello cycle then the equation formula is:

$$t_{hello} = N(t_d) + 0(t_{adj}) = N(t_d) \quad (5.7)$$

We can determine energy consumed during one hello cycle period as follows.

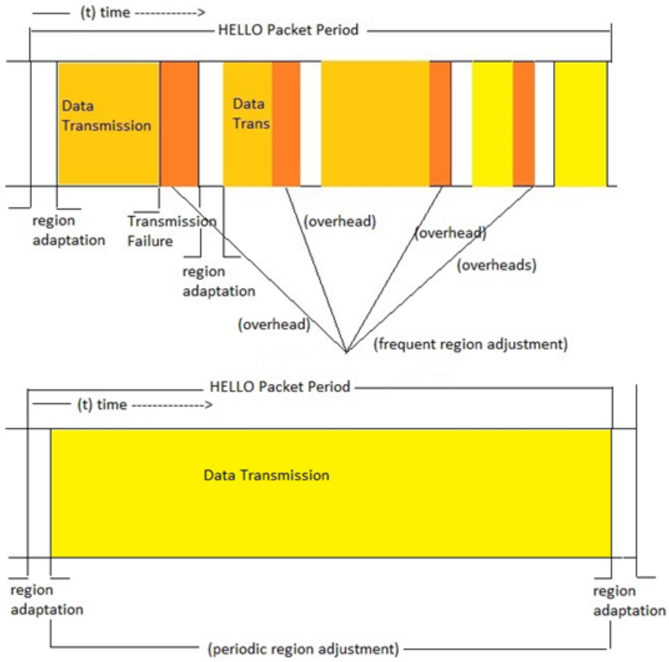

Figure 5.3: The difference of the region adjustment between the existing approach and the RuLeARR approach

$$E(one\_hello\_cycle) = n\left(\left(E_{BOT} + \left(tlT + \frac{LM}{R}\right) * Ptx\right) + (tlR + \frac{LM}{R}) * PRx\right) + \alpha(\left(E_{BOT} + \left(tlT + \frac{LREG}{R}\right) * Ptx\right) + (tlR + (\frac{LREG}{R}) * PRx) \quad (5.8)$$

Where E_{tx} is the energy for a node to transmit packet, E_{bot} is the energy to sense and ensure the availability of idle channel by executing back-off mechanism, L_M is data length, L_reg is region adjustment packet length, R is data rate (250 Kbps) and P_{tx} and P_{rx} are power for both transmission and reception. E_{rx} is the energy for a node to receive a message. T_{IT} and T_{IR} are the times for a node to change its state to transmit/receive.

If the algorithm does not change the routing region until the next hello cycle period, then the formula 5.8 changes to the following equation:

$$E(one\_hello\_cycle) = n\left(\left(E_{BOT} + \left(tlT + \frac{LM}{R}\right) * Ptx\right) + (tlR + \frac{LM}{R}) * PRx\right) \quad (5.9)$$

Referring to the equations (5.6 - 5.9), it is justified that the proposed scheme is able to reduce the number of overhead messages for adjusting the region as this approach uses the period of the hello packet to initiate region adjustment. Thus, most of the time, from the current hello packet to the next hello packet the network is used for data transmission instead of region adjustments. Hence the proposed scheme can improve the utilisation of the network while achieving energy efficiency.

To set up a region, the BS periodically updates the region attributes, which include the region's average delay, region's average delivery ratio and region's energy that are used for the basis for a routing region adaptation. As explained previously, the information of the region attributes is collected at the BS using the interval time of the hello packet that is broadcast regularly. Thus, the hello packet will include both the node attributes as described in Chapter 3 and the region attributes.

In addition, to establish an energy efficient situation, the BS measures the energy consumed in a region that is defined as the total remaining energy of all nodes in a region. Each node in a region regularly (using hello packet period) updates its current energy of the hello packet and sends the hello packet to both the BS and its neighbours. If the transmission of the 'energy message' collides and does not arrive at the BS, a notification message is sent back to the node in order to do retransmission. The BS starts to calculate the region attributes in order to adjust the routing region after all region attributes are obtained.

It is not easy to define the best and most efficient region for routing for every source-destination pair. To optimise routing in terms of energy efficiency and reliability, the proposed scheme should be able to incorporate minor changes in the network layer parameters to meet the QoS requirements. The proposed algorithm brings forward two adaptation approaches that implement a self-adaptive routing region adjustment. They define the region based on WSN QoS performances but they use different techniques in adjusting the routing zone according to context changes. They are a Rule-based Learning approach, and they use a threshold value for adjusting the region; and a situation-aware adaptation approach (SASARR), which will be discussed further in the next chapter. The following section explains the details of the first approach (Rule-based Learning Adaptation Approach for Routing Regions, RuLeARR).

5.3.3 Realisation of Self-Adaptation of the routing region

This section presents a self-learning BS that performs a dynamic adjustment of the routing zone based on WSN QoS performance using the network layer parameters. To ensure a self-learning (automatic) adjustment of the routing region radius, the following information is required. They are (1) the defined attributes that are sent and calculated at the destination (BS) such as delivery ratio, average transmission delay and the actual energy of the current region, which is defined as the total remaining energy of nodes in the region; and (2) the knowledge of the decision support system when the attributes are not collected at the BS.

Periodically, the BS adjusts the routing region using the information about the region, which is updated periodically in order to obtain the actual situation of the region. The BS can receive the number of successful transmissions of data packets as knowledge for making decision. Further, making a decision for the region radius adjustment using only the number of successful packet transmissions is not sufficient to define radius. The proposed algorithm should consider other aspects like energy and latency for achieving both efficiency and appropriateness for the automatic routing region adjustment. The self-learning of the BS in adjusting the radius for the region occurs as the BS counts the number of successful transmissions, transmission delay and the energy of the region using an approach, named as rule-based learning adaptation, which uses the following formula:

$$\text{RR(region)} = \text{Avg}\ ((\omega_1\ \text{QoS}(_{de}) + \omega_2\ \text{QoS}(_{En}) + \omega_3\ \text{QoS}(_{DelRat})) \quad (5.10)$$

where $\omega_1\text{QoS}(_{de})$ is the function of delay of all data traffics in a certain time period. The lower the delay, the better the QoS performance. ω_1 is the weighting constant to standardise the value of delay. $\omega_2\text{QoS}(_{En})$ is the function that represents the energy of all nodes in the routing region. More energy means higher chances for a node to stay alive to deliver data. ω_2 is the standardisation weighting factor for energy. $\omega_3\text{QoS}(_{DelRat})$ is the function of data Delivery ratio at BS. Higher ratio reflects the effectiveness of the routing algorithm to route packets. ω_3 is the delivery ratio standardisation weighting factor.

As the rule-based learning adaptation scheme sets the region for routing based on the formula 5.10, the results in the above equation provide three possible scenarios in adjusting routing region. The adjustable region includes low, medium and high as presented in the Table 5.2. After the completion of the updatable QoS attributes, the BS then defines a threshold value T for the average of cumulative region attributes with the purpose to vary the region range "R" as

If RegionAttrs($_i$)< T_{Low}, then a larger region range is set to "R1"
Else if RegionAttrs($_i$) > T_{low} AND ≤ T_{high}, a medium region range is set "R2"
Otherwise a smaller region range is set "R3"

The following table describes that the QoS values: energy, delay and delivery ratio are important criteria for setting up a threshold like "T" to vary the region of routing.

No	Definition Routing Region	Region Attributes	Zone
1	High (R1)	$0 <$ value $\leq T_{low}$	2R
2	Medium (R2)	$T_{low} <$ value $\leq T_{high}$	R
3	Low (R3)	value $> T_{high}$	$\frac{1}{2}$R

Table 5.2: The setup process for the routing region between a source node and BS

Table 5.2 shows that adaptation of the routing region can be classified into three different levels, e.g. (low, medium, high). The definition of 'low' refers to the radius of the routing region of "$\frac{1}{2}$R". The 'medium' has the value of ("R") and the 'high' refers to "2R". R is the initial power transmission range.

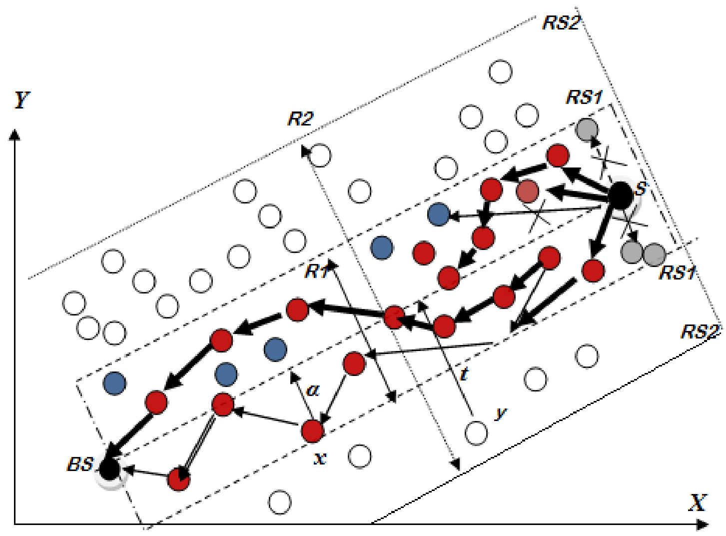

Figure 5.4: The routing direction when the network is congested using the proposed routing region algorithm

We conduct simulations with several T values for the same network settings and configuration to find the optimum route performance. The best value of T is used to verify the effectiveness of the rule-based learning adaptation in order to compare to the fuzzy scheme.

According to the scenarios in the formula 5.10, in the situation that leads to packets losses as a result of packet route failure, the result of the cumulative attributes will be lower than T_{low}. Thus, the proposed algorithm needs to expand the radius into 2R to involve more nodes participating in the region as depicted in Figure 5.4. This can increase the opportunity to find nodes whose qualities fit the requirements of traffic classes thus improving the continuity of data transmission. The expanding region aims to achieve high throughput. On the other hand when the situation changes to normal and the BS determines that the region radius is exceeding the threshold T_{high}, then area of routing will be reduced to $\frac{1}{2}$R. The smaller region can avoid packets being traversed via a detour therefore it can reduce packet looping or "broadcast storm" in the MAC layer as well as packet delay. This smaller routing region can reduce the energy consumption of the entire network since nodes outside of the region can go into energy saving mode.

Nodes may become congested or "die faster" if there is heavy traffic, this leads to packet loss because of packet route failure. Congestion causes difficulty to route packets from source to destination. According to the scenario in figures 5.4, even the routing algorithm in the proposed routing region scheme lets the source node to find the closest neighbours, the packets are still not able to be forwarded to destination. This is because the routing region scheme will not relay packets to the nodes outside the region even no route for delivering packets left inside the region. Thus, next packet delivery is dropped as depicted in figure 5.4.

In contrast, in the original fitness scheme (TeGaR), when nodes having less hop count are not activated, the source node will find the closest neighbours to route data packets to destination as shown in figure 5.5. The packet delivery will then be continued to the next nodes whose locations are outside the region until the packets arrive at the BS. This may increase throughput but it causes longer transmission delays or the packet may be lost due to packet detours (broadcast storm).

In this algorithm, the nodes create the tables to calculate the attributes of the region, and store the value of the different R. The tables are dynamic depending on two parameters i.e. region attributes, which represent the performance of the routing algorithm according to the changes of the radius R, and the initial radius Rinit of the region. The proposed routing region realises that the failure of data transmission does not directly affect the change of the region radius until the next hello packet occurs in order to reduce the frequent changes, which results in network inefficiency and higher energy consumption. The following presents the sequence of exchanging Hello packets and the need for adjusting the radius of the region when the value has changed. The above order of Hello packets

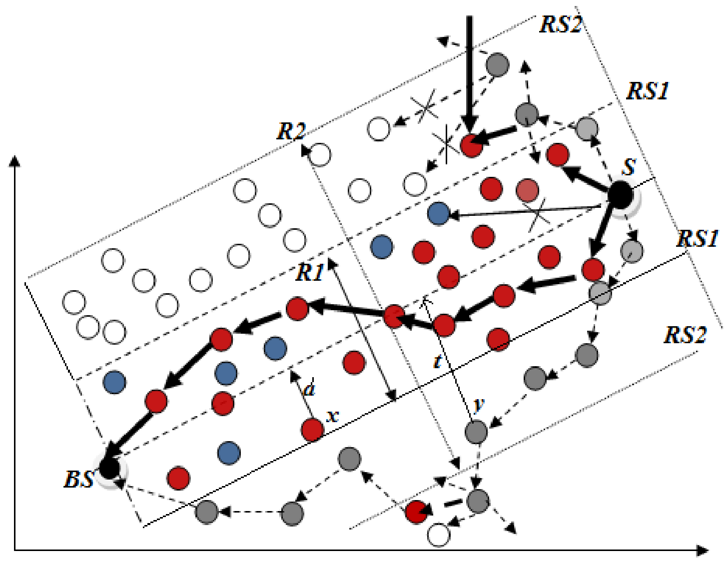

Figure 5.5: The routing direction when the situation is congested for the fitness algorithm

describes the establishment of a routing region using the initial radius. When the value of the calculation of the region attributes such as delay, energy, delivery ratio is increased, it indicates the performance of the network is high thus, the radius of the region decreases to save energy by reducing the number of nodes participating in the current routing region. Therefore, the self-adaptation is able to increase the QoS performances. As the variations to the region radius R are in-frequent thus, the state information storage will not consume significant sensor node's resources in terms of power or memory. The following section describes the simulation results and discusses the analysis of all the schemes.

5.4 Model of Simulation

In this section, I develop a simulation model in order to study routing protocols/schemes. Further, this model helps designers and engineers to study and comprehend the operation of the schemes under different conditions of the WSNs. The simulation model consists of simulation performance metrics and the environment and parameter of simulation that can be used to make the simulation results more meaningful for assessment and comparative evaluation between the schemes. We use the results to analyse and evaluate the benefit of the proposed routing region as the enhancement approach to the original scheme (TeGaR) to improve both energy efficiency and route optimisation. In order for the simulator to be applied to such assessment, I first describe a set of metrics to meet the same standard of evaluation.

5.4.1 Performance Metric

To measure the performance of the schemes, I determine a set of objective performance metrics to evaluate the routing schemes' performances. The following metrics are described to quantify the simulation results.

- *Network lifetime.* As the aim of the proposed routing region scheme is to achieve efficiency for packet routing, so this metric represents the effectiveness of the algorithm to extend the network life time (in simulation) until the first node fails due to energy depletion.
- *Network Throughput and delivery ratio.* Defined as the average rate of successful packet transmission for a pair of source-destination, for both reliability and effectiveness evaluation.
- *Average delay per packet.* This metric represents the response time of routing schemes in packet transmission under different network situations. It is defined as the average time taken to transmit a packet until reaching the destination (BS).

Based on the above-mentioned standard metrics, I design a simulation environment in order to ensure that the schemes are operational and easy to analyse in the WSNs simulation in the following subsection.

5.4.2 Simulation Environment

In this section, I outline environment and parameters for conducting simulation for implementing the schemes. As I compare the proposed RuLeARR to TeGaR and SPEED, I introduce a new module with new features that extends the capability of my tailor-made simulator I have developed to facilitate the implementation of routing region approach. The update of the features also contains some bug fixes and code extensions to the graphical runtime environment for details. The features upgrade the functionality of the simulator to define a specific region/zone for packet routing based on the geographic information of nodes in the network. In addition, the updated features support the algorithm to report the location information of nodes to the BS in order for the BS to decide whether each node should be included in the defined region or not.

Similar to the simulation parameter described in the Chapter 3 and 4, I exploit a CSMA/CA mechanism (Kohvakka et al., 2006) which uses the 802.11 protocols medium accesss phase in the network. In the experiments, the network consists of 350 randomly placed nodes in a 300300 meter square area. The BS position is determined in the middle of the network. Congestion is produced by creating data flows from 6 sources with different traffic rate (1 - 50 pps: packet per-second). The sources are placed at corners and edges of the grid. To test the robustness and the energy efficiency of the schemes, I run simulation for 300 sec with various parameters as: stationary nodes, data rate (250Kbps), packet size (1 Kb), Hello cycle period (8 s), initial energy (15J) and initial transmission range (30 m).

5.5 Analysis of the Simulation Result

In WSNs, since running real experiments is costly and time consuming, I run simulation to implement the model and to predict the performances of the routing schemes in the real world. In this section, I have run experimental works in order to examine the effectiveness of the RuLeARR compared to the original (TeGaR) and SPEED (He et al., 2003) in terms of throughput, network utilisation, energy consumption and latency.

Before I start to analyse the results, the proposed RuLeARR that employs routing region approach may have high packet drop ratio when the network is heavily congested as many nodes inside the region will fail due to energy depletion. Thus, the remaining data will not be forwarded to the destination. In contrast, other schemes (TeGaR and SPEED) that do not implement routing region for controlling routing area are still able to find nodes to deliver packets to the BS. However, the region of packet routing can manage the number of nodes involving in data delivery thereby nodes outside the region will change to inactive mode thus reducing the overall energy consumption of the network. The following presents some performance results obtained by the simulation under different parameters.

5.5.1 Throughput

Overall, the second adaptation scheme (RuLeARR) has higher throughput than the two corresponding schemes (TeGaR and SPEED). The data capacity at the BS as identified as the network throughput is increasing when the load injected into the network is also increasing, and then the ratio of data delivery flattens for heavy traffic rate in the network.

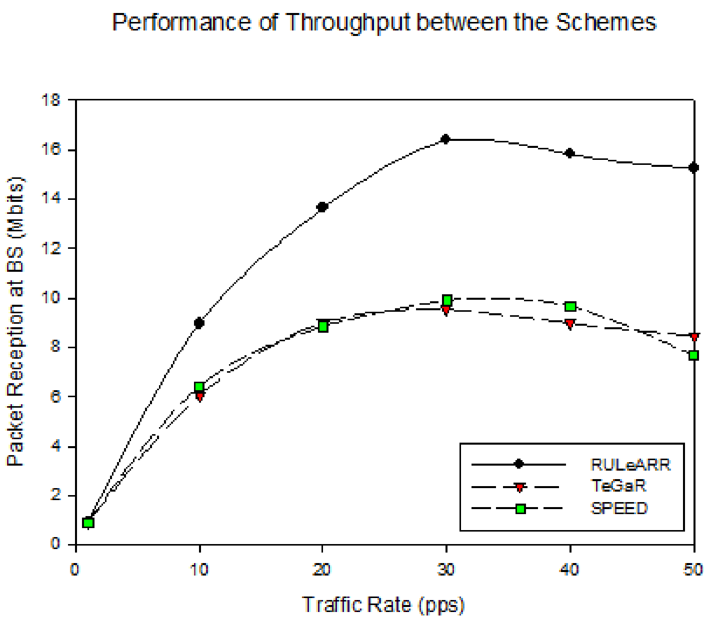

Figure 5.6: Packet reception of all types of traffic

As Figure 5.6, RuLeARR reaches a peak packet reception (16.5 Mb) at 30 pps before the rate drops to (15.5 Mb) for the rest of traffic conditions. In contrast, the original fitness approach and SPEED under similar conditions only achieve almost half the throughput (about 9-10 Mb). The difference in throughput performance is due to the fact, as RuLeARR only forwards the packets to the intermediate nodes that are in the certain region so that the routing of packets are properly guided from the source to ensure successful arrival at destination (BS) by controlling the area of routing. On the other hand, when no suitable path is available, the two schemes (TeGaR and SPEED) use traditional flooding method in order to continue route searching for packet transmission. Thus, more packets are lost during the transmission period.

The figure 5.6 demonstrated the effectiveness of the proposed routing region, which has used directional information for packet routing, and avoiding packet flooding in the whole network. However, as Figure 5.7 shows when the throughput increases and more data are flowing in the network, the number of lost packets also increases.

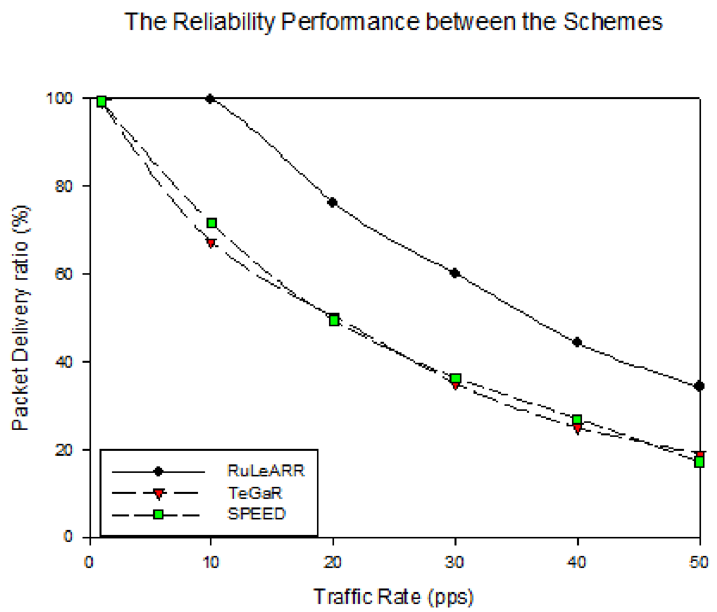

Figure 5.7: Data Delivery Ratio of all types of traffic

Note that packet reception increases under heavy traffic conditions (see Figure 5.6), but the ratio of packet delivery significantly decreases as more packets are injected in the network. Congestion causes many nodes to fail to relay packets to BS so many packets are dropped during data transmission. Figure 5.7 demonstrates that the proposed enhancement approach (RuLeARR), which controls data routing in a closed area succeed to avoid packet detouring and reduces communication cost. Hence, more packets can be delivered to the BS even the network becomes congested. Under heavy congestion, TeGaR can maintain delivery ratio at 35% while other schemes can provide no more than 20%. For low traffic conditions, RuLeARR achieves a near 100% packet delivery ratio, but the other two schemes only have a 67% packet delivery ratio. Figure 5.7 validates that the routing region algorithm (RuLeARR) is sensitive to the network dynamic.

The above figure also shows that under heavy congestion, all schemes fail to find the forwarding nodes and the number of lost packets significantly increased. However, as RuLeARR supports scalability and coverage control, it can reduce undirected packet transmission and routing overheads thus, energy significantly decreased.

5.5.2 Network Utilisation and Efficiency

The experiment evaluates the effectiveness of the schemes in improving the utilisation of the network, which also affects the energy efficiency in terms of the network lifetime and the total number of node failures due to energy depletion.

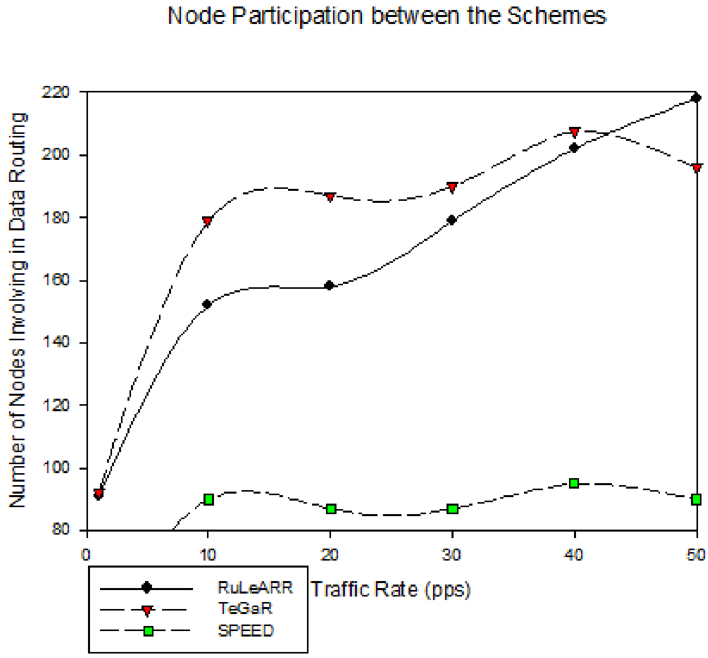

Figure 5.8: Load balancing during data delivery

Figure 5.8 explains the importance of the routing algorithms which perform route differentiation. Compared to SPEED, both RuLeARR and the original fitness approach, TeGaR attempt to balance the network load by involving more nodes to participate in data delivery. By increasing the number of nodes joining in routing, the algorithm maximises the network utilisation to achieve efficiency. This is not happening in case of SPEED because SPEED does not provide traffic differentiation and will not change the routing path until the required packet speed cannot be maintained.

Figure 5.9 shows the average lifetime. The longer the time it takes for an active node to fail, less route breakage occurs that degrades the network performance. The advantage of using a routing region conserves the nodes' battery for relaying packets compared to non-routing region algorithms. The proposed routing region scheme assures the nodes to avoid packet flooding by controlling the area of transmission, leading to energy saving and longer network lifetime. For instance, when the network is less congested, RuLeARR doubles the route lifetime compared to other schemes and the gap decreases when the network becomes more congested. Under heavy congestion, all schemes significantly achieve low-lifetime because many nodes fail quickly.

The efficiency of the proposed scheme is validated by fewer node failures during the simulation as shown in Figure 5.10. When the energy of the nodes in the region falls and the calculation value of the region attributes decreases, RuLeARR performs a self-adjusting region approach that dynamically increases the radius of the routing area or otherwise. The expanded routing region might increase node participation to deliver packets whereas the expansion balances the network load and helps many nodes to save energy. In contrast, the original fitness algorithm, without routing region approach, may experience packet detouring (broadcast storm) that increases the communication costs and overheads. Therefore, RuLeARR can result in fewer drained nodes while providing

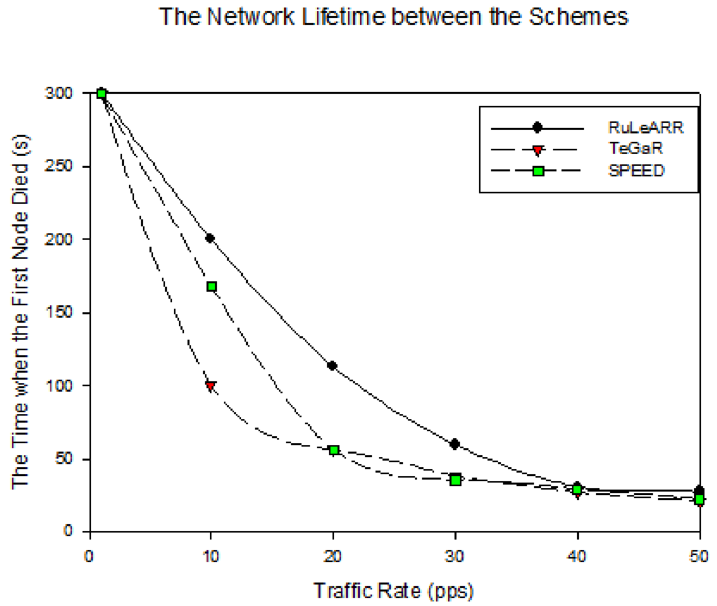

Figure 5.9: The time when the first node died (sec)

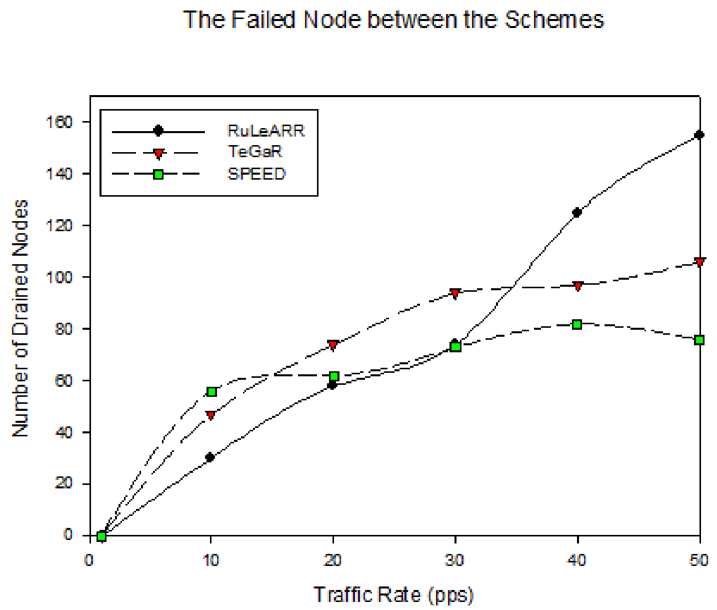

Figure 5.10: Energy consumption for data communication

higher throughput and a higher data delivery ratio except, when the traffic becomes more congested.

5.5.3 Average packet delay

The average delay tends to increase when the traffic rate is increasing because more packets being forwarded will incur longer delay for packet queuing at intermediate nodes. Yet, the superiority of a routing region mechanism is justified in Figure 5.11 where RuLeARR shortens packet delay for all data categories. The proposed scheme has a 20% (approximately) lower delay compared to the original fitness scheme and nearly 50% compared to SPEED. But the transmission delay becomes similar between RuLeARR and TeGaR when the network is heavily congested. Figure 5.11 describes that by controlling the region of packet routing, the proposed scheme avoids packet detouring and shortens en-route data from a source to the BS.

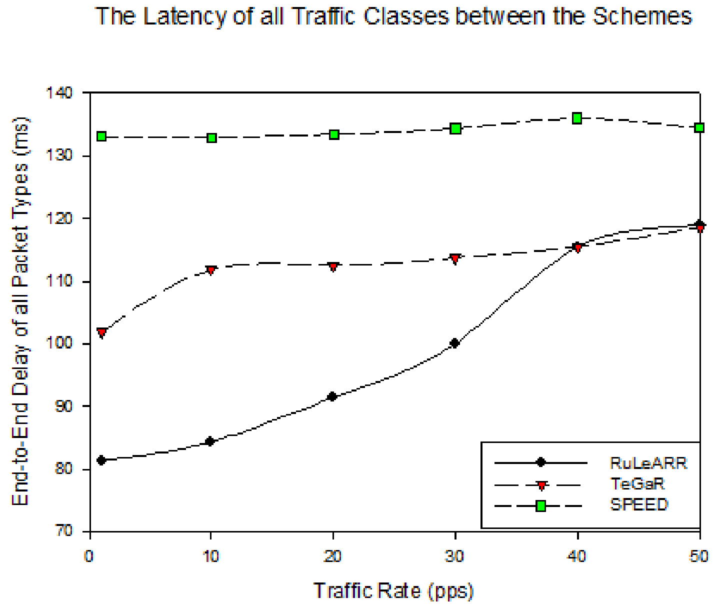

Figure 5.11: Average delay for all types of packet traffics

5.6 Summary

This chapter investigated analysis of the packet detouring caused by unguided data transmission, and introduced the second enhancement approach to address the problem of unguided packet relay due to improper area of routing that would affect the overall performance of the network. The proposed scheme termed RuLeARR performs a routing region mechanism for controlling the area of routing for achieving route optimisation. This mechanism is then combined with the fitness method for supporting heterogeneous traffic for different QoS requirements.

The proposed scheme uses a self-adaptive algorithm, named rule-based learning adaptation, that dynamically adjusts the size of the routing region, based on the BS's calculation of the network layer parameters to increase the successful data routing while considering energy efficiency. The results represent that the self-adaptation of routing region has greater tolerances for changes of the network conditions and reduces the need for human intervention to maintain the QoS performances in these dynamic environments. The simulation shows that the proposed scheme outperforms the original scheme (TeGaR) and SPEED in term of energy consumption, transmission delay, throughput and reliability (packet delivery ratio) under different congestion levels.

The next chapter will highlight the challenge to discuss the new design for the adaptation of the routing region in order to maximise the benefits of adaptation of routing region approach in terms of the improvement of the QoS performances in a cost-efficient manner. The next research work is to determine an optimal routing region in resource constrained wireless sensor networks by making routing-algorithms context-aware of the network dynamics. The variation in the region boundaries should take into account multiple parameters for accurate sensor allocations, real situations, minor changes in energy efficiency, routing response time and reliability.

Chapter 6

Situation Aware Algorithm based Routing Regions

The previous chapter described the second enhancement to the TeGaR approach, named RuLeARR that aims to address the problem of 'packet detouring'due to link instability and non-uniform node deployment in WSNs. This chapter introduces a Fuzzy Logic based adaptive algorithm for routing regions approach that extends the RuLeARR approach further in order to provide a higher level of accuracy and granularity. This is achieved by incorporating situation awareness into the adaptation process of routing region, inspired by a fuzzy logic based situation reasoning method.

6.1 Introduction

The preceding chapters presented the TeGaR scheme, a novel QoS differentiation routing approach based on Fitness Function, and its enhancement approaches (i.e. TReNs and RuLeARR) that extend TeGaR's capabilities to cope with the challenges of the 'energy hole problem' and 'unguided/detoured packet transmission' for achieving both route optimisation and energy efficiency in WSNs. According to the routing region approach in Chapter 5, there is a need for an efficient and intelligent approach that provides a smooth (non-sharp/sudden) adaptation to improve QoS performances by adjusting the routing region according to the current situation of network dynamics. The extended adaptation approach should meet a trade-off between the efficiency and the accuracy for adjusting the routing region in (Narayanan et al., 2000) thus an increased throughput and a decreased delay with lower energy consumption can be achieved.

In this chapter, I propose a Situation Aware and Self-Adaptive Routing Regions (SASARR) scheme, that incorporates situation awareness into the adaptation process of routing region using a the Fuzzy Situation Inference (FSI) (Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009) for Routing Regions (FSIRR). The FSI model uses fuzzy logic and therefore it is able to represent approximate and imprecise context (Mendel, 1995; Zadeh, 1988). In here, situations that I consider are based on the context parameters related to network dynamics. These parameters include delay, network resources and reliability while a situation is represented as real-condition of the networks such as 'normal', 'congested'or 'hybrid'.

The proposed extended scheme in this chapter aims (1) to adapt routing regions by using the information about current situations of network; (2) to enable variations in the region boundaries that take into account multiple parameters for accurate sensor allocations, real network situations and minor changes in QoS parameters (i.e. energy efficiency, routing response time, reliability, etc) and (3) to drive an efficient and fine-grained adaptation (Delir Haghighi et al., 2008) according to minor changes in WSNs situations by using the FSI approach. SASARR with its FSIRR module aims to improve route optimisation while extending the lifetime of the network, which is one of the main concerns in WSNs.

The rest of this thesis is structured as follows. Sections 6.2 and 6.3 discuss the motivations and requirements for employing an improved and efficient routing region approach using situation awareness. Section 6.4 presents an overview of a fuzzy logic theorem as the fundamental theory underlying the proposed context-aware routing region scheme. The basic concept of SASARR scheme is described in section 6.5. This section discusses the FSIRR model, the mapping of context attributes into fuzzy logic principles, the FSIRR components and its reasoning techniques to describe the process of integrating situation awareness into the process of routing region adaptation. Section 6.6 and 6.7 describes the method of simulation and the analysis of the simulation results. Finally, section 6.8 draws conclusion.

6.2 Motivation for improved and efficient routing region approach

As described in Chapter 5, the range of routing region is changed and adapted regularly based on the calculation of the QoS attributes at the BS, which include delay, delivery ratio and nodes' energy in the region. The BS periodically calculates the QoS attribute values based on the rule-based learning adaptation scheme as described in Chapter 5. When the result (value) exceeds the upper-bound threshold, the region decreases. Alternatively, the algorithm increases the region when the value drops below the lower-bound threshold. This adaptation approach provides a simple technique to enable an optimal expansion of the routing area and maintain a desirable QoS to improve energy efficiency.

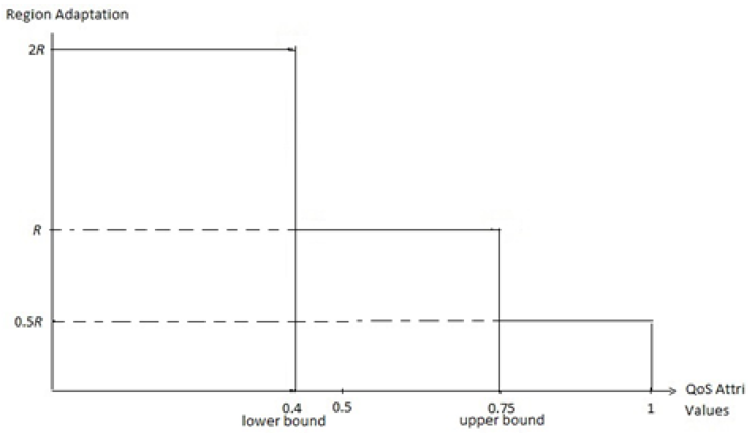

Figure 6.1: An overview of the region adaptation in RuLeARR

As shown in figure 6.1, RuLeARR classifies the adaptation process of the routing region into three different levels, e.g. low, medium, high. Each level requires a threshold value "T"that can be used for varying the region of routing. RuLeARR uses the best value of "T", which is determined through multiple simulation runs, and better network performances. RuLeARR defines each level of region with a fixed range of radius. The low region refers to 0.5R, the medium region to R, and the high region to 2R, where "R" is the initial power transmission range. Based on the routing region levels described above, it can be observed that the adaptation using the RuLeARR approach can only support the changes in three level of regions i.e. (0.5R, R and 2R or 'low', 'medium', and 'high') according to the QoS attributes calculation as depicted in Figure 6.1.

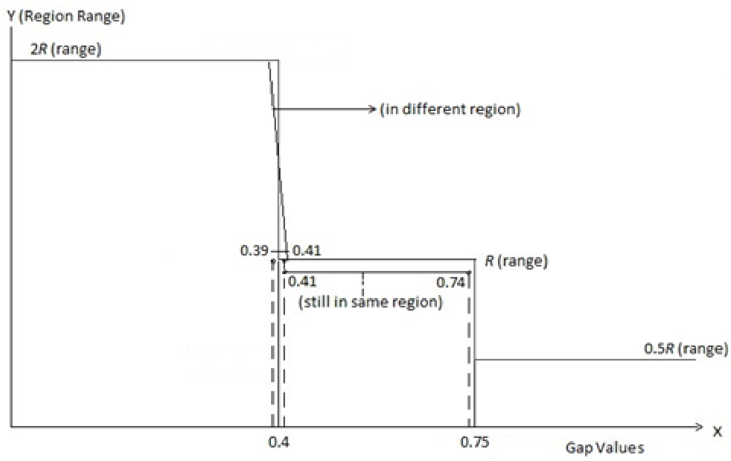

Figure 6.2: The upper and lower-bound for routing region adaptation

However, when a minor change in the attribute value occurs (e.g. a decrease from 0.41 to 0.39), RuLeARR suddenly decreases the region from (R) to a smaller region (0.5R) as the new value (0.41) is lower than the threshold for the lower-bound of (R) region. This sharp adaptation from R to 0.5R due to just some minor changes reduces the accuracy level and does not meet the real world and varying network related requirements.

Figure 6.2 illustrates the above-mentioned examples about the limitations of the RuLeARR approach in supporting different types of changes in QoS attribute values, and to perform a fine-grained and smooth adaptation. The RuLeARR reacts to perform adaptation only if the calculation of the QoS attributes exceeds or sharply drops a defined limit (threshold), and this adaptation model may degrade the performances of the overall network.

To improve efficiency of routing region, I propose a list of requirements that are discussed in the next section.

6.3 Requirements for optimal routing region approach

From the discussion and examples discussed in the previous section the following requirements for an improved and energy efficient routing region approach can be identified:

1. The re-positioning of the region boundaries should be established optimally in order to improve network performances;

2. The variation in the region boundaries should take into account multiple parameters according to QoS values for accurate sensor allocations;

3. Determining optimal routing region can be made efficient by making routing-algorithms situation-aware of the network dynamics that are inferred based on the region attribute values with respect the QoS attribute values;

4. To enable a gradual, smooth and fine-grained adaptation according to minor changes, the routing region approach should be aware of any changes in the current context (i.e. QoS attribute values of delay, energy and delivery ratio), and consider them in the adaptation process to provide better QoS performances;

5. The situation-aware adaptation for routing region ensures efficient and smooth adaptation, and avoid a very strict/sharp adaptation of routing region level that can result in resource wastage.

Based on the requirements mentioned above, I identify the need to improve adaptation of routing regions with a higher level of accuracy and efficiency through incorporating situation-awareness and leveraging the information about the occurring situations of the changes and energy level in the network. The consideration of using situation and context awareness makes the re-positioning of the region boundaries for individual communication sessions more accurate and efficient by taking into account contextual information. In WSNs, context may represent nodes, the state of nodes regarding location, energy level, connectivity degree, load status, sensor individual preferences, or any information that can have impact on the network layer such as routing, data aggregation, etc (van Bunningen et al., 2006; Dey, 2001).

In WSNs, the context is naturally dynamic and always changing, and it represents the state of the network with respect to average delay, average delivery ratio and energy level. By using these pieces of information, I can infer situations about the current network such as 'congested', 'normal', 'heavy congested', etc. We adopt a fuzzy logic based approach for developing situation-aware adaptation approach for routing regions The following section provides an overview of fuzzy logic theorem for situation and context-aware routing regions.

6.4 Fuzzy Logic Theorem for Context-awareness

Fuzzy logic is a form of many-valued logic that deals with uncertainty rather than fixed and exact contextual information (Novák et al., 1999). Fuzzy logic uses fuzzy sets where its variables have a value with a degree between 0 and 1 to handle the concept of partial truth, where the truth value may range between completely true and completely false. According to Zadeh in (Zadeh, 1975), Fuzzy logic has been applied to many fields, from control theory to artificial intelligence.

In the case of WSNs routing domain, to accomplish a gradual adaptation of routing region method, a situation modelling technique that is able to represent vague conditions of the QoS attributes and their degree of uncertainty is needed. A fuzzy logic has the potential to model imprecise context of uncertain environments (Cheverst et al., 2005) because fuzzy logic is an extension of multi-valued logic for approximate reasoning (Zadeh, 1988). Unlike Two-valued logic (Zadeh, 1988), Fuzzy logic is able to represent the meaning of predicate the expression of 'yes'or 'no' into vague predicate-modifiers such as 'very', 'slightly', 'extremely'and so on.

A fuzzy logic based system generally consists of the following components: Fuzzy Sets, Linguistic Variables, Fuzzification, and Fuzzy Rule Inference (FRI) (Zimmermann, 2001; Lowen, 1976) [24-26]. The following subsections describe these components.

6.4.1 Fuzzy Sets

Fuzzy sets are sets whose elements have degrees of membership. The membership degree is computed by a membership function that establishes the difference between classical sets and fuzzy sets (Zadeh, 1965). In classical set theory, the membership of elements in a set is assessed in binary terms or called crisp sets. The 'crisp' sets means an element belongs to a subset or it does not like two-valued logic which takes values 0 or 1 as well as 'true' or 'false'. In contrast, fuzzy sets assign an element with a continuum of grades of membership represented in the real unit interval [0, 1] (Gottwald, 2010). In fuzzy sets, the crisp sets are generalised and allow descriptions for the value to provide gradual assessment of the membership function. Thus, the fuzzy set theory can be used in a wide range of application domains in which information is vague, incomplete or imprecise (Mendel, 1995; Zadeh, 1965).

A fuzzy set is a pair (A, f) where A is a set and (f: A) → [0,1]. For each x ∈ A, the value $f(x)$ is called the grade of membership of x in (A, f), where this grade/degree is a value between 0 and 1. For a finite set A = (x_1, x_2,..., x_n), the fuzzy set (A, f) is often denoted by ($\frac{f(x_1)}{x_1}$ . . . , $\frac{f(x_n)}{x_n}$) .

Let x ∈ A, then x is not member (included) in the fuzzy set (A, f) if $f(x) = 0$. In contrast, the membership x is fully included if f(x) = 1, and x is called a fuzzy member if $0 < f(x) < 1$. The set x ∈ A| $f(x) > 0$ is called the support of a fuzzy set (A, f) and the function f is called the membership function of the fuzzy set (A, f) with values in interval [0, 1].

6.4.2 Linguistic Variable

Linguistic Variables define as variables whose values are not numbers but words or sentences in a natural or artificial language (Zadeh, 1965). The use of words or sentences is to describe a 'value' in the defined fuzzy sets when information is less specific than numerical information so that the use of linguistic characterisation suits the objective of the membership degree in fuzzy sets. The non-numeric values of linguistic variables are named as linguistic terms where each term is in fact a fuzzy set.

In sensor networks, different types of contextual information can be expressed by linguistic variables and fuzzy sets to represent the contextual parameters values. For example, "delay" is a linguistic variable if it's values are linguistic or non-numeric values rather than numerical. The values that it takes can include 'very low', 'low', 'middle', 'high' and 'very high'. In addition, each value in a fuzzy set using linguistic variables is paired with a continuum interval value [0,1] that expresses its degree of membership to the defined fuzzy set. For example, the 'low term' for a linguistic variable (delay) can be characterised by the following fuzzy set:

LowDelay (ms) = (25/1, 26/0.95, 27/0.9, 28/0.85, 29/0.8, ... , 50/0)

According to the above fuzzy set (delay), the membership degree of 29 ms in the 'low(delay)' fuzzy set is 0.8. The symbol of '/' is to separate the element value and its membership degree (Jang et al., 1997; Mendel, 1995).

6.4.3 Fuzzification

Fuzzification (Zimmermann, 2001) is a process that converts crisp input (i.e. sensory data) into fuzzy sets by using a membership function. The first step of fuzzification is defining linguistic variables and their terms. All linguistic variables form a term set as explained in the previous section. For each input and output variable selected, I apply membership functions to fuzzificate all the real values of the variables, and any of the values will belong to the membership functions with a certain degree of membership.

The next process of fuzzification is selecting membership functions. This function allows us to represent a fuzzy set in X by assigning a degree of membership to each value in X in real numbers with interval [0,1] to the defined fuzzy sets. For example, when I apply a delay of 20ms, the membership function can assign the degree of 0.25 to the 'normal' term of a fuzzy set or a different membership degree (i.e. 0.75) to the 'low' term fuzzy set. There are a number of different membership functions such as trapezoidal, triangular, Gaussian, generalised bell, and others (Jang et al., 1997).

6.4.4 Fuzzy Rule Inference

Fuzzy Rule Inference (FRI) is defined as a conditional statement expressed by If-Then rules to formalise the reasoning process of human language by mapping an input space to an output space using fuzzy logic (Jang et al., 1997). There are two common ways to represent fuzzy rules in direct method. They are Mamdani's inference method (Mamdani, 1976) and Sugeno fuzzy approach (Takagi and Sugeno, 1985). Mamdani and Sugeno methods are similar according to the process of fuzzifying the inputs and applying the fuzzy operator. The difference between Mamdani method (Mamdani, 1976) and Sugeno approach (Takagi and Sugeno, 1985) is that the Sugeno output is a function either linear or constant as follows.

If x is A and y is B then $f(x,y)$ or the output is $f(x,y)$ = ax + by + c. Then $f(x,y)$ = c if a,b=0

Mamdani's method is more widely adopted and the output is a fuzzy set according to simple structure of 'min-max' operations such as:

If x is A and y is B then z is C, where C is a fuzzy set

FRI should be simple and easy to be modified by adding or deleting rules in order to allow imprecise data in fuzzy sets to represent inferred contextual information or situations. The following section describes the adaptation process by using the fuzzy logic principle to define contextual information for the implementation of routing region method in WSNs.

6.5 Situation Aware and Self-Adaptive Routing Region

As discussed in Section 6.1, there are cases when applications do not require an increase in the radius of the routing region (i.e. during normal situations). A small routing region can preserve resources as a smaller zone only involves fewer nodes being involved in data routing. But, when the traffic becomes congested, and many nodes in the region fail due to energy depletion, a larger region is required to include other nodes that are located farther from the source nodes into the data delivery process. A larger region certainly consumes higher network resources. There can be other scenarios in which it is important to adjust routing region algorithms considering not only the current context but also the resource availability level in order to maintain the applications accuracy requirements.

Most applications in pervasive systems utilise context to perform their tasks in an efficient manner and this makes context -awareness an essential requirement of these systems (van Bunningen et al., 2005; Davidyuk et al., 2004). However, most of the context-aware techniques (Gaber and Yu, 2006; Gaber et al., 2004) consider only one individual contextual parameter (e.g. temperature) in performing adaptation. To improve cost-efficiency and to provide a wider view of real-world situations, it is important to consider multiple contextual parameters (Padovitz et al., 2006, 2005). In the case of routing region in WSNs, for inferring current situations that drive the adaptation, there is a need to combine multiple parameters representing the surrounding environment in the WSN such as delay, network reliability and energy level through employing a situation modelling and reasoning technique (Padovitz et al., 2004). Knowledge of such situations that are derived from multiple context using inference methods can be used to perform adaptation for routing regions that better cope with changes in the environment of WSNs and to improve both cost-efficiency and QoS performances.

To provide routing region algorithms with a holistic adaptation approach that represents current situations based on minor changes of multiple contextual parameters such as delay, energy level and network reliability, I propose the Situation Aware and Self-Adaptive

Algorithm based Routing Regions (SASARR) scheme as a key contribution of this chapter. SASARR extends the FSI modelling and reasoning approach (Delir Haghighi et al., 2010; Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009; Delir Haghighi et al., 2008) to achieve situation awareness in wireless sensor networks and incorporates this extended approach into the adaptive routing region algorithm. The SASARR scheme aims to maximise the adaptation benefits by performing a smooth and gradual adaptation according to occurring situations to enhance the QoS performance and cost-efficiency.

To achieve situation awareness, the SASARR has two important components which include:

- Extended Fuzzy Situation Inference for Routing Region (FSIRR) technique
- Situation-Aware Adaptation Strategy for controlling Routing Region in an efficient and fine-grained manner.

The Fuzzy Situation Inference Routing Region (FSIRR) component is based on the FSI model and SARA framework (Situation and Resource-Aware Adaptation) (Delir Haghighi et al., 2010; Delir Haghighi, Zaslavsky, Krishnaswamy and Gaber, 2009) and is responsible for inferring occurring network-related situations from contextual information. The input data to the SASARR scheme includes WSNs QoS performance attributes obtained from the base station (BS)'s updates. This contextual data is used by the FSIRR to reason about the current situations using FSIRRs fuzzy rules which define situations for the network dynamics. The inferred situations as the results of the FSIRR rules are continuously then reported to the Adaptation Module to adjust the routing region accordingly. In the SASARR, situations represent real-world network dynamics such as normal, light-congested and congested.

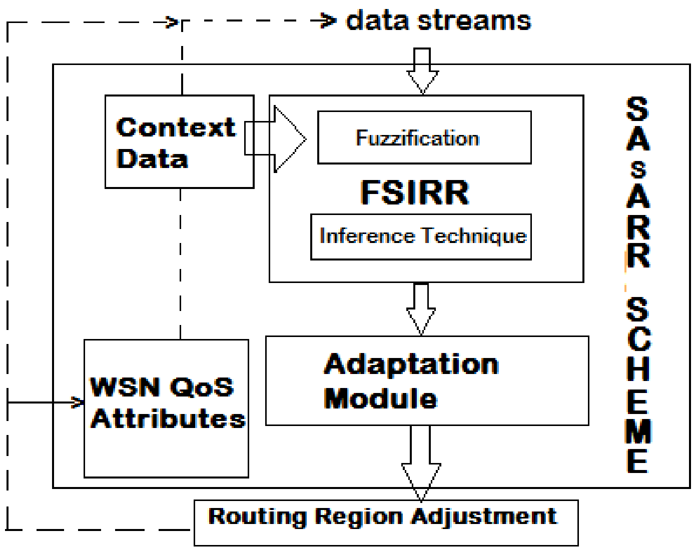

Figure 6.3: An overview of SASARR Scheme and its components

Figure 6.3 shows an overview of the SASARR and its components. The core of the SASARR scheme is the Adaptation Module that is responsible for adjusting the boundary of routing regions for guiding packet routing according to the FSIRRs situation inference results. This module integrates the FSIRR rules that include the defined situations into the strategies for the improved adaptations. The output of the SASARR scheme includes adjusted radius of the routing regions algorithm. These regions are used to control the packet routing from source nodes to a destination node. The following subsections discuss the components of SASARR in further detail.

6.5.1 Fuzzy Situation Inference for Routing Region (FSIRR)

As discussed in Section 6.3, one of the main objectives of the SASARR scheme is to provide situation-aware adaptation that adjusts the routing region based on the occurring situations. Towards this goal, there is a need for the SASARR scheme to use a context modelling and reasoning approach that infers situations from multiple contextual data/information obtained from the BS's update. This context reasoning approach is responsible for modelling situations and able to cater to minute changes when situations evolve and gradually transform into other situations. For example, when a situation changes from 'normal' to 'critical situation' (i.e. congested), the adaptation technique as a main component of the SASARR can use this information to gradually increase the routing region in order to create more routes for data routing as well as considering resource availability of WSNs. The FSIRR uses the fuzzy logic principle to define rules that are used to express the situations. The FSIRR is inspired by the Fuzzy Situation Inference (FSI) (Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009; Delir Haghighi et al., 2008) that integrates fuzzy logic principles into Context Spaces (CS) model (Padovitz et al., 2005, 2004), a formal and general context reasoning approach for pervasive computing. The following section describes the modules that are used by FSIRR to perform situation reasoning.

The FSIRR Modules

The Fuzzy Situation Inference Routing Region (FSIRR) approach performs situation modelling and reasoning using three main modules of the FSI approach: Fuzzifier, Rules and Situation Reasoner (Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009). Figure 6.4 shows the modules and the interaction between them.

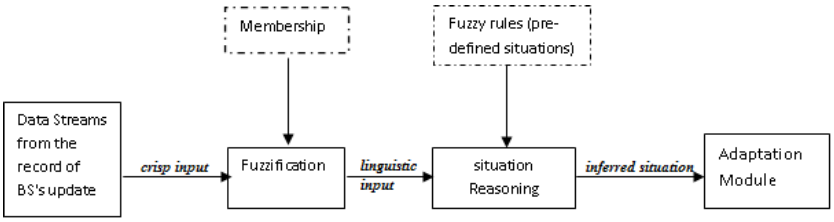

Figure 6.4: FSIRR Modules and Control Process
redrawn from (Cao et al., 2005)

As explained in Sub-Section 6.4.3, the fuzzification module is responsible for converting input values (i.e. crisp value from the BS's update) into fuzzy values by assigning them with membership degrees. The fuzzy rules includes a set of FSIRR rules where each rule represents a situation. The Situation Reasoning employs the FSIs reasoning method for inferring situations (Delir Haghighi et al., 2010). The reasoning results of FSIRR are inferred situations with degrees of confidence that are used by the Adaptation Module to drive adaptation of routing regions.

The first step to use FSIRR is defining linguistic variables and terms. In my SASARR scheme, linguistic variables include delay, energy, and delivery ratio (DR). In FSIRR, each variable associates with terms. The terms of linguistic variables take a name and parameters that are used during the fuzzification by the selected membership function to identify the fuzzy set boundaries. Even the selection of a membership function is subjective (Jang et al., 1997), according to the type of the application requirements in the SASARR, the FSIRR uses a trapezoidal membership function to map crisp values into fuzzy values during the fuzzification. For example, the variables of delay, energy and delivery ratio can be broken into the following terms:

Delay = ('low', 'middle', 'high')

Energy = ('low', 'middle', 'high')

Delivery Ratio = ('low', 'middle', 'high')

Each of the above terms such as low, middle or high, represents a fuzzy set.

Prior to performing situation inference, Figure 6.5 shows an example of defining the linguistic variables and their terms in FSIRR approach.

```
LinguisticVariable VarDelay = new LinguisticVariable ("delay", 0, 400);
Delay term1=new Delay('low', 0, 0, 195, 210);
Delay term1=new Delay('medium', 195, 210, 295, 310);
Delay term1=new Delay('high', 295, 310, 385, 400);
VarDelay.addDelay(term1);
VarDelay.addDelay(term2);
VarDelay.addDelay(term3);
Inferer.registerVar(var1);
```

Figure 6.5: An example of defining linguistic variables in FSIRR

All the defined linguistic variables are then registered with the Situation Reasoner. These variables and terms are used to define the FSI rules. When the linguistic variables are used for inferring a situation, each variable can have different levels of importance in representing a situation. In the FSI approach, this difference is modelled and represented by the concept of weights. Weights have values between 0 and 1 that are assigned to the linguistic variables based on their relative importance in representing a situation (Delir Haghighi et al., 2008). The weight of linguistic variables, for a situation adds up to 1 and is denoted as

$\sum_{i=1}^{n}\omega_i=1$ (6.1)

where ω_i is the weight.

During fuzzification, the results of fuzzifier are then applied to their associated FSI rules. In FSIRR, the format of rules is similar to the FSI method (Delir Haghighi, Zaslavsky, Krishnaswamy and Gaber, 2009; Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009; Delir Haghighi et al., 2008), which consists of multiple conditions that are joined with the AND operator.

An example of a FSIRR rule for reasoning the situation of the WSNs being 'congested' that is based on three context types (i.e. delay, energy level, reliability) can be expressed as:

If *delay is 'low'* AND *energy level is 'high'* AND *packet delivery ratio is 'high'* then situation is *'normal'*

The rule mentioned above has three conditions that is expressed by linguistic variables of delay and energy and delivery ratio with the terms of 'high'and 'low'. The first condition takes term 'low' fuzzy set, while the second and third conditions associate with two fuzzy sets of 'high'.

Table 6.1 presents examples of the FSIRR rules to define situations of 'normal', 'light-congested' and 'congested'. We have assigned weights of linguistic variables to conform to the weights specified for the context attributes.

Linguistic Variables	Terms
1 = Delay	low, medium, high
2 = Energy	low, medium, high
3 = DR	low, medium, high
Rule 1 : If delay is low and energy is high and DR is high then situation is normal Rule 2 : If delay is medium and energy is medium and DR is medium then situation is Light Congested Rule 3 : If delay is high and energy is low and DR is low then situation is congested	

Table 6.1: The definition of situations in FSIRR approach

After all the FSIRR rules are determined, they need to be registered with the Situation Reasoner. In FSIRR, the collected data about the current BS's updates of contextual attributes are used as an input for the FSIRR rules.

To reason about situations, FSIRR uses the FSI approach's main reasoning method (Delir Haghighi, Zaslavsky, Krishnaswamy and Gaber, 2009; Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009; Delir Haghighi et al., 2008) that is able to represent the occurrence of situations in uncertainty. Using the FSI method, the FSIRR can compute the individual contribution levels of context values to fuzzy sets. This contribution level reflects how well the context attributes relate to the modelled situation. The FSI reasoning method integrates the concepts of weights and contribution level to compute the confidence value (i.e. a value between 0 and 1) for the occurrence of a situation.

$$\sum_{i=1}^{n} \omega_i \mu(x_i) \quad (6.2)$$

where ω_i represents a weight assigned to a linguistic variable, and $(\mu x)_i$ denotes the membership degree of the element x_i given that it belongs to the fuzzy set associated with the linguistic variable. The result of $\omega_i(\mu x)_i$ represents a weighted membership degree of x_i and n represents the number of conditions in a rule ($1 \leq i \leq$ n). In FSIRR, I have adopted the same reasoning approach. To elaborate further on the above method, I show the computation of the confidence value for the occurrence of the situations in FSIRR using Equation 6.4. Table 6.2 presents an example of attribute values of delay, energy and delivery ratio/DR, and their respective membership degrees using a trapezoidal membership function.

The inference results in Table 6.2 indicate that situations from 'normal', 'light congested' and 'congested' are occurring with different certainty levels. Due to the strength of the FSI model in representing minute changes and transitions in situations, the FSIRR is able to identify minor changes in input values and reflect it in the inference results.

This shows that the FSIRR approach is able to provide more detailed inference results about the occurring situations and represent any minor changes in the network dynamics and situations. Capturing imprecise and uncertain context values and reflecting minor changes using the FSIRR approach enables a gradual and smooth adaptation in the SASARR scheme that is described in the following section.

6.5.2 Adaptation Module in SASARR

The Adaptation Module is responsible for the adjustment of the routing region's algorithm according to current situations of the network. The information regarding the occurring situations (i.e. the situation inference results) are passed into the Adaptation Module. Adaptation in the SASARR scheme is based on SARA (Situation and Resource Aware) framework (Delir Haghighi et al., 2010) that incorporates the FSI model. This framework has been successfully used for adaptation of data stream mining on mobile phones.

Situation	Condition	Values	Weights (ω)	Membership Degree(μ)
'Normal' (0.14)	Delay = Low Energy = High DR = High	320(ms) 72(%) 26(%)	0.45 0.2 0.35	0 0.7 0
Confidence = $\omega_{delay} \cdot \mu_{low}(302) + \omega_{energy} \cdot \mu_{high}(72) + \omega_{DR} \cdot \mu_{high}(26)$ $= (0.45 * 0.0) + (0.2 * 0.7) + (0.35 * 0.0) = 0.14$				
'Light Congested' 0.31	Delay = Medium Energy = Medium DR= Medium	302(ms) 72(%) 26(%)	0.48 0.15 0.37	0.55 0.3 0
Confidence = $\omega_{delay} \cdot \mu_{medium}(302) + \omega_{energy} \cdot \mu_{medium}(72) + \omega_{DR} \cdot \mu_{medium}(26)$ $= (0.48 * 0.55) + (0.15 * 0.3) + (0.37 * 0.0) = 0.31$				
'Congested' (0.55)	Delay = High Energy = Low DR= Low	302(ms) 72(%) 26(%)	0.45 0.2 0.35	0.45 0 1
Confidence = $\omega_{delay} \cdot \mu_{medium}(302) + \omega_{energy} \cdot \mu_{medium}(72) + \omega_{DR} \cdot \mu_{medium}(26)$ $= (0.45 * 0.45) + (0.2 * 0) + (0.35 * 1) = 0.55$				

Table 6.2: The computation of confidences of situations in FSIRR

The Adaptation Module in SASARR borrows the situation-aware strategy of the SARA framework and applies it to controlling the area of routing to determine an optimal routing region in resource constrained wireless sensor networks. The adaptation takes into account occurring situations and multiple parameters including energy level, routing response time and reliability that are periodically reported based on the BS's updates. This adaptation strategy can maximise the benefit of routing region adaptation as it controls the routing region according to any minor changes in situations. The following section explains further about the adaptation process in SASARR scheme.

SASARR Adaptation Process

One the main contributions of the SASARR scheme is extending the "situation-aware adaptation" of the SARA framework (Delir Haghighi et al., 2010) for adapting region boundary according to the network dynamics and current situations. In order to perform adjustment of region boundary, the Adaptation Module needs to be provided with the initialised value of the boundaries for each defined situation. Generally, I use the transmission range (Rn) of the sensor node (Xiao Hui et al., 2011) as the initialised region boundary, and a basis for computing the initialised boundary value for each situation. For example, in my experiment, the transmission range (R_n) is 30 meter according to the network graph. The values of the initialised boundary (I_b) are determined for each situation and reflect the accuracy needs of routing region algorithms according to their corresponding situations.

Let S = (S_1, S_2,... S_i) presents a set of situations where $S_i \in$ S. The value of an initialised region boundary for a situation Si can be calculated as follows.

$$I_b(S_i) = C_i R_n \quad (6.3)$$

where I_b denotes the initial value of a routing region boundary for the situation S_i, C_i as a constant for the situation S_i, and R_n represents the transmission range of a sensor node. With respect to equation 6.3, in order to provide appropriate initialised values for the region boundaries, I define the constant (C_i) for each situation based on the best value according to the experimental tests.

Situation	Constant	Initialised Boundary (m)
Normal	0.6	18
LightCongested	1	30
Congested	1.5	45

Table 6.3: Region Boundary Constant for FSIRR Situations

The situation-aware adaptation takes into consideration all the occurring situations and their respective membership degrees in the adaptation process to represent a smooth adaptation. As a result, this improves QoS performances and energy efficiency. Situation Inference results consist of multiple situations and the final region boundary value is calculated by using the weighted average function (Zimmermann, 2001). The following presents the weighted average formula of the SARA framework (Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009) that has been adopted and extended in the FSIRR's adaptation module for adjusting the routing region boundary:

$$\hat{B}r = \sum_{i=1}^{n} \frac{\mu(si)lb(S_i)}{\sum_{i=1}^{n} \mu(S_i)} \quad (6.4)$$

where $\mathrm{I}_b(S_i)$ represents the initialised value of a region boundary for the situation S_i and $\mu(s_i)$ denotes the membership degree of situation S_i, where 1 ≤ i ≤ n and n represents the number of situations. The $\hat{B}r$ symbol represents adjusted value of the region boundary as the result of adaptation.

According to the situation inference results in Table 6.3, by using the equation 6.4, I can determine the routing region boundary as:

$$\frac{\hat{B}r{=}(0.14*18)+(0.31*30)+(0.55*45)}{(0.14+0.31+0.55)} = 36.57 \quad (6.5)$$

The final adaptation result is used to adjust the region boundary that controls the area of routing. This value reflects the minute changes in current situations of the network. However, if I apply the context values as shown in Table 6.3 using the Rule-based Learning Adaptation Approach for Routing Regions (RuLeARR) scheme as described in Chapter 5, the calculation result shows that the region boundary will be adjusted to the initial value for situation of LightCongested that is set to 30. This value is calculated without considering network dynamics (i.e. energy level, delay and delivery ratio), and therefore it will not provide an efficient region boundary for the situations that are occuring in the network. Incorporating FSIRR, the proposed SASARR scheme enables the algorithm to take into account multiple QoS parameters in the adaptation process and provide a situation-aware and fine-grained adjustment of the routing region. The following section presents the analysis of the experimental results of SASARR and RuLeARR schemes.

6.6 Method of Simulation

In this section, I discuss my design method of the simulation. This method includes a set of simulation performance metrics and a simulation model to analyse the proposed schemes. The method of simulation helps designers and engineers to evaluate the behavior of the schemes under different conditions. In this section, simulation metrics are defined to quantify the performances of the routing schemes, and the simulation model represents the details about the experiment environments and parameters. A comparative evaluation is performed in order to compare SASARR and RuLeARR with regards to the efficiency and effectiveness of the adaptation approach in controlling routing regions. We first discuss the simulation performance metrics that are used to measure the effectiveness of the schemes for improved and efficient routing regions.

6.6.1 Performance Metric

The purpose of this evaluation is first to compare the adaptation approach of SASARR with the FSIRR model against RuLeARR that uses simple rule-based method for adjusting the routing region in terms of efficiency. The second aim is to demonstrate the situation-aware adaptation in representing minor changes of situations in the adaptation of routing region. The following describes the metrics in order to establish a standard set of evaluation criteria with respect to addressing the aforementioned aims. These metrics would be used to measure and capture the performances of different routing region schemes.

1. Adaptation to the changes. This metric represents the accuracy and the efficiency of both schemes in adjusting the routing region based on the changes of QoS attribute values.

2. The energy efficiency is determined by the rate of node participation in data delivery, the energy consumption in a region that is determined by residual energy of all nodes in the region and the number of failed nodes with respect to energy depletion.

3. The ratio of packet delivery ratio reflects the network utilisation and the reliability of the schemes. This metric counts the ratio of receiving packets the BS to packets originally sent by a source node.

4. Average packet delay is the average one-way time latency for all packets transmitting from a source to destination (BS). This metric reflects the response time of the routing schemes.

Following the defined metrics discussed above, I design the following model for the experimental works in order to ensure the evaluation of the routing schemes would be conducted as my purpose.

6.6.2 Simulation Model

To assess the effectiveness of the routing region schemes on the simulation performance, I have built and developed the FSIRR module written in Java language in order to run experimental works for the adaptation of routing regions. The FSIRR module enables situation-aware adaptation of routing regions through simulation works, and this new module enhances the capability of my modular WSN-specific simulator. Various experiments are performed to compare the following: i) adaptation performance according to changes of the situations due to network dynamic, ii) energy consumption and network lifetime, iii) distribution of energy consumption due to ‘energy hole problem’, iv) delivery ratio, v) heterogeneous traffic and vi) latency (end-to-end delay).

In this simulation, in order to get a good comparative evaluation between the two different routing region techniques, I construct the network graph with random node deployment where six nodes are located at the edge of the network. We generate traffic with various rates and deliver the packets using multi-hop routing to the BS which is placed in the middle of the network. We run simulation for 300s where each node has initial energy (15J) and the period for Hello packet to exchange information between nodes is 8s. The following table lists the environments of simulation.

Parameter	Value
Propagation model	Free space
Node type	Stationary
Sensor Node	360 nodes
Traffic Rate	CBR
Data Rate	250 Kbps
Packet Size	1 Kb
MAC Type	CSMA-CA
Destination	Single BS
Source Type	Multiple/Random
Hello Cycle	8 Secs
Initial Energy	15 J
Transmission Range	30 m (fix)
Packet Deadline (C1, C2)	(300 ms)

Table 6.4: System Properties

6.7 Analysis of the simulation results

This section presents and analyses the results of the extensive simulation experiments of SASARR and RuLeARR schemes. The experiments were performed to compare the effectiveness of adaptation techniques, the energy efficiency, even distribution of energy consumption, reliability, the support for heterogeneous traffic, reduced delay and improvement of network lifetime between the two different routing region schemes. The results of all experiments are described below. The first simulation experiment evaluates the effectiveness of the adaptation techniques of SASARR against RuLeARR.

6.7.1 Ability of the adaptation method in representing changes

In order to evaluate the ability of the two adaptation techniques in representing changes of the network dynamic under different traffic congestions, I monitored and recorded the size of the routing region periodically during the schemes' adaptation. The results are depicted in Figures 6.6, 6.7 and 6.8. The graphs show that both approaches have a relatively similar adaptation results due to changes of contexts (routing region QoS values) under different network conditions. The region increased due to the increase of traffic flowing in the network. This describes that both schemes are able to reconfigure the region according to dynamic and constrained-energy networks. However, when traffic becomes congested, the rule-based approach provides maximum routing region (60m) while the region in FSIRR model rises gradually but stays below the maximum region. This is important for keeping the packet relay up and running.

At a pre-defined condition, the results of SASARR and RuLeARR approaches almost overlap as they are based on a similar technique for adjusting the routing region. However, when the changes occur in context attributes including delay, energy and delivery ratio due to uncertain situations (i.e. situations that are not pre-defined); differences in adaptation for routing region between SASARR and rule-based learning adaptation are easily noticed. However, Figure 6.6 depicts that the SASARR scheme initiates the adjustment of region boundary at the early period of simulation compared to RuLeARR. This reflects the ability of SASARR by using FSIRR approach to capture minute changes of network dynamics. On the other hand, the RuLeARR performs adaptation according to changes later or in a sharper manner when the QoS values exceeds or drops below a defined limit (threshold). The adaptation in RuLeARR may not represent real-world situations of the network.

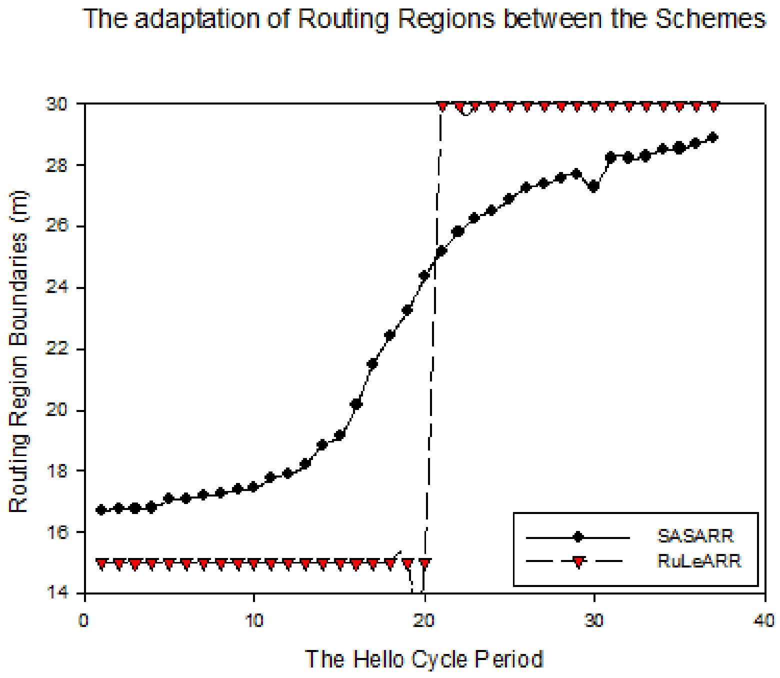

Figure 6.6: The adaptation of SASARR and RuLeARR under normal traffic

Compared to SASARR, the adaptation by RuLeARR represents sudden rises with sharp slopes under frequent situation changes occurring in the network as depicted in Figure 6.7 and Figure 6.8. The sharp changes in adaptation will not match the real world changes of current routing region QoS values. This is because the rule-based learning approach does not include a reasoning model that deals with minor changes of values and is not able to support gradual adaptation when one situation is evolving to another situation.

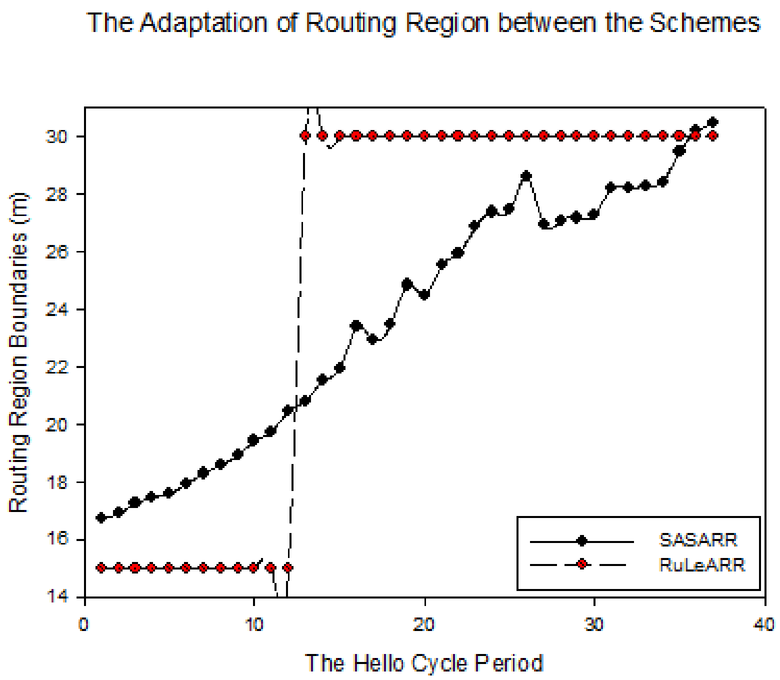

Figure 6.7: The adaptation between SASARR and RuLeARR under light congestion

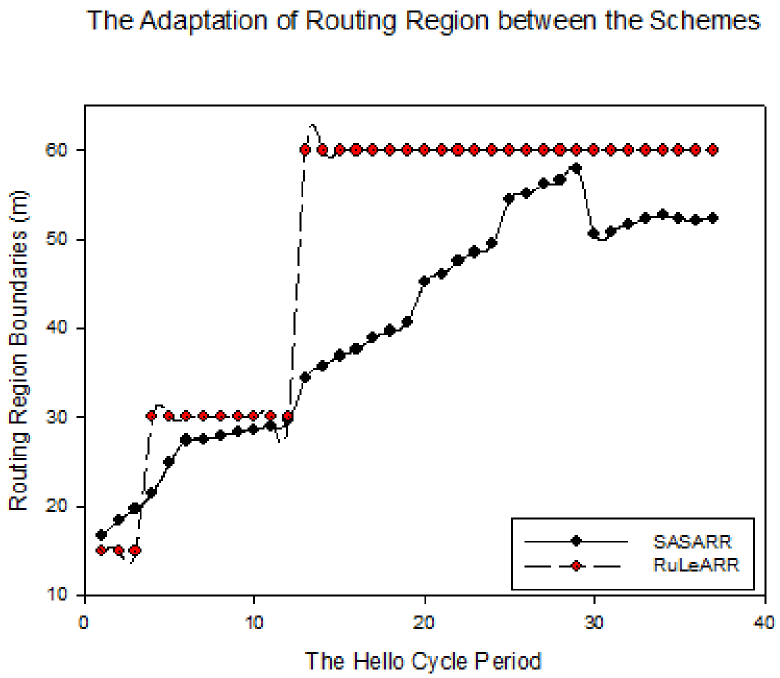

Figure 6.8: The sharp adaptation of RuLeARR compared to SASARR under heavy congestion

Figure 6.7 presents that RuLeARR suddenly increases its region to (30 meters) before half time of the simulation is reached while SASARR increases its region gradually til nearly (30 meters) after the whole period of simulation. At this time, RuLeARR defines that the BS's updates of QoS values decreases below the threshold for the smaller region of packet routing. Further, under heavy traffic (50 pps), after the halfway point of experimentation, RuLeARR adjusts its region to the highest (60 meters) faster than SASARR as it is shown in Figure 6.8. In contrast, SASARR based on FSIRR is able to represent vague and dynamic situations associated with WSN QoS performances by responding to marginal (little) changes in the values and achieve higher accuracy in modeling the real situations that are defined from the network dynamics. Figure 6.8 shows that when SASARR is used, the changes in routing region are gradual and smooth, even when the network is heavily congested.

The evaluation shows that SASARR with the FSIRR method proves to be more fitting approach compared to the non-fuzzy algorithm (RuLeARR) in adjusting the routing region according to network conditions. The figures show the strength of SASARR in enabling fine-grained and smooth adaptation according to the minor changes of routing region QoS values.

6.7.2 Effect on energy efficiency

In this sub-section, I discuss the strengths between SASARR and RuLeARR in achieving energy efficiency with respect to the rates of node participation, failures, and residual energy. The gradual and smooth adaptation of SASARR as presented in Figures 6.6 - 6.8 results in proper number of nodes to be involved in the region for packet routing compared to RuLeARR approach (see Figure 6.9). Under no or light congestion, the rule based learning adaptation approach has about 15% higher rate of node participation than SASARR. The rate increases nearly 30% when the traffic becomes more congested. The situation explains that under heavy congestion, SASARR performs adaptation by increasing the region gently performing fewer new nodes participating in the network so that the other nodes outside the region will go in an energy saving state and become inactive. As a result, nodes in SASARR can maintain higher residual energy than nodes in the RuLeARR approach under different congestion scenarios with even fewer nodes being involved in the network. The fall in the node participation rate describes the efficiency of SASARR in employing the routing region adaptation compared to RuLeARR .

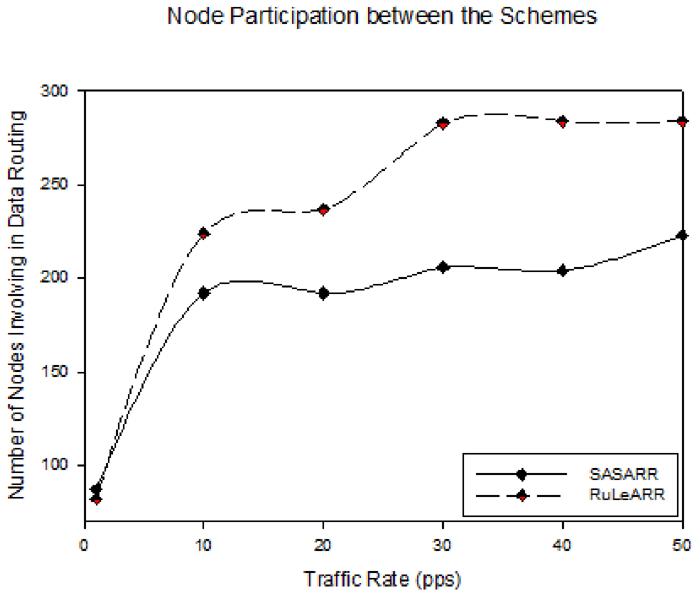

Figure 6.9: The efficiency of SASARR in using nodes in the network for employing adaptations compared to RuLeARR

Figure 6.10 presents the remaining energy of the nodes between SASARR and RuLeARR as the non-fuzzy approach. Figure 6.10 shows that SASARR achieves about 18% more residual energy than nodes in the opposite algorithm.

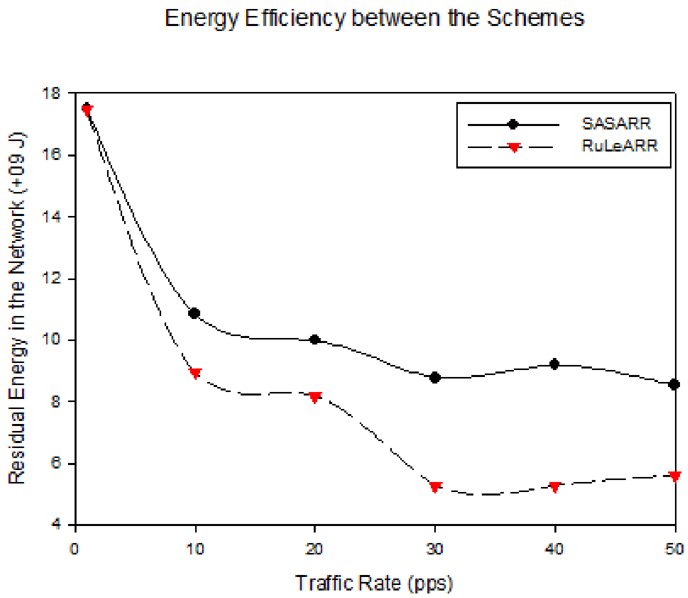

Figure 6.10: Energy efficiency of both adaptation approaches under different congestion levels

The rate increases significantly to about 40% under heavy congestion traffic. It is apparent that congestion causes both schemes to adapt by increasing the region for packet routing which adds more nodes in delivering packets. Both approaches automatically increase their routing region when congestion occurs in the network as depicted in Figure 6.9.

Another advantage of using the FSIRR model is represented by Figure 6.11 where the gradual adaptation in adjusting the area of routing provides fewer nodes failing during the simulation.

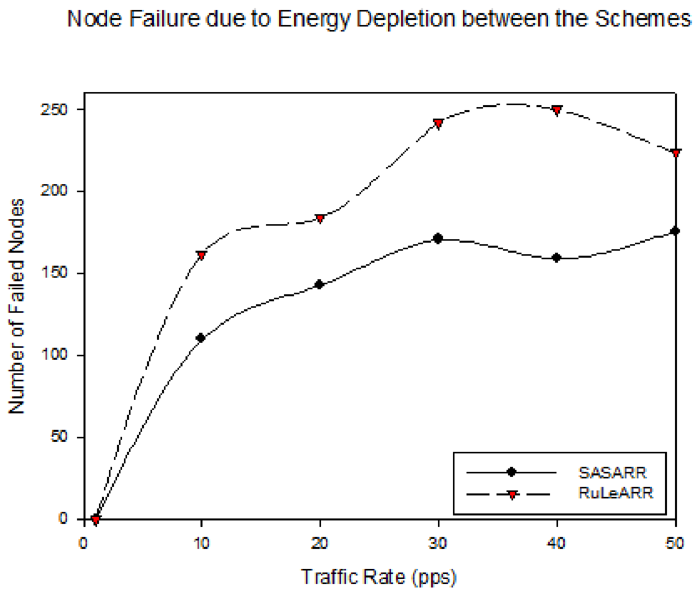

Figure 6.11: The comparative results between both adaptation approaches in terms of energy consumption under different congestion levels

Under normal traffic, for both schemes, node failure did not occur. However, when traffic becomes congested, both schemes encounter nodes failures. Under different congestion levels, SASARR provides lower failure, about 20% - 30% lower than the corresponding scheme. This is because the adaptation in the proposed scheme based on FSIRR represents minor changes of routing region QoS values to achieve higher efficiency in adjusting the routing region. In contrast, RuLeARR, as a non fuzzy approach increases the routing zone sharply to the maximum region (60m) when higher traffic is injected in the network. This results in inefficient energy consumption.

Figure 6.11 demonstrates that congestion causes the rate of failure to increase significantly for both schemes. Under a lighter congestion, SASARR provides 70% nodes that are still alive while the non fuzzy approach (RuLeARR) has about 55%. The rate decreases to nearly 60% for SASARR and 40% for RuLeARR when traffic becomes more congested. Finally, under heavy congestion, the SASARR still maintains more than 50% of nodes to stay active whereas the non-fuzzy algorithm drops the rate to about 30%.

The cause of increase in the failure of nodes in the RuLeARR is because this algorithm (approach) cannot deal with minor changes of values and is not able to make gradual changes in the adaptation of the routing region. Figures 6.9 - 6.11 show that SASARR presents better performance than the non-fuzzy approach (RuLeARR) in terms of energy consumption and preservation.

6.7.3 Effect on increasing traffic

This section discusses the comparative evaluation between SASARR and RuLeARR based on how they increase the traffic and address the 'energy hole problem'. As plotted in Figure 6.8, when traffic is heavily congested (50 pps), the rule-based learning adaptation approach wastes more than 50% of time to expand to and use the maximum routing region (60m) thereby increasing the number of nodes being used and then failing due to energy depletion. In contrast, SASARR approach spends most of time to route packets using the non-maximum regions for routing. This results in creating a network topology as shown in Figure 6.12 and Figure 6.13 where many nodes in both schemes are participating in more than one routing region. Heavy traffic results in significant node failures for both schemes because many nodes that are located in more than one region need to forward more packets for a number of source nodes (particularly for nodes around the BS which have to relay much higher traffic).

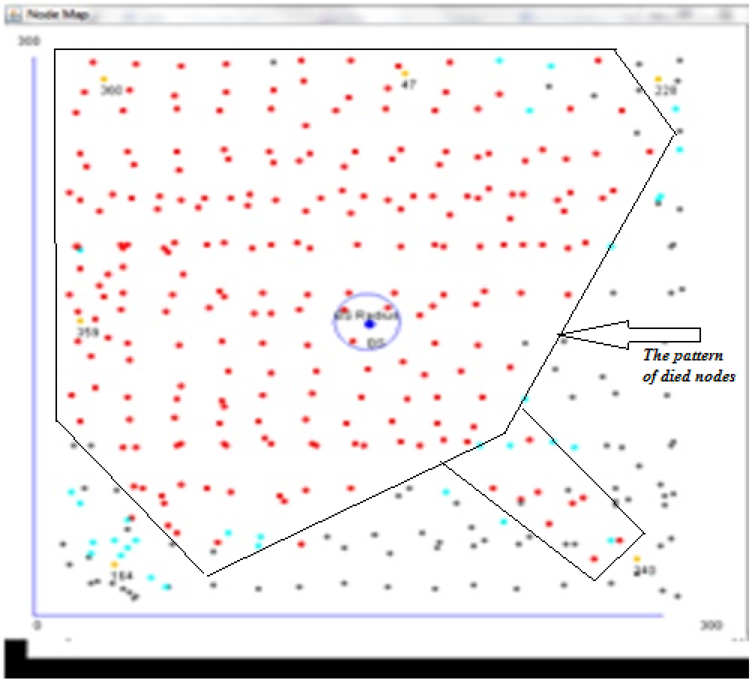

Figure 6.12: The performance of RuLeARR topology under heavy congestion

Figures 6.12 and 6.13 scenarios produce results given in Figure 6.11, which confirms that the larger region in the non-fuzzy approach (RuLeARR) may achieve higher node failure than the smaller and gradual adaptation of routing zone in SASARR. Figures 6.12 and 6.13 show red dots indicating nodes having less energy than required threshold. The light blue dots are active nodes while the grey dots are inactive nodes.

Figure 6.13 shows that SASARR can avoid driving many nodes to energy depletion in a wider area of the network but the non fuzzy technique cannot avoid it as shown in Figure 6.12.

Similarly, when the traffic is normal or heavy, SASARR results in less node failure than the opponent algorithm, even many nodes in SASARR are associated with multiple routing regions. This is because SASARR results in a smaller region area with gradual adaptation. A smaller region decreases node participation and can save more energy as many nodes located out of the region will go to the inactive mode. On the contrary, the non-fuzzy approach is not able to adjust routing region values based on minor changes and therefore the region area increases sharply and involves many nodes joining the region for data delivery. This mechanism certainly fails to save energy. Therefore, it is apparent that the proposed routing region approach (i.e. SASARR) can provide more efficient routing.

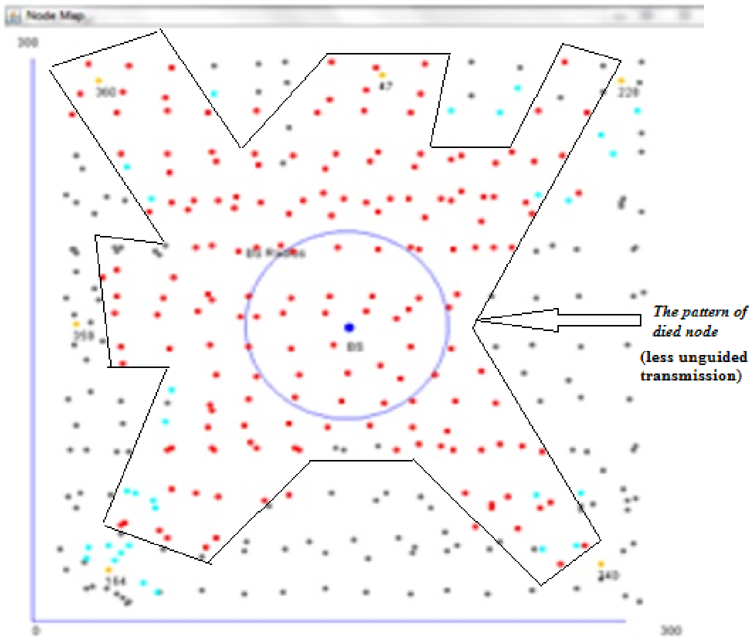

Figure 6.13: The performance of SASARR with FSIRR topology under heavy congestion

6.7.4 Effect on average packet delivery ratio

In here, I evaluate the reliability of both approaches in terms of packet delivery ratio. Figure 6.14 shows that the ratio of packet delivery decreases significantly as more packets are injected into the network. Since most nodes around the BS fail when the traffic is too high, the next packet transmission will not reach the BS so the ratio of packet delivery drops significantly. However, the simulation demonstrates that SASARR can maintain nodes' energy longer than the non fuzzy scheme so that SASARR can deliver a slightly higher amount of packets to the BS as compared to the corresponding algorithm (approach) as depicted in Figure 6.14.

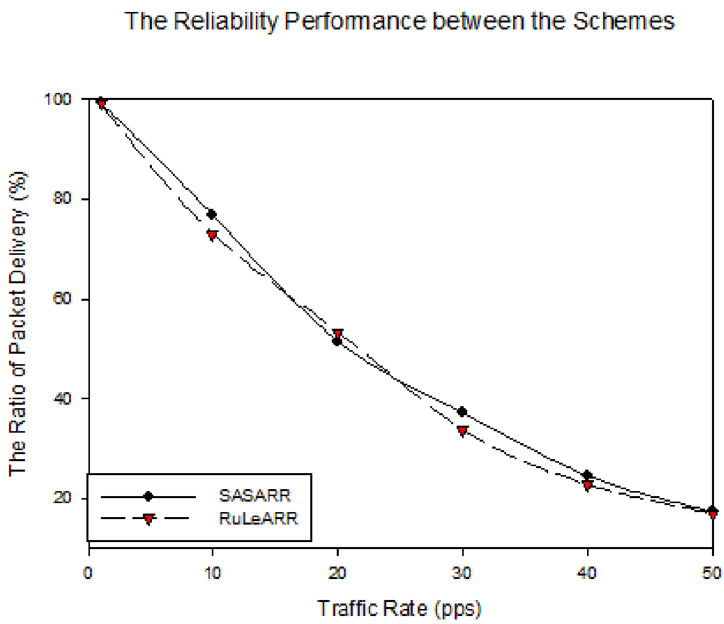

Figure 6.14: Packet delivery ratio under different congestion levels

It is noteworthy that the adaptation in SASARR provides more appropriate approach which maximises the node usage and the reliability of the routing protocol. The gradual and smooth adaptation provided by SASARR utilises the network efficiently to achieve better routing efficiency. In contrast, an abrupt adaptation in the non fuzzy scheme involves more nodes to participate in data delivery and it cannot distribute the packet transmission evenly amongst the nodes. As a result, the non fuzzy approach involves more nodes which are inactive, even the nodes located inside the region. The proposed scheme which does not exploit the fuzzy method cannot optimise the node utilisation. The simulation validates that SASARR achieves slightly better throughput (see Figure 6.15), which correlate to provide higher ratio of packet delivery compared to the non-fuzzy approach (RuLeARR). The figures present that SASARR is a relatively more robust system than RuLeARR to employ adaptation of routing region for higher data capacity at the BS.

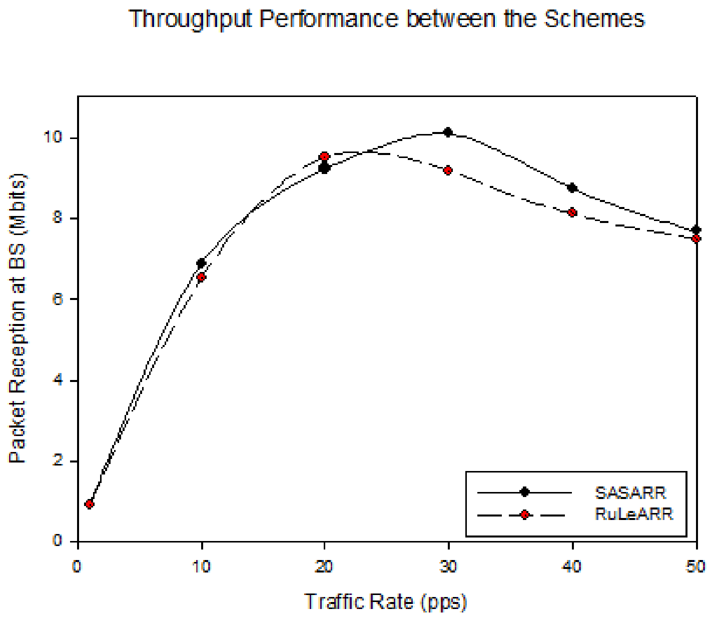

Figure 6.15: The performance of both approaches in terms of throughput

6.7.5 Effect of using a fitness function method

This section evaluates the performance of SASARR in adopting the fitness function method for supporting heterogeneous traffic in terms of throughput and transmission delay between the real-time (C1 and C2) and non real-time data (C3 and C4).

Figure 6.16 depicts the benefits of the proposed SASARR scheme to support heterogeneous traffic. As expected, the throughput of the total packets of C1 and C3 which require less tolerance to packet loss is higher than C2 and C4 packets which do not require reliability in data transmission. Under different levels of congestion, up to 20% of C1 and C2 packets reached the BS in comparison to other types of packets. This means that the buffer policy that allocates more space for C1 and C3 packets can incur less packet loss. When more traffic is injected in the network, lots of C2 and C4 packets are dropped because the nodes are overwhelmed and they have no more space to keep these types of packets. At similar situation, some C1 and C3 packets will be able to reach the BS because these packets are not dropped but queued in a buffer before being sent to the BS causing the throughput of these data types to increase. The fitness approach that considers the quality of the nodes as well as the requirements of each type of traffic helps the nodes to

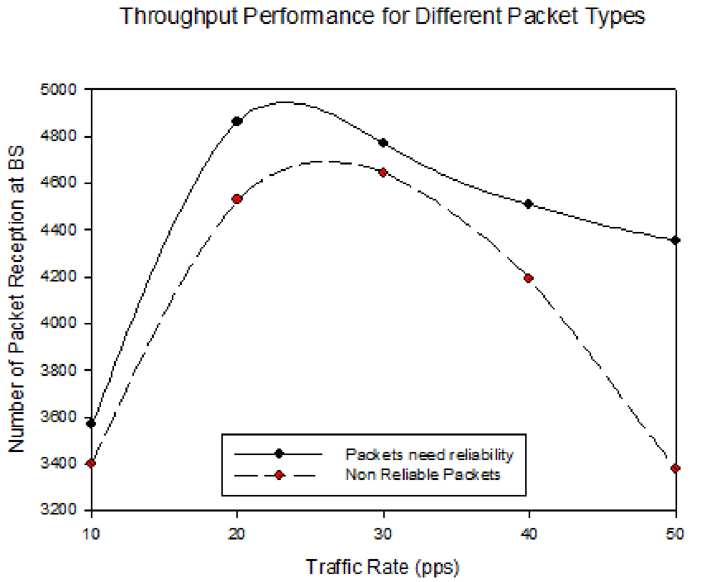

Figure 6.16: Data capacity comparative at the BS for heterogeneous traffic

send the packets out of the buffer quickly, thus the buffer always has a space for holding next incoming packets. This will increase the number of C1 and C3 packets arriving to the BS as shown in Figure 6.16.

It is apparent that the queuing module assigns higher priority to deliver critical packets (C1 and C2) first than non real-time packets (C3 and C4), which causes the real time packets achieves lower end-to-end delay than the non real-time traffic (see Figure 6.17). However the delay differences between the real time packets and the non real-time data categories are small.

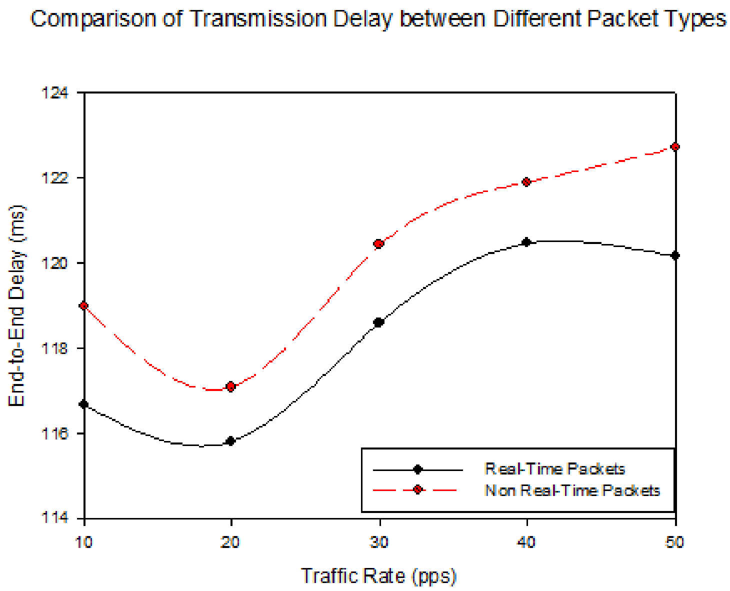

Figure 6.17: Average delay of SASARR between real time and non real time packets under different congestion levels

This can be explained that many non real-time packets and non-reliable data traffic (in this case is C4) are lost, fewer packets travel full path from source to destination and the dropped packets just take small number of hops so this leads to less delay. Hence, the sum of average end to end delay for non real-time packets will not be far from the real-time data.

6.7.6 Effect on the end-to-end delay

Figure 6.18 shows the results of experiments for effect of end-to-end delay between SAsAAR and RuLeARR. As the figure shows SASARR reduces the average one-way latency of data transmission from the source to the BS for all data categories.

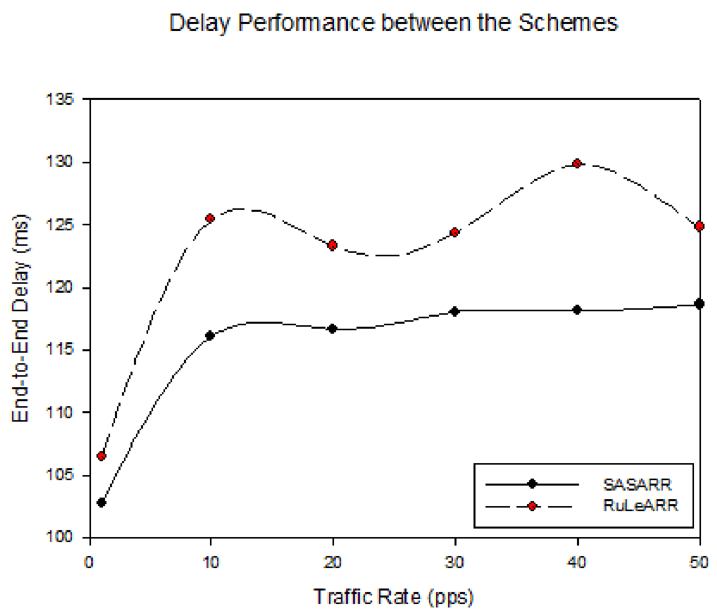

Figure 6.18: Average one-way latency for data transmission under various congestion levels

The proposed SASARR scheme can lower delay 5-10(%) as compared to the non-fuzzy approach (RuLeARR) under different congestion levels. The lower average delay reflects faster response time for SASARR enabling packet routing. Again, the accuracy in adaptation benefits the SASARR to define the proper region that controls the area of packet routing. A proper region in SASARR reduces the path of data routing that lessens delay transmission as well as decreases packet detouring which causes unexpected communication costs. Therefore, the proposed SASARR scheme based on the FSI model saves power consumption as shown in Figures 6.9 - 6.11.

6.8 Summary

We have proposed a new situation-aware routing region scheme, named SASARR for wireless sensor networks. The proposed scheme is the enhancement of the previous routing region (RuLeARR) approach, and it integrates situation awareness into the adaptation of routing region by taking into account multiple parameters for accurate sensor allocations, real network conditions, minor changes in energy efficiency, routing response time and reliability. In doing so, I extend the existing situation modelling and reasoning approach, named Fuzzy Situation Inference (FSI) (Delir Haghighi et al., 2010) to design a novel Fuzzy Situation Inference for Routing Regions (FSIRR) based on fuzzy logic theorems to present uncertain situations and the minute changes of context for re-positioning the region boundaries for individual communication sessions. In here, the context has been represented as residual energy, average delay transmission and average success delivery ratio of the network. The consideration of using the FSIRR model in SASARR aims to improve the adaptation to deal with the occurring situation in representing the vagueness (incomplete) changes of context to drive a smooth adaptation for adjusting routing regions with a higher level of accuracy.

We compare both routing region schemes through experimental/simulation works to evaluate the strengths and the benefits of the SASARR compared to the RuLeARR. The simulation results show that SASARR can adjust the routing region in a gradual and fine-grained manner and reduce margin of sensor coverage range that decreases resource wastage. Moreover, the obtained results show that SASARR ensures better adaptation for improved QoS performances according to reliability (delivery ratio), response time (delay), heterogeneous traffic support and energy saving which is the main concern in WSNs. This makes the enhanced approach more suitable for WSNs to meet energy-efficient routing protocols.

Having presented my new QoS differentiating routing based Fitness scheme and its extended approaches such as TReNs, RuLeARR and SASARR for addressing the issues of 'energy hole problem' or 'packet detouring' due to the nature of data communication in WSNs while considering energy efficiency in chapters 3, 4, 5 and 6, I have now completed the discussion of the contributions of this thesis. The following chapter concludes the works of the thesis.

Chapter 7

Conclusion

In the preceding chapters I have proposed, developed, implemented and evaluated the proposed differentiating QoS routing scheme that includes three enhancement approaches to cope with the problems of energy hole and data flooding due to the communication characteristics in WSNs, and one of WSNs challenges that is to be address is the need for conserving energy without losing accuracy for routing applications. This chapter concludes the dissertation and its contributions and discusses future directions of this work.

7.1 Summary of the research works

Over the last few years, the development of sensor networks has gained increasing importance due to their potential to support challenging research in a wide range of applications. Certainly, applications that consume different data types require different levels of data quality and accordingly might have different priorities in terms of data transmission and routing (Monowar et al., 2012). The recent approaches thus far do not make a more comprehensive traffic classification and perform a clear route selection for a higher number of applications in WSNs that have different types of data traffic. They only focus on a limited number of aspects of QoS for supporting heterogeneous traffic in WSNs. Further, those approaches have not been designed to solve the potential problems caused by the nature of multi-hop communication in WSNs that significantly affect the performances of the network.

This thesis, however, argues that the conducted research should contribute to design a new localised routing scheme that considers different QoS services in terms of energy, latency and reliability combined in a generic approach to deal with the different application requirements. The proposed scheme provides a new traffic classification to differentiate and fulfil the required QoS for each type of data traffic for supporting WSN's applications, and tailors adaptation of route selection based on differentiating QoS requirements to suit application specific needs.

In doing so, I define a set of evaluation metrics to evaluate my proposed QoS scheme compared to the established QoS routing algorithm (SPEED) from different aspects. As a result of this comparative evaluation, this thesis shows that the differentiation QoS routing scheme has the highest potential to support heterogeneous traffic in WSNs while considering energy efficiency, lengthening lifetime, increasing reliability and reducing latency.

Furthermore, to enhance and improve the performance of the proposed differentiation QoS routing scheme to contend with various constraints due to the characteristic of data communication models in WSNs, three enhancements/adaptations were proposed in order to address the problems of i) energy hole problem due to uneven energy depletion, ii) unguided packet relay or packet detouring due to link instability caused by WSN's MAC operation and iii) less efficiency and accuracy according to frequent changes of the network topology and QoS network parameters. The following describes all the extended approaches.

In Chapter 3, I have presented the proposed TeGAR, a new differentiation QoS routing scheme. The proposed TeGaR takes into account the traffic diversity to provide customised QoS metrics for each traffic category, and it provides different QoS paths for different data requirements using Fitness Function based Genetic Algorithm. The fitness function considers the quality of each node (i.e. energy, reliability, latency and other communication parameters) that fits the requirements of each data category to enable a differentiation routing using different QoS metrics for factoring in different application requirements. To perform adaptation of route selection according to different data traffic, the fitness function computes some attributes of the nodes from the updated neighbouring entries in order to provide the fitness degree of the node due to the traffic category.

In addition, the traffic classification in TeGaR enables categorising the required data-related QoS to each type of traffic based on data prioritisation, and the data classification is designed in a modular approach to ensure exactly the required QoS metrics in a more efficient way. Moreover, the proposed scheme uses multi-queuing that classifies the priority of packets into critical, real-time and non real-time queues, and assigns higher priority for critical and real time packets to meet the given deadline. Hence, this policy may satisfy multiple QoS requirements while still maximizing throughput for all traffic classes. TeGaR uses geographical information for routing that avoids the need of propagating information and does not implement multi-path approach to increase reliability thereby saving more energy. To enhance the capabilities of TeGaR in reducing the 'energy hole problem' in WSNs, I have also proposed an extended routing approach aiming to evenly distribute network load, named Transmission Range Extension based on the neighbourhood Size (TReNs)(Rizal et al., 2012).

Chapter 4 was devoted to describe the mechanism and evaluation of TReNs. TReNs is inspired by the idea that each node is allowed to adjust and extend its transmission range for data delivery (Rizal et al., 2012; Jarry et al., 2006) when its forwarding neighbourhood size decreases below the limit. Through incorporating an analytical model with respect to the 'energy hole problem', the TReNs approach is able to provide a balanced power distribution that reduces uneven energy consumption and further lengthens the function of network resulting in an improved network performance. The other routing problem according to the inherent characteristic of routing in WSNs, termed data flooding due to unguided data transmission is presented in Chapter 5 and 6.

Chapter 5 proposed Rule-based Learning Adaptation Approach for Routing Regions (RuLeARR) that uses the information of locations of nodes to guide packet routing to improve route optimisation and restrict data flooding that consumes much energy. The ability of RuLeARR is to control the area (region) of routing for individual communication sessions by defining a dynamic radius for every routing region using a self-adaptive algorithm based on the QoS values. In order to improve a higher level of accuracy and granularity to drive the adaptation process of routing regions, I provide Situation Aware and Self-Adaptive Algorithm Based Routing Region (SASARR) with fuzzy logic that is presented in Chapter 6. Through incorporating fuzzy logic based adaptation that is extended from Fuzzy Situation Inference Model (FSI) (Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009; Delir Haghighi et al., 2008), SASARR is able to capture the minute changes of context and represent uncertain and vague situations regarding network performances and perform a fine-grained adaptation of routing regions.

We have conducted and presented a comprehensive evaluation of all routing approaches in Chapter 3, 4, 5 and 6 to demonstrate the strengths of the proposed QoS routing protocol that includes TeGaR, TReNs, RuLeARR and SASARR schemes under different congestion levels. The proposed approaches in Chapter 3 and 4 are compared to SPEED (He et al., 2003), a prominent QoS routing algorithm that has been widely used in WSNs which uses localised information and considers QoS factors. The simulation results showed that my routing approaches outperform SPEED in terms of multiple QoS metrics of throughput, delivery ratio, energy efficiency, network utilisation and latency for the application specific needs. In addition, using TReNs in Chapter 4, the results demonstrated the benefits of the extended approach to reduce the occurrance of 'energy hole problem' with the improvement of the data capacity delivery as well as the network lifetime. Further, the second extended approach in Chapter 5 (RuLeARR) performed well in addressing the packet detouring due to the nature of multihop routing in WSNs. Finally, SASARR as the enhancement approach to RuLeARR, is able to control the length/size of routing region by representing minor changes of the QoS performance metrics of the nodes inside the region in terms of energy level, average packet delay and data delivery ratio thereby improving both the cost-efficiency and the network performance. Following all the results, I can conclude that my QoS routing protocol can satisfy many application's requirements in WSNs that pose different types of data traffic to support heterogeneous traffic and the design of the three extended approaches are able to address the common problems caused by the nature of routing and the design constraints of WSNs.

Based on the summary of the research as presented in Section 7.1, I describe the outcome of the thesis and its contributions in Section 7.2 and further directions of this thesis research in Section 7.3.

7.2 Research Outcome and Contribution

Following is the summary of the research described above, I confirm the outcome of the thesis as follows.

- An improved QoS differentiating routing scheme was developed. This scheme provides benefits to ensure the required QoS metrics for certain type of traffic according to the application-specific needs in an energy-aware way. The proposed scheme can be used to design the realisation of the applications of node heterogeneity in which specific nodes may have different roles and functionalities related to the implementation of different applications.

- The proposed scheme with its extended approaches ensures a self-configuration feature that can help the nodes to adapt to topology changes and to maintain the network running for performing the tasks (i.e. sensing, computing and routing) in unattended and constrained energy network.

- The implementation of routing region approach advantages the proposed scheme to design a new routing that considers a trade-off between energy consumption and communication overhead reduction while improving the network reliability.

- The integration of the situation awareness into the research works enhances the benefit of the proposed QoS differentiating routing to provide situation-aware routing. This routing is able to capture the minor changes in the value of the network attributes thereby improving the QoS performances while it still conserves energy consumption.

The mentioned-above outcomes clearly demonstrated a new routing protocol that includes four different approaches (i.e. TeGaR, TReNs, RuLeARR and SASARR) to make a significant improvement to the efficiency and effectiveness of QoS routing in WSNs. The specific contributions are presented as follows:

- In heterogeneity of sensors, where sensors collect and transmit heterogeneous traffic having different kinds of data traffic that is based on differentiating of QoS requirements, my proposed novel routing can be as a generic approach/scheme that encompasses all the important attributes of QoS for factoring in different application requirements in WSNs while considering energy efficiency, latency and reliability. The framework has components of (i) traffic classification; (ii) route selection using Fitness Function; and (iii) Priority based Delivery with Queuing Manager.

- In the sensing networks that pose some routing challenges due to the unique nature of data transmission, my proposed framework confirmed effective solutions in coping with the challenges of both 'energy hole problem' and 'broadcast flooding'by implementing two strategies. They are (i) non-uniform transmission range approach that benefits the framework to balance the power consumption that reduces the occurring of energy hole problem due to uneven energy depletion; and (ii) routing region strategy, which uses the information of geographical of nodes to guide packet routing in a certain routing region to restrict data flooding that consumes much energy while improving route optimisation.

- In the network environments with sudden and uncertain changes, my adaptation framework based on Fuzzy Situation Inference (FSI) (Delir Haghighi, Zaslavsky, Krishnaswamy, Gaber and Loke, 2009; Delir Haghighi et al., 2008) that uses fuzzy logic principles to model and reason of situations is able to capture the minor changes in the network parameters in WSNs.

The TReNs in the research works has been published in peer-refereed international conference. Having discussed the principal outcomes and contributions of this research works, I now outline future directions of this work.

7.3 Research Direction

We have developed and evaluated the QoS routing framework that provides several routing approaches described previously for addressing the objectives of the research. Although my novel adaptation framework have been exerted so far on the research aims and objectives of the thesis, there are some challenges that need to be considered for further research directions. The future works are as follows:

- the feasibility and viability of the proposed approaches have been proven through initial experimental work. To validate the applicability and feasibility of the proposed approaches in real situations the presented techniques can be implemented into prototypes to provide a basis for real testing works,

- the development and evaluation of the proposed approaches I discussed in this thesis have been validated through a custom-made Java based simulator. It is suitable for the very early stage of the design and development of a new protocol for wireless sensor networks. Next, the implementation and evaluation of the approaches by using one of a well-known and widely used WSN simulator is required to make the experiments more repeatable and verifiable,

- as my proposed approaches have not been designed for specific application purposes, further research can be applied to implement and evaluate of the proposed approaches for some application domains such as i) Body Area Sensor Networks for healthcare monitoring system, ii) Mobile Sensor Networks for intelligent transportation (vehicular) system, iii) Home automation, and etc. Thus, the approaches can be designed as generic approaches.

Given the contributions and directions of this research, I have demonstrated the benefits of adaptation based situational context for QoS routing algorithm in WSNs. In summary, this thesis has made significant contribution to QoS routing algorithm in WSNs through the development and validation of the TeGAR, TReNs, RuLeARRR and SASARR.

References

Adipat, B. and Zhang, D. (2003). A real-time adaptive traffic monitoring approach for multimedia content delivery in wireless environment, *IEEE International Conference on Systems, Man and Cybernetics, 2003*, Vol. 1, pp. 280–285 vol.1.

Ahmed, M., Dao, S. and Katz, R. (2001). Positioning range extension gateways in mobile ad hoc wireless networks to improve connectivity and throughput, *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, Vol. 1, pp. 331–335 vol.1.

Akkaya, K. and Younis, M. (2003). An energy-aware QoS routing protocol for wireless sensor networks, *Proc. of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003*, pp. 710–715.

Akyildiz, I., Melodia, T. and Chowdury, K. (2007). Wireless multimedia sensor networks: A survey, *Wireless Communications, IEEE* **14**(6): 32–39.

Akyildiz, I., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002). Wireless sensor networks: a survey, *Computer Networks* **38**(4): 393 – 422.
URL: *http://www.sciencedirect.com/science/article/pii/S1389128601003024*

Al-Karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey, *Wireless communications, IEEE* **11**(6): 6–28.

Al-Karaki, J., Ul-Mustafa, R. and Kamal, A. (2004). Data aggregation in wireless sensor networks - exact and approximate algorithms, *Workshop on High Performance Switching and Routing, 2004. HPSR*, pp. 241–245.

Alanazi, T. and Ouni, R. (2013). Traffic differentiation and scheduling for QoS support in vehicular sensor networks, *International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2013*, pp. 34–39.

Almalkawi, I. T., Zapata, M. G., Al-Karaki, J. N., Morillo-Pozo, J., Almalkawi, I., Zapata, M., Al-Karaki, J., Morillo-Pozo, J. et al. (2010). Wireless multimedia sensor networks: current trends and future directions, *Sensors (Basel, Switzerland)* **10**(7): 6662.

Amer, S. H. and Hamilton, Jr., J. A. (2008). Dsr and tora in fixed and mobile wireless networks, *Proceedings of the 46th Annual Southeast Regional Conference on XX*, ACM-SE 46, ACM, New York, NY, USA, pp. 458–463.
URL: *http://doi.acm.org/10.1145/1593105.1593226*

Andersen, J., Rappaport, T. and Yoshida, S. (1995). Propagation measurements and models for wireless communications channels, *Communications Magazine, IEEE* **33**(1): 42–49.

Ares, B. Z., Park, P. G., Fischione, C., Speranzon, A. and Johansson, K. H. (2007). On power control for wireless sensor networks: System model, middleware component and experimental evaluation, *In European Control Conference*.

Bagula, A. and Mazandu, K. (2008). Energy constrained multipath routing in wireless sensor networks, *in* F. Sandnes, Y. Zhang, C. Rong, L. Yang and J. Ma (eds), *Ubiquitous Intelligence and Computing*, Vol. 5061 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 453–467.
URL: *http://dx.doi.org/10.1007/978-3-540-69293-5$_3$6*

Barry, M. (n.d.). Vhf/uhf/microwave radio propagation: A primer for digital experimenters, *Ottawa.[Links]* .

Basagni, S., Chlamtac, I., Syrotiuk, V. R. and Woodward, B. A. (1998). A distance routing effect algorithm for mobility (dream), *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '98, ACM, New York, NY, USA, pp. 76–84.
URL: *http://doi.acm.org/10.1145/288235.288254*

Basaran, C. and Kang, K.-D. (2009). Quality of service in wireless sensor networks, *Guide to Wireless Sensor Networks*, Springer, pp. 305–321.

Bellur, B. and Ogier, R. (1999). A reliable, efficient topology broadcast protocol for dynamic networks, *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 1, pp. 178–186 vol.1.

Ben-Othman, J. and Yahya, B. (2010). Energy efficient and QoS based routing protocol for wireless sensor networks, *Journal of Parallel and Distributed Computing* **70**(8): 849 – 857.
URL: *http://www.sciencedirect.com/science/article/pii/S0743731510000341*

Braginsky, D. and Estrin, D. (2002). Rumor routing algorthim for sensor networks, *Proceedings of the 1st ACM international workshop on Wireless Sensor Networks and Applications*, ACM, pp. 22–31.

Cao, J., Xing, N., Chan, A., Feng, Y. and Jin, B. (2005). Service adaptation using fuzzy theory in context-aware mobile computing middleware, *Proceedings. 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2005.*, pp. 496–501.

Capkun, S., Hamdi, M. and Hubaux, J.-P. (2001). Gps-free positioning in mobile ad-hoc networks, *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001.*, pp. 10 pp.–.

Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M. and Zhao, J. (2001). Habitat monitoring: Application driver for wireless communications technology, *SIGCOMM Comput. Commun. Rev.* **31**(2 supplement): 20–41.
URL: *http://doi.acm.org/10.1145/844193.844196*

Chang, J.-H. and Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks, *IEEE/ACM Transactions on Networking* **12**(4): 609–619.

Chen, D. and Varshney, P. K. (2004). QoS support in wireless sensor networks: A survey, *International Conference on Wireless Networks*, pp. 227–233.

Chen, H., Li, J. and Mohapatra, P. (2004). Race: time series compression with rate adaptivity and error bound for sensor networks, *IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2004.*, pp. 124–133.

Chen, Y. and Nasser, N. (2008). Enabling QoS multipath routing protocol for wireless sensor networks, *IEEE International Conference on Communications, 2008. ICC '08.*, pp. 2421–2425.

Chen, Y. and Zhao, Q. (2005). On the lifetime of wireless sensor networks, *Communications Letters, IEEE* **9**(11): 976–978.

Cheverst, K., Byun, H., Fitton, D., Sas, C., Kray, C. and Villar, N. (2005). Exploring issues of user model transparency and proactive behaviour in an office environment control system., *User Model. User-Adapt. Interact.* **15**(3-4): 235–273.
URL: *http://dblp.uni-trier.de/db/journals/umuai/umuai15.htmlCheverstBFSKV05*

Chung, Y. and Song, S.-H. (2009). Implementation of low-voltage static ram with enhanced data stability and circuit speed, *Microelectron. J.* **40**(6): 944–951.
URL: *http://dx.doi.org/10.1016/j.mejo.2008.11.063*

Cordeiro, C. D. M. and Agrawal, D. P. (2006). *Ad Hoc & Sensor Networks: Theory And Applications*, World Scientific Publishing Co., Inc., River Edge, NJ, USA.

Crawley, E., Nair, R., Rajagopalan, B. and Sandick, H. (1998). Rfc 2386: A framework for qos-based routing in the internet, *Technical report*, IETF.
URL: *http://www.ietf.org/rfc/rfc2386.txt*

Daabaj, K., Dixon, M. and Koziniec, T. (2009). Experimental study of load balancing routing for improving lifetime in sensor networks, *5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09.*, pp. 1–4.

Dallas, D. P. and Hanlen, L. W. (2009). Optimal transmission range and node degree for multi-hop routing in wireless sensor networks, *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, PM2HW2N '09, ACM, New York, NY, USA, pp. 167–174.
URL: *http://doi.acm.org/10.1145/1641913.1641937*

Davidyuk, O., Riekki, J., mikko Rautio, V. and Sun, J. (2004). Context-aware middleware for mobile multimedia applications, *in MUM 04: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia*, ACM Press, pp. 213–220.

Deb, B., Bhatnagar, S. and Nath, B. (2003). Reinform: Reliable information forwarding using multiple paths in sensor networks, *Proceedings. 28th Annual IEEE International Conference on Local Computer Networks, 2003. LCN'03.*, IEEE, pp. 406–415.

Delir Haghighi, P., Gaber, M., Krishnaswamy, S. and Zaslavsky, A. (2010). Situation-aware adaptive processing (saap) of data streams, *in* A.-E. Hassanien, J. H. Abawajy, A. Abraham and H. Hagras (eds), *Pervasive Computing*, Computer Communications and Networks, Springer London, pp. 313–338.
URL: *http://dx.doi.org/10.1007/978-1-84882-599-4*$_1$4

Delir Haghighi, P., Krishnaswamy, S., Zaslavsky, A. and Gaber, M. M. (2008). Reasoning about context in uncertain pervasive computing environments, *Smart Sensing and Context*, Springer, pp. 112–125.

Delir Haghighi, P., Zaslavsky, A., Krishnaswamy, S. and Gaber, M. (2009). Mobile data mining for intelligent healthcare support, *42nd Hawaii International Conference on System Sciences, 2009. HICSS '09.*, pp. 1–10.

Delir Haghighi, P., Zaslavsky, A., Krishnaswamy, S., Gaber, M. M. and Loke, S. (2009). Context-aware adaptive data stream mining, *Intelligent Data Analysis* **13**(3): 423–434.

Dey, A. K. (2001). Understanding and using context, *Personal Ubiquitous Computing* **5**(1): 4–7.
URL: *http://dx.doi.org/10.1007/s007790170019*

Dharmaraju, D., Roy-Chowdhury, A., Hovareshti, P. and Baras, J. (2002). Inora-a unified signaling and routing mechanism for qos support in mobile ad hoc networks, *Proceedings. International Conference on Parallel Processing Workshops, 2002.*, pp. 86–93.

Djenouri, D. and Balasingham, I. (2011). Traffic-differentiation-based modular qos localized routing for wireless sensor networks, *IEEE Transactions on Mobile Computing* **10**(6): 797–809.

Doherty, L., Warneke, B. A., Boser, B. E. and Pister, K. (2001). Energy and performance considerations for smart dust, *International Journal on Parallel and Distributed Systems and Networks* **4**(3): 121–133.

Dulman, S., Nieberg, T., Wu, J. and Havinga, P. (2003). Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks, *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, Vol. 3, pp. 1918–1922 vol.3.

Dunkels, A., Alonso, J. and Voigt, T. (2003). Making tcp/ip viable for wireless sensor networks, *SICS Research Report* .

Dutta, P., Grimmer, M., Arora, A., Bibyk, S. B. and Culler, D. E. (2005). Design of a wireless sensor network platform for detecting rare, random, and ephemeral events, *IPSN*, pp. 497–502.

Ee, C. T. and Bajcsy, R. (2004). Congestion control and fairness for many-to-one routing in sensor networks, *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, ACM, New York, NY, USA, pp. 148–161.
URL: *http://doi.acm.org/10.1145/1031495.1031513*

Ennaji, R. and Boulmalf, M. (2009). Routing in wireless sensor networks, *International Conference on Multimedia Computing and Systems, 2009. ICMCS '09.*, pp. 495–500.

Essa, I. (2000). Ubiquitous sensing for smart and aware environments, *Personal Communications, IEEE* **7**(5): 47–49.

Fapojuwo, A. and Cano-Tinoco, A. (2009). Energy consumption and message delay analysis of qos enhanced base station controlled dynamic clustering protocol for wireless sensor networks, *IEEE Transactions on Wireless Communications,* **8**(10): 5366–5374.

Felemban, E., Lee, C.-G. and Ekici, E. (2006). Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks, *IEEE Transactions on Mobile Computing,* **5**(6): 738–754.

Floyd, S. and Jacobson, V. (1995). Link-sharing and resource management models for packet networks, *IEEE/ACM Trans. Netw.* **3**(4): 365–386.
URL: *http://dx.doi.org/10.1109/90.413212*

Gaber, M. M., Krishnaswamy, S. and Zaslavsky, A. (2004). Cost-efficient mining techniques for data streams, *In Proc. Australasian Workshop on Data Mining and Web Intelligence (DMWI2004)*, Australian Computer Society, Inc, pp. 109–114.

Gaber, M. M. and Yu, P. S. (2006). Detection and classification of changes in evolving data streams, *International Journal of Information Technology & Decision Making (IJITDM)* **5**(04): 659–670.
URL: *http://ideas.repec.org/a/wsi/ijitdm/v05y2006i04p659-670.html*

Ganesan, D., Govindan, R., Shenker, S. and Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks, *SIGMOBILE Mob. Comput. Commun. Rev.* **5**(4): 11–25.
URL: *http://doi.acm.org/10.1145/509506.509514*

Ganz, A., Ganz, Z. and Wongthavarawat, K. (2004). *Multimedia Wireless Networks: Technologies, Standards, and QoS*, Prentice Hall PTR communications engineering and emerging technologies series, Prentice Hall PTR.
URL: *http://books.google.com.au/books?id=m4dQAAAAMAAJ*

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Goodman, D. (1997). *Wireless personal communications*, The Addison-Wesley Wireless Communications Series, Addison-Wesley Longman.
URL: *http://books.google.com.au/books?id=srhvo0TPSlEC*

Gottwald, S. (2010). An early approach toward graded identity and graded membership in set theory, *Fuzzy Sets Syst.* **161**(18): 2369–2379.
URL: *http://dx.doi.org/10.1016/j.fss.2009.12.005*

Goyal, D. and Tripathy, M. (2012). Routing protocols in wireless sensor networks: A survey, *Second International Conference on Advanced Computing Communication Technologies (ACCT), 2012*, pp. 474–480.

Gsottberger, Y., Shi, X., Stromberg, G., Sturm, T. and Weber, W. (2004). Embedding low-cost wireless sensors into universal plug and play environments, *in* H. Karl, A. Wolisz and A. Willig (eds), *Wireless Sensor Networks*, Vol. 2920 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 291–306.
URL: *http://dx.doi.org/10.1007/978-3-540-24606-0*$_2$0

Gu, L., Jia, D., Vicaire, P., Yan, T., Luo, L., Tirumala, A., Cao, Q., He, T., Stankovic, J. A., Abdelzaher, T. and Krogh, B. H. (2005). Lightweight detection and classification for wireless sensor networks in realistic environments, *in SenSys*, pp. 205–217.

Hamid, M., Alam, M. M. and seon Hong, C. (2008). Design of a QoS-aware routing mechanism for wireless multimedia sensor networks, *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–6.

Hanzo, L., I. and Tafazolli, R. (2007). A survey of QoS routing solutions for mobile ad hoc networks, *Communications Surveys Tutorials, IEEE* **9**(2): 50–70.

He, T., Krishnamurthy, S., Stankovic, J. A., Abdelzaher, T., Luo, L., Stoleru, R., Yan, T. and Gu, L. (2004). Energy-efficient surveillance system using wireless sensor networks, *In Mobisys*, ACM Press, pp. 270–283.

He, T., Stankovic, J., Lu, C. and Abdelzaher, T. (2003). Speed: a stateless protocol for real-time communication in sensor networks, *Proceedings. 23rd International Conference on Distributed Computing Systems, 2003.*, pp. 46–55.

Healy, M., Newe, T. and Lewis, E. (2008). Wireless sensor node hardware: A review, *Sensors, 2008 IEEE*, pp. 621–624.

Hedetniemi, S. M., Hedetniemi, S. T. and Liestman, A. L. (1988). A survey of gossiping and broadcasting in communication networks, *Networks* **18**(4): 319–349.

Heinzelman, W., Chandrakasan, A. and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks, *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000.*, pp. 10 pp. vol.2–.

Heinzelman, W. R., Kulik, J. and Balakrishnan, H. (1999). Adaptive protocols for information dissemination in wireless sensor networks, *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, ACM, New York, NY, USA, pp. 174–185.
URL: *http://doi.acm.org/10.1145/313451.313529*

Herring, C. and Kaplan, S. (2000). Component-based software systems for smart environments, *Personal Communications, IEEE* **7**(5): 60–61.

Hill, J. L. (2003). *System Architecture for Wireless Sensor Networks*, PhD thesis, Computer Science. AAI3105239.

Hoblos, G., Staroswiecki, M. and Aitouche, A. (2000). Optimal design of fault tolerant sensor networks, *Proceedings of the 2000 IEEE International Conference on Control Applications, 2000.*, pp. 467–472.

Huang, X. and Fang, Y. (2008). Multiconstrained qos multipath routing in wireless sensor networks, *Wirel. Netw.* **14**(4): 465–478.
URL: *http://dx.doi.org/10.1007/s11276-006-0731-9*

Hull, B., Jamieson, K. and Balakrishnan, H. (2004). Mitigating congestion in wireless sensor networks, *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM, pp. 134–147.

Intanagonwiwat, C., Govindan, R. and Estrin, D. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks, *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, ACM, New York, NY, USA, pp. 56–67.
URL: *http://doi.acm.org/10.1145/345910.345920*

Iyengar, S., Jayasimha, D. N. and Nadig, D. (1994). A versatile architecture for the distributed sensor integration problem, *IEEE Transactions on Computers* **43**(2): 175–185.

Iyer, R. and Kleinrock, L. (2003). QoS control for sensor networks, *IEEE International Conference on Communications, 2003. ICC '03.*, Vol. 1, pp. 517–521 vol.1.

Iyer, S., Rao Kompella, R. and Shelat, A. (2001). Classipl: an architecture for fast and flexible packet classification, *Network, IEEE* **15**(2): 33–41.

Jang, J., Sun, C. and Mizutani, E. (1997). Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review], *Automatic Control, IEEE Transactions on* **42**(10): 1482–1484.

Jarry, A., Leone, P., Powell, O. and Rolim, J. (2006). An optimal data propagation algorithm for maximizing the lifespan of sensor networks, *Proceedings of the Second IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'06, Springer-Verlag, Berlin, Heidelberg, pp. 405–421.
URL: *http://dx.doi.org/10.1007/11776178*$_2$5

Ji, P., Wu, C., Zhang, Y. and Jia, Z. (2008). Dast: A QoS-aware routing protocol for wireless sensor networks, *International Conference on Embedded Software and Systems Symposia, 2008. ICESS Symposia '08.*, pp. 259–264.

Jovanov, E., Milenkovic, A., Otto, C. and de Groen, P. (2005). A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation, *Journal of NeuroEngineering and Rehabilitation* **2**(1): 1–10.
URL: *http://dx.doi.org/10.1186/1743-0003-2-6*

Kandris, D., Tsagkaropoulos, M., Politis, I., Tzes, A. and Kotsopoulos, S. (2011). Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks, *Ad Hoc Netw.* **9**(4): 591–607.
URL: *http://dx.doi.org/10.1016/j.adhoc.2010.09.001*

Karp, B. and Kung, H.-T. (2000). Gpsr: Greedy perimeter stateless routing for wireless networks, *Proceedings of the 6th annual International Conference on Mobile Computing and Networking*, ACM, pp. 243–254.

Keeney, J. and Cahill, V. (2003). Chisel: a policy-driven, context-aware, dynamic adaptation framework, *IEEE 4th International Workshop on Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003.*, pp. 3–14.

Khamforoosh, K. and Khamforoush, H. (2009). A new routing algorithm for energy reduction in wireless sensor networks, *2nd IEEE International Conference on Computer Science and Information Technology, 2009. ICCSIT 2009.*, pp. 505–509.

Khanna, R., Liu, H. and Chen, H.-H. (2010). Self-organization of wireless sensor network for autonomous control in an it server platform, *IEEE International Conference on Communications (ICC), 2010*, pp. 1–5.

Kim, D., Kim, J. and Park, K. H. (2011). An event-aware mac scheduling for energy efficient aggregation in wireless sensor networks, *Comput. Netw.* **55**(1): 225–240.
URL: *http://dx.doi.org/10.1016/j.comnet.2010.08.010*

Kim, J. M. and Cho, T. H. (2007). Routing path generation for reliable transmission in sensor networks using ga with fuzzy logic based fitness function, *Computational Science and Its Applications–ICCSA 2007*, Springer, pp. 637–648.

Kleinrock, L. and Tobagi, F. (1975). Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics, *IEEE Transactions on Communications* **23**(12): 1400–1416.

Ko, Y.-B. and Vaidya, N. H. (2000). Location-aided routing (lar) in mobile ad hoc networks, *Wireless Networks* **6**(4): 307–321.

Kohvakka, M., Kuorilehto, M., Hännikäinen, M. and Hämäläinen, T. D. (2006). Performance analysis of ieee 802.15.4 and zigbee for large-scale wireless sensor network applications, *Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks*, PE-WASUN '06, ACM, New York, NY, USA, pp. 48–57.
URL: *http://doi.acm.org/10.1145/1163610.1163619*

Kordafshari, M., Pourkabirian, A., Faez, K. and Rahimabadi, A. (2009a). Energy-efficient speed routing protocol for wireless sensor networks, *Fifth Advanced International Conference on Telecommunications, 2009. AICT '09.*, pp. 267–271.

Kordafshari, M., Pourkabirian, A., Faez, K. and Rahimabadi, A. (2009b). Energy-efficient speed routing protocol for wireless sensor networks, *Fifth Advanced International Conference on Telecommunications, 2009. AICT '09.*, pp. 267–271.

Kreinovich, V., Quintana, C. and Fuentes, O. (1993). Genetic algorithms: What fitness scaling is optimal?, *Cybernetics and Systems: an International Journal* **24**: 9–26.

Kuhn, F., Wattenhofer, R. and Zollinger, A. (2003). Worst-case optimal and average-case efficient geometric ad-hoc routing, *Proceedings of the 4th ACM International Symposium on Mobile Ad hoc Networking & Computing*, ACM, pp. 267–278.

Kulik, J., Heinzelman, W. and Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks, *Wireless Netw., Springer-Verlag New York* **8**(2/3): 169–185.
URL: *http://dx.doi.org/10.1023/A:1013715909417*

Kumar, P., Singh, M. and Triar, U. (2012). A review of routing protocols in wireless sensor networks, *International Journal of Engineering Research Technology (IJERT)* **1**.

Li, J. and Mohapatra, P. (2007). Analytical modeling and mitigation techniques for the energy hole problem in sensor networks, Vol. 3, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, pp. 233–254.
URL: *http://dx.doi.org/10.1016/j.pmcj.2006.11.001*

Li, Q., Aslam, J. and Rus, D. (2001). Hierarchical power-aware routing in sensor networks, *In Proceedings of the DIMACS Workshop on Pervasive Networking*.

Lian, J., Naik, K. and Agnew, G. B. (2006). Data capacity improvement of wireless sensor networks using non-uniform sensor distribution., *IJDSN* **2**(2): 121–145.
URL: *http://dblp.uni-trier.de/db/journals/ijdsn/ijdsn2.htmlLianNA06*

Liang, X., Chen, M., Xiao, Y., Balasingham, I. and Leung, V. (2009). A novel cooperative communication protocol for QoS provisioning in wireless sensor networks, *5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009.*, pp. 1–6.

Liao, W.-H., Sheu, J.-P. and Tseng, Y.-C. (2001). Grid: A fully location-aware routing protocol for mobile ad hoc networks, *Telecommunication Systems, Springer* **18**(1-3): 37–60.

Lim, H. and Kim, C. (2001). Flooding in wireless ad hoc networks, *Computer Communications* **24**(34): 353 – 363.
URL: *http://www.sciencedirect.com/science/article/pii/S0140366400002334*

Lindsey, S. and Raghavendra, C. (2002). Pegasis: Power-efficient gathering in sensor information systems, *Aerospace Conference Proceedings, 2002. IEEE*, Vol. 3, pp. 3–1125–3–1130 vol.3.

Liu, Y., Ngan, H. and Ni, L. M. (2007). Power-aware node deployment in wireless sensor networks, *Int. J. Distrib. Sen. Netw.* **3**(2): 225–241.
URL: *http://dx.doi.org/10.1080/15501320701205597*

Lowen, R. (1976). Fuzzy topological spaces and fuzzy compactness, *Journal of Mathematical Analysis and Applications* **56**(3): 621 – 633.
URL: *http://www.sciencedirect.com/science/article/pii/0022247X76900299*

Lu, C., Blum, B. M., Abdelzaher, T. F., Stankovic, J. A. and He, T. (2002). Rap: A real-time communication architecture for large-scale wireless sensor networks, *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE*, IEEE, pp. 55–66.

Madkour, M., GHANAMI, D. E., Maach, A. and Hasbi, A. (2013). Context-aware service adaptation: An approach based on fuzzy sets and service composition., *Journal of Information Science & Engineering* **29**(1): 1–16.

Mahapatra, A., Anand, K. and Agrawal, D. P. (2006). QoS and energy aware routing for real-time traffic in wireless sensor networks, *Computer Communications* **29**(4): 437 – 445. Current areas of interest in wireless sensor networks designs.
URL: *http://www.sciencedirect.com/science/article/pii/S014036640500040X*

Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R. and Anderson, J. (2002). Wireless sensor networks for habitat monitoring, *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, ACM, New York, NY, USA, pp. 88–97.
URL: *http://doi.acm.org/10.1145/570738.570751*

Mamdani, E. (1976). Advances in the linguistic synthesis of fuzzy controllers, *International Journal of Man-Machine Studies* **8**(6): 669 – 678.
URL: *http://www.sciencedirect.com/science/article/pii/S0020737376800284*

Manjeshwar, A., Zeng, Q.-A. and Agrawal, D. P. (2002). An analytical model for information retrieval in wireless sensor networks using enhanced apteen protocol, *IEEE Trans. Parallel Distrib. Syst.* **13**(12): 1290–1302.
URL: *http://dx.doi.org/10.1109/TPDS.2002.1158266*

Martinez, J.-F., Garca, A.-B., Corredor, I., Lpez, L., Hernndez, V. and Dasilva, A. (2007). Modelling QoS for wireless sensor networks, *in* L. Orozco-Barbosa, T. Olivares, R. Casado and A. Bermdez (eds), *Wireless Sensor and Actor Networks*, Vol. 248 of *IFIP International Federation for Information Processing*, Springer US, pp. 143–154.
URL: *http://dx.doi.org/10.1007/978-0-387-74899-3*$_1$3

Medeiros, H., Park, J. and Kak, A. (2007). A light-weight event-driven protocol for sensor clustering in wireless camera networks, *First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC '07.*, pp. 203–210.

Mendel, J. (1995). Fuzzy logic systems for engineering: a tutorial, *Proceedings of the IEEE* **83**(3): 345–377.

Mohapatra, P., Li, J. and Gui, C. (2003). QoS in mobile ad hoc networks, *IEEE Wireless Communications* **10**(3): 44–53.

Monowar, M., Rahman, O., Pathan, A.-S. K. and Hong, C. S. (2012). Prioritized heterogeneous traffic-oriented congestion control protocol for wsns., *International Arab Journal of Information Technology (IAJIT)* **9**(1).

Murthy, S. and Garcia-Luna-Aceves, J. J. (1996). An efficient routing protocol for wireless networks, *Mob. Netw. Appl.,Springer-Verlag New York* **1**(2): 183–197.
URL: *http://dx.doi.org/10.1007/BF01193336*

Narayanan, D., Flinn, J. and Satyanarayanan, M. (2000). Using history to improve mobile application adaptation, *Third IEEE Workshop on Mobile Computing Systems and Applications, 2000*, pp. 31–40.

Norouzi, A., Babamir, F. S. and Zaim, A. H. (2011). A new clustering protocol for wireless sensor networks using genetic algorithm approach., *Wireless Sensor Network* **3**(11).

Novák, V., Perfilieva, I. and Močkoř, J. (1999). *Mathematical Principles of Fuzzy Logic*, The Springer International Series in Engineering and Computer Science, Springer US.
URL: *http://books.google.com.au/books?id=pJeu6Ue65S4C*

Olariu, S. and Stojmenovic, I. (2006). Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting, *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12.

Osadciw, L. and Veeramachaneni, K. (2007). Sensor network management through fitness function design in multi-objective optimization, *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, 2007. ACSSC 2007*, pp. 1648–1651.

Ota, N., Hooks, D., Wright, P., Auslander, D. and Peffer, T. (2003). Poster abstract: Wireless sensor network characterization - application to demand response energy pricing, *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, ACM, New York, NY, USA, pp. 334–335.
URL: *http://doi.acm.org/10.1145/958491.958547*

Othman, M. F. and Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system, *Procedia Engineering* **41**(0): 1204 – 1210. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).
URL: *http://www.sciencedirect.com/science/article/pii/S1877705812027026*

Padovitz, A., Loke, S. W. and Zaslavsky, A. (2004). Towards a theory of context spaces, *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004*, pp. 38–42.

Padovitz, A., Loke, S. W., Zaslavsky, A., Burg, B. and Bartolini, C. (2005). An approach to data fusion for context awareness, *Proceedings of the 5th International Conference on Modeling and Using Context*, CONTEXT'05, Springer-Verlag, Berlin, Heidelberg, pp. 353–367.
URL: *http://dx.doi.org/10.1007/11508373$_2$7*

Padovitz, A., Zaslavsky, A. and Loke, S. W. (2006). A unifying model for representing and reasoning about context under uncertainty, *In Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU*.

Park, C. and Jung, I. (2010). Traffic-aware routing protocol for wireless sensor networks, *International Conference on Information Science and Applications (ICISA), 2010*, pp. 1–8.

Park, J., Tyan, H.-R. and Kuo, C.-C. (2006). Internet traffic classification for scalable QoS provision, *IEEE International Conference on Multimedia and Expo, 2006*, pp. 1221–1224.

Park, V. D. and Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks, *Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, INFOCOM '97, IEEE Computer Society, Washington, DC, USA, pp. 1405–.
URL: *http://dl.acm.org/citation.cfm?id=839292.843010*

Patel, A. and Kulkarni, L. (2011). Qos parameter analysis on aodv and dsdv protocols in a wireless network, *International Journal of Communication Network & Security* **1**(1).

Pathan, A.-S., Heo, G. and seon Hong, C. (2007). A secure lightweight approach of node membership verification in dense hdsn, *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pp. 1–6.

Patil, M. and Biradar, R. (2012). A survey on routing protocols in wireless sensor networks, *18th IEEE International Conference on Networks (ICON), 2012*, pp. 86–91.

Peng, S., Yang, S. X., Gregori, S. and Tian, F. (2008). An adaptive QoS and energy-aware routing algorithm for wireless sensor networks, *International Conference on Information and Automation, 2008. ICIA 2008*, pp. 578–583.

Pucha, H., Das, S. M. and Hu, Y. C. (2007). The performance impact of traffic patterns on routing protocols in mobile ad hoc networks, *Computer Networks* **51**(12): 3595–3616.

Rabaey, J., Ammer, M., da Silva, J.L., J., Patel, D. and Roundy, S. (2000). Picoradio supports ad hoc ultra-low power wireless networking, *Computer, IEEE* **33**(7): 42–48.

Rahman, M., GhasemAghaei, R., El Saddik, A. and Gueaieb, W. (2008). M-iar: Biologically inspired routing protocol for wireless multimedia sensor networks, *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pp. 1823–1827.

Rizal, M. N., Gondal, I., Delir Haghighi, P. and Qiu, B. (2012). Priority based expansion of neighborhood size for heterogeneous traffic routing in wsn, *Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '12, ACM, New York, NY, USA, pp. 101–106.
URL: *http://doi.acm.org/10.1145/2387027.2387045*

Rodoplu, V., Member, S. and Meng, T. H. (1999). Minimum energy mobile wireless networks, *IEEE Journal on Selected Areas in Communications* **17**: 1333–1344.

Sadagopan, N., Krishnamachari, B. and Helmy, A. (2003). The acquire mechanism for efficient querying in sensor networks, *Proceedings of the First IEEE. 2003 IEEE International Workshop on Sensor Network Protocols and Applications, 2003*, pp. 149–155.

Salber, D., Dey, A. K. and Abowd, G. D. (1999). The context toolkit: Aiding the development of context-enabled applications, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, ACM, New York, NY, USA, pp. 434–441.
URL: *http://doi.acm.org/10.1145/302979.303126*

Savvides, A., Han, C.-C. and Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors, *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, ACM, New York, NY, USA, pp. 166–179.
URL: *http://doi.acm.org/10.1145/381677.381693*

Saxena, N., Roy, A. and Shin, J. (2009). Quest: a QoS-based energy efficient sensor routing protocol, *Wireless Communications and Mobile Computing* **9**(3): 417–426.

Schilit, B. N. and Theimer, M. M. (1994). Disseminating active map information to mobile hosts, *Netwrk. Mag. of Global Internetwkg.* **8**(5): 22–32.
URL: *http://dx.doi.org/10.1109/65.313011*

Schweitzer, F. and Tilch, B. (2002). Self-assembling of networks in an agent-based model, *Physical Review E* **66**(2): 026113.

Sen, J. and Ukil, A. (2009). An adaptable and qos-aware routing protocol for wireless sensor networks, *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009*, pp. 767–771.

Sha, K., Du, J. and Shi, W. (2006). Wear: a balanced, fault-tolerant, energy-aware routing protocol in wsns, *International Journal of Sensor Networks* **1**(3): 156–168.

Sha, K., Shi, W. and Watkins, O. (2006). Using wireless sensor networks for fire rescue applications: Requirements and challenges, *IEEE International Conference on Electro/information Technology, 2006*, pp. 239–244.

Shah, R. and Rabaey, J. (2002). Energy aware routing for low energy ad hoc sensor networks, *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, Vol. 1, pp. 350–355 vol.1.

Sivakumar, R., Sinha, P. and Bharghavan, V. (1999). Cedar: a core-extraction distributed ad hoc routing algorithm, *IEEE Journal on Selected Areas in Communications* **17**(8): 1454–1465.

Sohrabi, K., Gao, J., Ailawadhi, V. and Pottie, G. (2000). Protocols for self-organization of a wireless sensor network, *Personal Communications, IEEE* **7**(5): 16–27.

Stojmenovic, I. (2002). Position-based routing in ad hoc networks, *Communications Magazine, IEEE* **40**(7): 128–134.

Subramanian, L. and Katz, R. (2000). An architecture for building self-configurable systems, *First Annual Workshop on Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000*, pp. 63–73.

Sumathi, R. and Srinivas, M. (2012). A survey of qos based routing protocols for wireless sensor networks., *Journal of Information Processing Systems* **8**(4).

Sun, W., Yamaguchi, H., Yukimasa, K. and Kusumoto, S. (2006). Gvgrid: A QoS routing protocol for vehicular ad hoc networks, *14th IEEE International Workshop on Quality of Service, 2006. IWQoS 2006*, pp. 130–139.

Sun, Y., Ma, H., Liu, L. and Zheng, Y. (2008). Asar: An ant-based service-aware routing algorithm for multimedia sensor networks, *Frontiers of Electrical and Electronic Engineering in China* **3**(1): 25–33.
URL: *http://dx.doi.org/10.1007/s11460-008-0013-7*

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control, *Systems, Man and Cybernetics, IEEE Transactions on* **SMC-15**(1): 116–132.

Tilak, S., Abu-Ghazaleh, N. B. and Heinzelman, W. (2002). Infrastructure tradeoffs for sensor networks, *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, ACM, New York, NY, USA, pp. 49–58.
URL: *http://doi.acm.org/10.1145/570738.570746*

Tsai, H.-W., Chu, C.-P. and Chen, T.-S. (2007). Mobile object tracking in wireless sensor networks, *Computer Communications* **30**(8): 1811 – 1825.
URL: *http://www.sciencedirect.com/science/article/pii/S014036640700093X*

van Bunningen, A., Feng, L. and Apers, P. (2006). A context-aware preference model for database querying in an ambient intelligent environment, *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006)*, Vol. 4080 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, pp. 33–43. Also appeared as CTIT Technical Report TR-CTIT-06-17.
URL: *http://doc.utwente.nl/63694/*

van Bunningen, A., Feng, L. and Apers, P. M. G. (2005). Context for ubiquitous data management, *International Workshop on Ubiquitous Data Management, 2005. UDM 2005*, pp. 17–24.

Vass, D., Vincze, Z., Vida, R. and Vidcs, A. (2006). Energy efficiency in wireless sensor networks using mobile base station, *in* C. Kloos, A. Marn and D. Larrabeiti (eds), *EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World*, Vol. 196 of *IFIP International Federation for Information Processing*, Springer US, pp. 173–186.
URL: *http://dx.doi.org/10.1007/0-387-31170-X*$_1$3

Vidhyapriya, R. and Vanathi, P. T. (2007). Energy aware routing for wireless sensor networks, *International Conference on Signal Processing, Communications and Networking, 2007. ICSCN '07*, pp. 545–550.

Wang, Y.-H., Mao, H.-J., Tsai, C.-H. and Chuang, C.-C. (2005). Hmrp: Hierarchy-based multipath routing protocol for wireless sensor networks, *in* T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai and L. Yang (eds), *Embedded and Ubiquitous Computing EUC 2005 Workshops*, Vol. 3823 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 452–459.
URL: *http://dx.doi.org/10.1007/11596042*$_4$7

Warneke, B. and Pister, K. (2002). Mems for distributed wireless sensor networks, *9th International Conference on Electronics, Circuits and Systems, 2002*, Vol. 1, pp. 291–294 vol.1.

Woo, A. and Culler, D. (2003). Evaluation of efficient link reliability estimators for low-power wireless networks, *Technical Report UCB/CSD-03-1270*, EECS Department, University of California, Berkeley.
URL: *http://www.eecs.berkeley.edu/Pubs/TechRpts/2003/6239.html*

Wu, K., Harms, J. et al. (2001). QoS support in mobile ad hoc networks, *Crossing Boundaries-the GSA Journal of University of Alberta* **1**(1): 92–106.

Wu, X., Chen, G. and Das, S. (2006). On the energy hole problem of nonuniform node distribution in wireless sensor networks, *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2006*, pp. 180–187.

Wu, X., Chen, G. and Das, S. (2008). Avoiding energy holes in wireless sensor networks with nonuniform node distribution, *IEEE Transactions on Parallel and Distributed Systems* **19**(5): 710–720.

Xiao Hui, L., Seung Ho, H. and Kang Ling, F. (2011). Location-based self-adaptive routing algorithm for wireless sensor networks in home automation, *EURASIP Journal on Embedded Systems* **2011**.

Xu, Y., Heidemann, J. and Estrin, D. (2001). Geography-informed energy conservation for ad hoc routing, *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ACM, pp. 70–84.

Yao, Y. and Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks, *SIGMOD Rec.* **31**(3): 9–18.
URL: *http://doi.acm.org/10.1145/601858.601861*

Ye, F., Chen, A., Lu, S. and Zhang, L. (2001). A scalable solution to minimum cost forwarding in large sensor networks, *Proceedings of the Tenth International Conference on Computer Communications and Networks, 2001*, IEEE, pp. 304–309.

Ye, F., Luo, H., Cheng, J., Lu, S. and Zhang, L. (2002). A two-tier data dissemination model for large-scale wireless sensor networks, *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, MobiCom '02, ACM, New York, NY, USA, pp. 148–159.
URL: *http://doi.acm.org/10.1145/570645.570664*

Ye, Z., Krishnamurthy, S. and Tripathi, S. (2003). A framework for reliable routing in mobile ad hoc networks, *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, Vol. 1, pp. 270–280 vol.1.

Yu, H.-C. and Tseng, S.-M. (2007). A wireless based sensor for patient monitoring system with remote diagnostic, *Third International Conference on Networking and Services, 2007. ICNS.*, pp. 92–92.

Yu, Y., Govindan, R. and Estrin, D. (2001). Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, *Technical report*, University of California, Los Angeles.

Zadeh, L. (1965). Fuzzy sets, *Information and Control* **8**(3): 338 – 353.
URL: *http://www.sciencedirect.com/science/article/pii/S001999586590241X*

Zadeh, L. (1975). Fuzzy logic and approximate reasoning, *Synthese* **30**(3-4): 407–428.
URL: *http://dx.doi.org/10.1007/BF00485052*

Zadeh, L. A. (1988). Fuzzy logic, *Computer* **21**(4): 83–93.

Zeng, K., Ren, K., Lou, W. and Moran, P. J. (2009). Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply, *Wirel. Netw., Springer-Verlag New York* **15**(1): 39–51.
URL: *http://dx.doi.org/10.1007/s11276-007-0022-0*

Zhang, G., Sivanand, K., Chin, F. and Chung, K. C. (2008). Quality of service and coverage optimization in wireless sensor networks, *IEEE Vehicular Technology Conference, 2008. VTC Spring 2008*, IEEE, pp. 2824–2828.

Zhang, Y. and Fromherz, M. (2004). Message-initiated constraint-based routing for wireless ad-hoc sensor networks, *First IEEE Consumer Communications and Networking Conference, 2004. CCNC 2004.*, IEEE, pp. 648–650.

Zhang, Y., He, C. and Jiang, L. (2008). Modeling the s-mac protocol in single-hop wireless sensor networks, *IEEE International Conference on Communications, 2008. ICC '08.*, pp. 317–321.

Zhao, J., Govindan, R. and Estrin, D. (2002). Sensor network tomography: monitoring wireless sensor networks, *ACM SIGCOMM Computer Communication Review* **32**(1): 64–64.

Zhao, S. and Raychaudhuri, D. (2009). Scalability and performance evaluation of hierarchical hybrid wireless networks, *IEEE/ACM Transactions on Networking* **17**(5): 1536–1549.

Zhaohua, L., Jianfeng, L. and Guiquan, J. (2010). Survey of routing protocols in wireless sensor networks based on applications, *2nd International Conference on Industrial and Information Systems (IIS), 2010*, Vol. 2, pp. 381–385.

Zimmermann, H. (2001). *Fuzzy Set Theory and Its Applications*, Springer Netherlands.
URL: *http://books.google.com.au/books?id=Uqtwf6bcUxMC*

Appendix A

Network Definition in Custom Simulator Source Code

Below is the content of the modules that represents a main framework for the custom discrete event simulation environment. The modules are included in a main class of (Node.java). This class provides the following configurations.

1. **The attribute of a node**

```
    public Nodes() {
         changeTimer = false;
   //currentEventQueue =new ArrayBlockingQueue<Event>(30);
         //buffer
         packetLimit = 2;
         hopCount = Integer.MAX_VALUE;
         fit = new ForwardInformationTable();
         doBroadcastOnce = true;
         macLayer = new MACLayer(this);
         phyLayer = new PhysicalLayer(this);
         battery = new Battery();
         rand = new Random(System.currentTimeMillis() % 6);
         errorProbability = 3;
         waitACK = new ConcurrentHashMap<Event, Long>();
   droppedPacket = new ConcurrentHashMap<Event, Long>();
         //currentEvent = Event.EventType.IDLE;
       }
```

2. **How to get the node's coordinate and the radius power**

```
     public void setXPos(int x) {
          xPos = x;
       }
       public int getXPos() {
           return xPos;
       }
       public void setYPos(int y) {
           yPos = y;
```

```
    }
    public int getYPos() {
        return yPos;
    }

    /**
     * Setting power transmission, temporary in db, next in mW
     *
     * @param power in db
     */
    public void setTransmitPower(int power) {
        transmitPower = power;
    }
    int getTransmitPower() {
        return transmitPower;
    }
    public void setRadiusPower(double _radiusPow) {
        if(Double.compare(_radiusPow, 100.0) >= 0)
            radiusPower = 100.0d;
        else
            radiusPower = _radiusPow;
    }
    public double getRadiusPower() {
        return radiusPower;
    }
    public int getNodeId() {
        return nodeId;
```

3. **How to scan channel access with CSMA-CA MAC layer**

```
  private int scanChannelActivity(long timeToSend) {
        int jmlBackOff = 0;
         MAC_CSMA_CA_WithoutSleep mac = null;
         StringBuilder out = new StringBuilder();
         if (macLayer instanceof MAC_CSMA_CA_WithoutSleep) {
             mac = (MAC_CSMA_CA_WithoutSleep) macLayer;
             mac.performNetworkInit();
             if (phyLayer.getChannelNumber() == 0) {
                 phyLayer.setChannelNumber(mac.pickChannel());
             }
//System.out.println("Node-" + nodeId + " Done waiting - BackOff Time");
//System.out.println("Node-" + nodeId + " Perform Network Init (MAC act)\
boolean statusChannel = timeSimulator.scanChannel(nodeId, phyLayer.\
getChannelNumber(), timeToSend);
//boolean statusChannel = randomizeNodeChannelAccess();
             if (statusChannel) {
                 //success reset
                 okToSend = macLayer.updateScanChannel();
                 out.append("Node-").append(nodeId).append\
               (" ::: channel ").append(phyLayer.getChannelNumber()).\
               append(" free to use");
```

```
            } else {
                //increment
                out.append("Node-").append(nodeId).append(" ::: channel").
                append(phyLayer.getChannelNumber()).append\
                (" in use, re-scan channel");
                if (!macLayer.retryScanChannel()) {
                    macLayer.reset();
                    return -1;
                }
            }
            try {
                //System.out.print("Send request scan channel ");
                drainBattery(EnergyConstant.ENERGY_SCAN_CHANNEL, 0);
            } catch (Exception ex) {
                //Logger.getLogger(Nodes.class.getName()).\
                log(Level.SEVERE, null, ex);
            }
            //if (!isBaseStation) {
            //battery.reduceEnergy(EnergyTimeConstant.\
            E_SCANNING_CHANNEL(transmitPower));
            //}
            if (statusChannel) {
                jmlBackOff += mac.getBackOffPeriod();
                addMacWaitTimeToSend = 0;
            } else {
                //retry disini - ambil channel lain
                phyLayer.setChannelNumber(mac.pickChannel());
                jmlBackOff += scanChannelActivity(timeToSend + \
                (mac.getBackOffPeriod() * MAC_CSMA_CA_WithoutSleep.\
                backoff_period_length));
                addMacWaitTimeToSend = (mac.getBackOffPeriod() * \
                MAC_CSMA_CA_WithoutSleep.backoff_period_length);
            }
        }
        addMacWaitTimeToSend += (mac.getBackOffPeriod() * \
        MAC_CSMA_CA_WithoutSleep.backoff_period_length);
        System.out.println(out.toString() + " ,no of backoff : " \
        + jmlBackOff);
        return jmlBackOff;
    }
```

4. **How to update the node's attributes (energy, coordinates, throughput, delivery ratio, delay)**

```
  public int getBaseXPos() {
        return baseXPos;
    }

    public int getTotalPacketDrop() {
        return totalPacketDrop;
    }
```

```
public float getPacketRatio() {
    if (totalDataPacketReceived != 0) {
        packetDeliveryRatio = (float) totalDataPacketSent / \
        TotalDataPacketReceived;
    }
    return packetDeliveryRatio;
}

public void setPacketRatio(float ratio) {
    packetDeliveryRatio = ratio;
}

public float getThroughput() {
    if (throughPut == 0) {
        return numOfBitSent;
    } else {
        return throughPut;
    }
}

public double getBatteryResidu() {
    return battery.getBatteryResidu();
}

public void setBatteryResidu(double energy) {
    battery.setEnergyResidu(energy);
}

public int getNumOfAccessFail() {
    return accessChannelFail;
}

public void setNumOfAccessFail(int total) {
    accessChannelFail = total;
}

public void setThroughput(float tput) {
    throughPut = tput;
}

void setRoutingAlgorithm(RoutingAlgorithm algorithm) {
    fit.setAlgoritmType(algorithm);
    EventComparator comp = new EventComparator();
    comp.setRoutingAlgorithm(algorithm);
    comp.setLimitTimeOut(4);
    incomingQueue = new PriorityBlockingQueue<Event>(size_queue, \
    new EventComparator());
  outgoingQueue = new PriorityBlockingQueue<Event>(size_queue, comp);
```

```
            if (algorithm instanceof route_algorithm\ || algorithm instanceof
    {
                hNode = new ConcurrentHashMap<Integer, Double>();
            }
        }

        public ForwardInformationTable getFITTable() {
            return fit;
        }

        private long processHelloMessage(Event e, long timeToQueue) \
        { //throws Exception {
          long helloDelay = timeToQueue - e.getTimeStart();
          fit.updateFromHello(e.getSrcNodeID(), helloDelay, nodeId);
          if (fit.getAlgorithm() instanceof Tegar3_1Algorithm) {
            if (e.getData(0) != null && e.getData(0) instanceof Float) {
              fit.updateNodeFuzzy(e.getSrcNodeID(), (Float) e.getData(0));
                }
            }
            drainBattery(EnergyConstant.E_HELLO_UPDATE_NEIGHBOUR\
            (fit.countNeightbour()), timeToQueue);
            //prepare reply
            long timeToSent = TimeConstant.T_HELLO_UPDATE_TABLE\
            (fit.countNeightbour());
            timeToSent += TimeConstant.T_HELLO_TO_SEND();
            return timeToSent;
        }

       private boolean drainBattery(double energyConsumed, long timeExecuted)
            boolean result = true;
            if (!isBaseStation) {
                try {
                    battery.reduceEnergy(energyConsumed);
                } catch (Exception ex) {
                    this.expired = true;
                    //Logger.getLogger(Nodes.class.getName()).log\
                    (Level.SEVERE, null, ex);
                    //info ke simulator
                    timeSimulator.notifyNodeOff(nodeId, timeExecuted);
                    //info ke semua tetangga
                    sendNotifyHello();
                    result = false;
//                  throw new Exception(ex.getMessage());
                }
            }
            return result;
        }

        private void updateThroughput() {
            if (numOfBitSent != 0) {
                if (!switched) {
```

```
                    aSecond = numOfBitSent;
                    switched = true;
                } else {
                    bSecond = numOfBitSent;
                    switched = false;
                }
                throughPut = (aSecond + bSecond) /
                2;//Math.round((aSecond + bSecond)/2);
            }
            //reset
            numOfBitSent = 0;
        }
```

5. **The data type and the process to send out data**

```
    private void countDataTypeSent(int type) {
            switch (type) {
                case 0:
                    totalHighestQoS++;
                    break;
                case 1:
                    totalDelaySensitiveData++;
                    break;
                case 2:
                    totalReliableData++;
                    break;
                case 3:
                    totalNormalData++;
                    break;
            }
        }

        void setEnergyThreshold(int limit) {
            battery.setLimit(limit);
        }

        private void sendNotifyHello() {
            if (!sendNotifyProcess) {
                sendNotifyProcess = true;
                Event hello = new Event();
                hello.setType(EventType.HELLO_MESSAGE_BROADCAST);
                hello.addData("NODE-REACH-ENERGY-LIMIT");
                long timeToSend = timer + TimeConstant.T_HELLO_TO_SEND();
                int jmlBackOff = scanChannelActivity(timeToSend);
                hello.setChannelNumber(phyLayer.getChannelNumber());
                long timeScanChannel = TimeConstant.TIME_SCAN_CHANNEL;
                if (jmlBackOff != -1) {
//                      try {
                    drainBattery(EnergyConstant.E_HELLO_TO_SEND(jmlBackOff, \
                    getRadiusPower()), timeToSend);
//                      } catch (Exception ex) {
//                          //Logger.getLogger(Nodes.class.getName()).log\
```

```
//                          (Level.SEVERE, null, ex);
//                      }
                    timeSimulator.triggerEvent(hello, timeToSend + \
                timeScanChannel, TimeConstant.T_BACKOFF_TIME(jmlBackOff)
                            + TimeConstant.TIME_HELLO_RECEIVE + \
                  TimeConstant.TIME_HELLO_PROPAGATION, nodeId, 0);
                    totalPacketControlSent++;
                }
                expired = true;
            }
        }
```

6. **CSMA-CA MAC Implementation**

```
package org.jwsn.timemodel.core.layer;
import java.util.Random;
import org.jwsn.timemodel.core.Nodes;

 */
public class MAC_CSMA_CA_WithoutSleep extends MACLayer {
    static final int maxMAC_CSMA_Backoffs = 4;
    static final int maxMAC_BE = 5;
    int NB;
    int BE;
    int CW;//contention window
    public static long backoff_period_length = 320L; //(320 microsecond)
    int jmlBackOff;

    public MAC_CSMA_CA_WithoutSleep(Nodes aThis){
        super();
        CW = 2;
        NB = 0;
        BE = 3;
        main = aThis;
        //rand = new Random(main.getXPos()+main.getYPos());
        rand = new Random(System.currentTimeMillis()%main.getNodeId());
    }

    @Override
    public void performNetworkInit(){
        waitperiod();
        //main.phyLayer.doCCA();
    }

    @Override
    public void reset(){
        CW = 2;
        NB = 0;
        BE = 3;
    }

    @Override
```

```
    public boolean retryScanChannel(){
        boolean result = true;
        NB ++;
        BE ++;
        if(BE > maxMAC_BE) BE = maxMAC_BE;
        if(NB > maxMAC_CSMA_Backoffs) result = false;
        return result;
    }

    @Override
    public boolean updateScanChannel() {
        CW--;
        if(CW==0) {
            reset();
            return true;
        }else {
            return false;
        }
    }

    public int getBackOffPeriod() {
        return jmlBackOff;
    }

    private void waitperiod() {
        //System.out.println("Waiting " + (long)((Math.pow(2,BE) - 1) *\
        backoff_period_length));
        jmlBackOff = rand.nextInt((int)Math.pow(2,BE) - 1);
    }
}
```

7. **The implementation of the time constant in the custom simulator**

```
public class TimeConstant {
    //all time unit in mikro second
    public static long TIME_RECEIVE_SINK = 1067L;
    public static long TIME_CAPSULATE_PHY = 294L;
    public static long TIME_PHY_TO_MAC = 25L;
    public static long TIME_CAPSULATE_MAC = 510L;
    public static long TIME_CHECK_HOP_VALUE = 100L;
    public static long TIME_UPDATE_HOP = 100L;
    public static long TIME_PROPAGATION = 875L;

    public static long TIME_CAPSULATE_APP = 174L;
    public static long TIME_APP_TO_NET = 25L;
    public static long TIME_CAPSULATE_NET = 342L;
    public static long TIME_NET_TO_MAC = 25L;

    public static long TIME_IDLE_RECEIVE = 192L;
    public static long TIME_CCA = 128L;
    public static long TIME_BOP = 320L;
    public static long TIME_SCAN_CHANNEL = 61632L;
```

```
public static long TIME_DATA_PROPAGATION = 3984L;//3984,375
public static long TIME_RECEIVE_DATA = 4204L;//4203,375

public static long TIME_COUNT_ATTR = 600L;
public static long TIME_HELLO_PROPAGATION = 1343L;
public static long TIME_HELLO_RECEIVE = 1563L;

//public static long TIME_WAIT_ACK = 12310L;
public static long T_WAIT_ACK(RoutingAlgorithm algorithm, \
int jmlNode){
//return ((T_SEND_DATA_CHANNEL() + T_DATA_TO_INCOMING() + (864*3))*2\
 + T_RUN_ALGORITHM(algorithm, jmlNode));
    return (2*(T_SEND_DATA_CHANNEL() + T_DATA_TO_INCOMING()));//
     + (864*3))*2 + T_RUN_ALGORITHM(algorithm, jmlNode));
    //return (16075 + 864*3);
};

public static long T_SINK_TO_INCOMING_Q(){
 return TIME_CAPSULATE_PHY + TIME_PHY_TO_MAC + TIME_CAPSULATE_MAC;
}

public static long T_SINK_PROCESS_TO_OUTGOING(boolean needHopUpdate){
    long time = TIME_CHECK_HOP_VALUE + TIME_CAPSULATE_MAC + \
    TIME_PHY_TO_MAC + TIME_CAPSULATE_PHY;
    if(needHopUpdate)
        time += TIME_UPDATE_HOP;
    return time;
}

public static long T_ACK_PROCESS_TO_OUTGOING(){
long time = TIME_CAPSULATE_MAC + TIME_PHY_TO_MAC + TIME_CAPSULATE_PHY;
    return time;
}
public static long T_BACKOFF_TIME(int jmlBackOff){
  if(jmlBackOff == 0) jmlBackOff = 1;
 return 2*(TIME_IDLE_RECEIVE + TIME_CCA) + (jmlBackOff * TIME_BOP);
}

public static long T_SEND_DATA_TO_OUTGOING() {
    long time = TIME_CAPSULATE_APP + TIME_APP_TO_NET +
      TIME_CAPSULATE_NET + TIME_NET_TO_MAC +
      TIME_CAPSULATE_MAC + TIME_PHY_TO_MAC + TIME_CAPSULATE_PHY;
    return time;

}

public static long T_SEND_DATA_CHANNEL(){
    return TIME_DATA_PROPAGATION + TIME_RECEIVE_DATA;
}
```

```
        public static long T_SEND_DATA_RUN_ALGORITHM(RoutingAlgorithm route,\
        int jmlNode){
                if(route instanceof Route algorithm){
         time = Set_constant(variable) * jmlNode * 1;\\based on the algorithm
            }else
           return time;
        }

    public static long T_DATA_TO_INCOMING(){
            //kurang kalkukasi process Hello in incoming queue
    return TIME_CAPSULATE_PHY + TIME_PHY_TO_MAC + TIME_CAPSULATE_MAC +
    TIME_NET_TO_MAC + TIME_CAPSULATE_NET + TIME_APP_TO_NET +
    TIME_CAPSULATE_APP;
        }

    public static long T_HELLO_TO_SEND(){
    return TIME_COUNT_ATTR + TIME_CAPSULATE_MAC + TIME_CAPSULATE_PHY +
                    TIME_PHY_TO_MAC;
        }

    public static long T_HELLO_TO_INCOMING_Q() {
            return TIME_CAPSULATE_PHY + TIME_PHY_TO_MAC + TIME_CAPSULATE_MAC;
        }

    public static long T_HELLO_UPDATE_TABLE(int numOfNeighbour){
            return 150 * numOfNeighbour;
        }

    public static long T_RUN_ALGORITHM(RoutingAlgorithm route, int jmlNode){
            long time = 0;
         if(route instanceof TegarAlgorithm){
       time = Set_constant(variable) * jmlNode * 1;\\based on the algorithm;
            }else
        {
            return time;
        }

    }
```

8. **Updating Forwarding Information Table of a node**

```
     public class ForwardInformationTable {
        int capacity;
        ***********************************************
        public ForwardInformationTable(){
            capacity = 10;
            rand = new Random(capacity);
            nodeArray = new ConcurrentHashMap<Integer, Nodes>();
            weightArray =  new HashMap<Integer, Float>();
            timerArray =  new HashMap<Integer, Long>();
            countArray =  new HashMap<Integer, Integer>();
        }
```

```
    public void updateStatusByTime(long interval)
    public void setCapacity(int cap){
        capacity = cap;
    }

    public Nodes getNeighbourNode(int id){
        return nodeArray.get(id);
    }

    void update(int nodeId, int hopcount, int xPos, int yPos, \
    double radius, byte[] macAddr, double hopDelay) {
        Nodes get = nodeArray.get(nodeId);
        if(get==null){
            Nodes add = new Nodes();
            add.setNodeId(nodeId);
            add.setXPos(xPos);
            add.setYPos(yPos);
            add.setRadius(radius);
            add.setHopCount(hopcount);
            if(algorithm instanceof RouteAlgorithm) {
           .updateHopDelayHistory(hopDelay);
                add.setHopDelay(hopDelay);
            } else
                add.setHopDelay(hopDelay);
            add.setAddress(macAddr);
            nodeArray.put(nodeId, add);
        }else {
            //update
            //if(hopcount < get.getHopCount()){
                get.setXPos(xPos);
                get.setYPos(yPos);
                get.setHopCount(yPos);
                get.setRadius(radius);
                get.setHopCount(hopcount);
                get.setAddress(macAddr);
                get.setHopDelay((hopDelay + get.getHopDelay())/2);
                nodeArray.replace(nodeId, get);
            //}
        }
        //update data algorithm
    }
************************************************


    void updateFromHello(int nodeId, long hopDelay, int selfNodeId) {
        Nodes source = TimeSimulator.getInstance().getNode(nodeId);
        Nodes neighbour = nodeArray.get(nodeId);
        if(neighbour!=null) {
            neighbour.setHopCount(source.getHopCount());
            neighbour.setBatteryResidu(source.getBatteryResidu());
```

```
                neighbour.setThroughput(source.getThroughput());//throughput
                neighbour.setPacketRatio(source.getPacketRatio());//ratio
                if(algorithm instanceof TReNsAlgorithm) {
                ((TegarAlgorithm) algorithm).updateHopDelayHistory(hopDelay);
               neighbour.setHopDelay((neighbour.getHopDelay() + hopDelay)/2);
                //hopdelay
                } else
               neighbour.setHopDelay((neighbour.getHopDelay() + hopDelay)/2);
                    //hopdelay
                neighbour.setNodeOff(false);
                neighbour.setNodeExpired(source.getNodeExpired());
                //node off
                //buffer full
                neighbour.setNumOfAccessFail(source.getNumOfAccessFail());
                neighbour.setBufferFullCount(source.getBufferFullCount());
                nodeArray.replace(nodeId, neighbour);
            } else {
                if(algorithm instanceof TegarAlgorithm || \
                if algorithm instanceof TReNs
                algorithm instanceof RULeARR
                algorithm instanceof SASARR
              }

    float weight;
    if(algorithm instanceof RouteAlgorithm){
    //calculate weight
    {
    weight = TegarAlgorithm.calculateWeight(TimeSimulator.getInstance().\
      getNode(nodeId), neighbourNode);
      weightArray.put(neighbourNodeId, weight);
                }
            }

        }

        void setAlgoritmType(RoutingAlgorithm route) {
            this.algorithm = route;
        }

        RoutingAlgorithm getAlgorithm() {
            return algorithm;
        }

        void setDataClassType(int classType) {
            algorithm.setDataType(classType);
        }


        }

    }
```

Appendix B

Routing Algorithm Source Code

The following provides the source codes of the proposed routing algorithms which consist of TeGaR, TReNs, RuLeARR and SASARR.

1. **TeGaR Algorithm**

```
***************************************
public class Tegar2Algorithm extends RoutingAlgorithm {
    public static float calculateWeight(Nodes i, Nodes neighbourNode, \
    int tipeDatax) {
        float a = 0, b = 0, c = 0, d = 0, e = 0;

       switch(tipeDatax){
            case 0:
                //highest//Delay
                a = 0.2f;
                b = 0.3f;
                c = 0.3f;
                d = 0.1f;
                e = 0.1f;
                break;
            case 1:
                //delay//Highest
                a = 0.23f;
                b = 0.1f;
                c = 0.4f;
                d = 0.17f;
                e = 0.1f;
                break;
            case 2:
                a = 0.25f;
                b = 0.4f;
                c = 0.1f;
                d = 0.1f;
                e = 0.15f;
                break;
            case 3:
                //no weighting all equal
                a = 0.4f;
```

```
                b = 0.15f;
                c = 0.15f;
                d = 0.15f;
                e = 0.15f;
                break;
        }

        TimeSimulator timer = TimeSimulator.getInstance();
        Nodes obj = timer.getNode(neighbourNode.getNodeId());
        float result = 0;

        float Es  = (float) ((obj.getBatteryResidu() - 0) / \
        (obj.getBatteryInitialEnergy() - 0));
        //float Ts  = obj.getThroughput()/1020 * timer.getCBR()/1000000;\
        //obj.getThroughput() * timer.getCBR()/1000;
        float PDRs = obj.getPacketRatio();
        float Ds  = (float) (obj.getRadius() / i.getRadius());
//          int OFFs = neighbourNode.getNumOffNodeOff()/TimeSimulator.\

         getInstance().getNumOfHello();
        //int FAILs = neighbourNode.getNumOfAccessFail()/TimeSimulator.\
        getInstance().getNumOfHello();
        //int FULLs = neighbourNode.getBufferFullCount()/TimeSimulator.\
        getInstance().getNumOfHello();
        int dataSentFromSource = i.getTotalDataSent();
        if(dataSentFromSource==0)
            dataSentFromSource = 1;
        //int OFFs = neighbourNode.getNumOffNodeOff()/dataSentFromSource;
        int FAILs = neighbourNode.getNumOfAccessFail()/dataSentFromSource;
        int FULLs = neighbourNode.getBufferFullCount()/dataSentFromSource;

        //if(Ts == 0)
        //      Ts = 0.9f;
        if(PDRs == 0)
            PDRs = 0.9f;

        float distance;
        //if(Ds > 1)
        //      distance = (Ds -1);
        //else
            distance = (1-Ds);
        //(E_s) * (PDR_s) * (1 - D_s) * (1 - real channel fail_s) * \

( 1 - real bufferexceeds threshold)
        result = (a*Es) +(b*PDRs)+ (c*distance) + (d*(1-FAILs)) + \

(e*(1-FULLs));
        return result;
    }

    private int tipeData;
```

```
    @Override
  public int getNextNode(Collection<Nodes> it, Nodes self, int nodeIdSender,
        System.out.println("Node-" + self.getNodeId() + " \

         Running Tegar2 Algorithm . . . . . . !! "
               + " num of neighbour " + it.size());
      HashMap<Integer, Float> array = self.getFITTable().getWeightArray();
        //int[] key = new int[array.size()];
        float[] valTemp = new float[array.size()];
        //System.out.println("Valtemp length" + valTemp.length);
        float[] valTempBackup = new float[valTemp.length ];
        float[] valFiltered, valFilteredBackup;
        int i =0, foundNode = 0, foundNodeBackup = 0;
        Nodes sender = TimeSimulator.getInstance().getNode(nodeIdSource);
        //for (Iterator<Integer> keyArray = array.keySet().iterator(); \
          keyArray.hasNext();) {
        for (Iterator<Nodes> it1 = it.iterator(); it1.hasNext();) {

            //int id = keyArray.next();
            //key[i] = id;
            Nodes nextNeighbour = it1.next();
            int id = nextNeighbour.getNodeId();
            //Nodes next = TimeSimulator.getInstance().getNode(id);
            Nodes next = TimeSimulator.getInstance().getNode(id);
            if(!nextNeighbour.getNodeExpired()){
            if(nextNeighbour.getNodeId()!= nodeIdSender && nextNeighbour.\
               getNodeId()!= nodeIdSource){
                    if(self.getHopCount() > next.getHopCount()){
                        //valTemp[foundNode] = array.get(id);
            valTemp[foundNode] = calculateWeight(sender, next, tipeData);
           self.getFITTable().replaceWeightArray(id, valTemp[foundNode]);
               System.out.println("[ --- Node-" + id + " Hopcount : "
               + next.getHopCount() + " Radius " + next.getRadius() +\
               " Weight " + valTemp[foundNode] + " --- ]");
                        foundNode++;
                    }else if(self.getHopCount() == next.getHopCount()){
                    //if node with lower hopcount unavailable
           valTempBackup[foundNodeBackup] = calculateWeight\
           (sender, next, tipeData);
                 self.getFITTable().replaceWeightArray(id, valTempBackup\
                 [foundNodeBackup]);
                 System.out.println("[ --- Node-" + id + " Hopcount : "
                         \
                  + next.getHopCount() + " Radius " + next.getRadius() +\
              " Weight(B) " + valTempBackup[foundNodeBackup] + " --- ]");
                        foundNodeBackup++;
                    }
                }
            } else {
                i++;
```

```
            }
            //i++;
        }

        valFiltered = new float[foundNode];
        valFilteredBackup = new float[foundNodeBackup];
        System.arraycopy(valTemp, 0, valFiltered, 0, foundNode);
        System.arraycopy(valTempBackup, 0, valFilteredBackup, 0, \

        foundNodeBackup);
        Arrays.sort(valFiltered);
        Arrays.sort(valFilteredBackup);
        float selectedVal = 0;
        if(foundNode > 1){
        System.out.print("Length sufficient " + valFiltered.length +\
         " tipe " + tipeData);
            selectedVal = chooseNextNode(valFiltered);
        } else {

            if(valFiltered.length==0){
                System.out.print("[No Matching Node , Num of Node expired
                System.out.println(" Try backup route \

                  (node with hopcount equals) ]");
                if(foundNodeBackup > 1){
                    selectedVal = chooseNextNode(valFilteredBackup);
                } else {
                    if(valFilteredBackup.length==0)
                        return 0;
                    else
                        selectedVal = valFilteredBackup[0];

                }
            }else {
    System.out.println("Length limited " + valFiltered.length +\
     " value " + valFiltered[0]);
                selectedVal = valFiltered[0];
            }
        }
        if(Float.compare(selectedVal, 0.0f)== 0 && valFiltered.length > 1
            selectedVal = valFiltered[1];
        System.out.println(" Selected Val " + selectedVal);
        array = self.getFITTable().getWeightArray();
        //if(Float.compare(selectedVal, 0.0f) != 0){
            for (Iterator<Integer> keyArray = array.keySet().iterator();\
             keyArray.hasNext();) {
                int id = keyArray.next();
                //key[i] = id;
                int comp = Float.compare(selectedVal, array.get(id));
                if(comp == 0){
                    System.out.println(" Found Node-" + id);
```

```
                    return id;
                }
            }
        return 0;
    }

    float chooseNextNode(float[] valFiltered){
        float selectedVal = 0;
        switch(tipeData){
                case 0://hQos
                    if(valFiltered.length > 1)
                        selectedVal = valFiltered[valFiltered.length -1];
                    break;
                case 1://delay'''->Reliable
                     if(valFiltered.length > 2)
                      selectedVal = valFiltered[valFiltered.length -2];
                    else if(valFiltered.length == 2)
                      selectedVal = valFiltered[valFiltered.length -1];
                    else
                      selectedVal = valFiltered[valFiltered.length -1];
                    break;
                case 2://reliable....->Delay
                    if(valFiltered.length >=4)
                       selectedVal = valFiltered[valFiltered.length -4];
                    else if(valFiltered.length ==3 )
                        selectedVal = valFiltered[valFiltered.length -3];
                    else if(valFiltered.length == 2)
                        selectedVal = valFiltered[valFiltered.length -2];
                    else
                        selectedVal = valFiltered[valFiltered.length -1];
                     break;
                case 3://normal
                     selectedVal = valFiltered[0];
                       break;
                     //  default:
                 //    selectedVal = valFiltered[valFiltered.length - 6];
                 //    break;
            }
        return selectedVal;
        }
}
```

2. **TReNs Algorithm**

```
  TimeSimulator timer = TimeSimulator.getInstance();
        int nodeNext = 0;
        //order based on distance ==> radius
        double[] rad = new double[it.size()];
```

```
        double[] radBackup = new double[it.size()];
        int[] idNode = new int[it.size()];
        int[] idNodeBackup = new int[it.size()];
        double[] energyResidu = new double[it.size()];
        double[] energyResiduBackup = new double[it.size()];
        int i = 0;
        int iBackup = 0;
        int reduce = 0;
        int reduceBackup = 0;
        for (Iterator<Nodes> it1 = it.iterator(); it1.hasNext();) {
            Nodes nd = it1.next();
            if (nd.getNodeId() == self.getBaseNodeId()) {
                return nd.getNodeId();
            }
          if (!nd.getNodeExpired() && nd.getNodeId() != nodeIdSender && \
                nd.getNodeId() != nodeIdSource
                    && nd.getHopCount() < self.getHopCount()) {
                rad[i] = nd.getRadius();
                idNode[i] = nd.getNodeId();
                energyResidu[i] = nd.getBatteryResidu();
                i++;
            } else {
                reduce++;
                if (nd.getHopCount() == self.getHopCount()) {
                    radBackup[iBackup] = nd.getRadius();
                    idNodeBackup[iBackup] = nd.getNodeId();
                    energyResiduBackup[iBackup] = nd.getBatteryResidu();
                    iBackup++;
                    reduceBackup++;
                }
            }

        }

//compute weight A = [1/3 * (E_s + Hopdelay_s + hardware fail_s)]
        int idA = 0, idB = 0;
        float weightA = 0.0f, weightB = 0.0f;
        double selectA = 0.0d, selectB = 0.0d;

        Nodes neighbour, obj;
        if (rad.length >= 2) {
            for (i = 0; i < rad.length - reduce; i++) {
                if (Double.compare(selectA, 0.0d) != 0) {
                    if (Double.compare(selectB, 0.0d) != 0) {
                        if (Double.compare(rad[i], selectB) < 0) {
                            if (Double.compare(rad[i], selectA) < 0) {
                                selectB = selectA;
                                idB = idA;
                                selectA = rad[i];
                                idA = idNode[i];
                            } else {
```

```
                                        selectB = rad[i];
                                        idB = idNode[i];
                                    }
                                }
                            } else {
                                if (Double.compare(rad[i], selectA) > 0) {
                                    selectB = rad[i];
                                    idB = idNode[i];
                                } else {
                                    selectB = selectA;
                                    idB = idA;
                                    selectA = rad[i];
                                    idA = idNode[i];
                                }
                            }
                        } else {
                            selectA = rad[i];
                            idA = idNode[i];
                        }
                    }
                } else {
                    if (rad.length == 1) {
                        return idNode[0];
                    } else {
                        return 0;
                    }
                }

                if (tipeData == 0 || tipeData == 1) {
                    float Es;
                    int FAILs;
                    int dataSentFromSource;
        if (idA != 0) {
        obj = timer.getNode(idA);
        Es = (float) ((obj.getBatteryResidu()) / \
         (obj.getBatteryInitialEnergy()));
        neighbour = self.getFITTable().getNeighbourNode(idA);
        dataSentFromSource = self.getTotalDataSent();
        if (dataSentFromSource == 0) {
        dataSentFromSource = 1;
                        }
        FAILs = neighbour.getNumOfAccessFail() / dataSentFromSource;
        weightA = (float) (Es * (1 - (getLast2HopDelay() / c)) * \
         (1 - FAILs));
        //weightA = (float) (Es * (1-(neighbour.getHopDelay()/c))\
         * (1-FAILs));// * (1-neighbour.getPacketQueueRatio()));
                    }

                    if (idB != 0) {
                        obj = timer.getNode(idB);
                        Es = (float) ((obj.getBatteryResidu()) / \
```

```
                    (obj.getBatteryInitialEnergy()));
                    neighbour = self.getFITTable().getNeighbourNode(idB);
                    dataSentFromSource = self.getTotalDataSent();
                    if (dataSentFromSource == 0) {
                        dataSentFromSource = 1;
                    }
    FAILs = neighbour.getNumOfAccessFail() / dataSentFromSource;
    //weightB = (float) (1 / 3 * (Es + neighbour.getHopDelay() + FAILs));
    //weightB = (float) (Es * (1-(neighbour.getHopDelay()/50000)) * (1-FAILs))
    weightB = (float) (Es * (1 - (getLast2HopDelay() / 50000)) * (1 - FAILs));
                }

                if (Float.compare(weightB, weightA) > 0) {
                    if (tipeData == 0) {
                        nodeNext = idB;
                    } else {
                        nodeNext = idA;
                    }
                } else {
                    if (tipeData == 0) {
                        nodeNext = idA;
                    } else {
                        if (idB != 0) {
                            nodeNext = idB;
                        } else {
                            nodeNext = idA;
                        }
                    }
                }

            } else {
    if (rad.length > 2) {
    //tipe data 2 - reliable dan 3
    selectA = 0.0d;
    selectB = 0.0d;
    int idNodeA = 0, idNodeB = 0;
    for (i = 0; i < rad.length - reduce; i++) {
      if (idNode[i] != idA && idNode[i] != idB) {
        if (Double.compare(selectA, 0.0d) != 0) {
          if (Double.compare(selectB, 0.0d) != 0) {
           if (Double.compare(TimeSimulator.getInstance().\
            getNode(idNode[i]).getBatteryResidu(), selectB) < 0) {
             if (Double.compare(TimeSimulator.getInstance().\
               getNode(idNode[i]).getBatteryResidu(), selectA) < 0) {
                 selectB = selectA;
                 idNodeB = idNodeA;
                 selectA = TimeSimulator.getInstance().getNode(idNode[i]).\
                 getBatteryResidu();
                  idNodeA = idNode[i];
                                          }
                   else {
```

```
        selectB = TimeSimulator.getInstance().getNode(idNode[i]).\
         getBatteryResidu();
         idNodeB = idNode[i];
                                }
                            }
                        } else {
        if (Double.compare(TimeSimulator.getInstance().\
        getNode(idNode[i]).getBatteryResidu(), selectA) > 0) {
        selectB = TimeSimulator.getInstance().getNode(idNode[i]).\
        getBatteryResidu();
            idNodeB = idNode[i];
                            } else {
            selectB = selectA;
            idNodeB = idNodeA;
            selectA = TimeSimulator.getInstance().\
            getNode(idNode[i]).getBatteryResidu();
            idNodeA = idNode[i];
                            }
                        }
                    } else {
                selectA = TimeSimulator.getInstance().getNode\
                        (idNode[i]).getBatteryResidu();
                         idNodeA = idNode[i];
                    }
                }
            }

            if (tipeData == 2) {
                if (idNodeB != 0) {
                    nodeNext = idNodeB;
                } else if (idNodeA != 0) {
                    nodeNext = idNodeA;
                } else {
                    nodeNext = idB;
                }

            } else {
                if (idNodeA != 0) {
                    nodeNext = idNodeA;
                } else {
                    nodeNext = idB;
                }
            }
            //
            if (nodeNext == 0) {
                nodeNext = idA;
            }

        } else {
            int idNodeA = 0, idNodeB = 0;
            for (i = 0; i < radBackup.length - reduceBackup; i++) {
```

```
                    if (Double.compare(selectA, 0.0d) != 0)
                   }

                   {
                       if (idNodeA != 0) {
                           nodeNext = idNodeA;
                       } else {
                           nodeNext = idB;
                       }
                   }

               }
           }

           if (nodeNext == 0) {
               nodeNext = idNode[rad.length - 1];
           }
           return nodeNext;
       }
```

3. **RuLeARR Algorithm**

```
/*
 *
public class TegarPositionalNoFuzzyBSAdjust extends RoutingAlgorithm {
    private int tipeData;
    float minimumWeight = 0.0f, maxWeight = 0.0f;
    int idMinWeight = 0, idMaxWeight = 0;

    @Override
    public int getNextNode(Collection<Nodes> it, Nodes self,\
     int nodeIdSender, int nodeIdSource) {


        HashMap<Integer, Float> realtimePath = new HashMap();
        HashMap<Integer, Float> nonrealtimePath = new HashMap();
        HashMap<Integer, Float> backupRealtimePath = new HashMap();
        HashMap<Integer, Float> backupNonRealtimePath = new HashMap();
        double h_init = TimeSimulator.getInstance().getH_Init();
        //int num = checkInsideNodes(it, self, nodeIdSender, nodeIdSource);
        //reset value-value sebelumnya
        idMaxWeight = 0;
        idMinWeight = 0;
        maxWeight = 0;
        minimumWeight = 0;

        for (Iterator<Nodes> it1 = it.iterator(); it1.hasNext();) {
            Nodes nextNeighbour = TimeSimulator.getInstance().\
            getNode(it1.next().getNodeId());
         if(!nextNeighbour.getNodeExpired() && nextNeighbour.getNodeId()\
            != nodeIdSender
                   && nextNeighbour.getNodeId() != nodeIdSource){
                //jika node BS langsung pilih
```

```
                if(nextNeighbour.getNodeId() == self.getBaseNodeId())
                    return nextNeighbour.getNodeId();
                //ambil hanya yg lower hopcount
                if(nextNeighbour.getHopCount() < self.getHopCount()){
                    //hnode < hinit (inside) - realtime
                    if(Double.compare(nextNeighbour.getHNode(nodeIdSource)
                    h_init) <=0)
                     realtimePath.put(nextNeighbour.getNodeId(), \
          TegarAlgorithm.calculateWeight(self, nextNeighbour, tipeData));
                    //hnode > hinnit (outside) - normal paket
                    else
              nonrealtimePath.put(nextNeighbour.getNodeId(), \
          TegarAlgorithm.calculateWeight(self, nextNeighbour, tipeData));
                }else if(nextNeighbour.getHopCount() == self.getHopCount())
                    //jalur cadangan
          if(Double.compare(nextNeighbour.getHNode(nodeIdSource), h_init)
           backupRealtimePath.put(nextNeighbour.getNodeId(),
           TegarAlgorithm.calculateWeight(self, nextNeighbour, tipeData));
                    else
            backupNonRealtimePath.put(nextNeighbour.getNodeId(),\
           TegarAlgorithm.calculateWeight(self, nextNeighbour, tipeData));
                }
            }
        }

     System.out.println("Realtime path size \t" + realtimePath.size());
    System.out.println("Non Realtime path size\t " + nonrealtimePath.size
    System.out.println("Backup Realtime path size \t" + \
    backupRealtimePath.size());
    System.out.println("Backup Non Realtime path size\t " + \
    backupNonRealtimePath.size());

        int selectedNode = 0;
        switch(tipeData){
            case 0://realtime
            case 1://delaysensitive
            case 2://reliable
            case 3://normaldata
                if(realtimePath.size()>0){
                    if(realtimePath.size()==1)
                  selectedNode = realtimePath.keySet().iterator().next();
                    else
                        selectedNode = selectNodeOnRealTimePath\
                        (realtimePath, selectedNode);
                }else {
                    if(backupRealtimePath.size() > 1)
                        selectedNode = selectNodeOnRealTimePath\
                        (backupRealtimePath, selectedNode);
                    else if(backupRealtimePath.size() == 1)
                  selectedNode = backupRealtimePath.keySet().iterator().\
                        next();
```

```
                    else if(nonrealtimePath.size() > 1)
                        selectedNode = selectNodeOnRealTimePath\
                        (nonrealtimePath,\
                        selectedNode);
                    else if(nonrealtimePath.size() == 1)
                selectedNode = nonrealtimePath.keySet().iterator().next();
                }
                break;
        }
        return selectedNode;
    }
    private int selectNodeOnRealTimePath(HashMap<Integer, Float> \
    realtimePath, int selectedNode) {
        int secondNode = 0; float secondWeight = 0.0f;
        int thirdNode = 0; float thirdWeight = 0.0f;
for (Iterator<Integer> ids = realtimePath.keySet().iterator(); \

ids.hasNext();) {
            int idNode = ids.next();
            float idWeight = realtimePath.get(idNode);
            if(Float.compare(idWeight,maxWeight)>0){
                if(Float.compare(minimumWeight, 0.0f) == 0){
                    idMinWeight = idMaxWeight;
                    minimumWeight = maxWeight;
                }else if (Float.compare(maxWeight, minimumWeight) < 0){
                    idMinWeight = idMaxWeight;
                    minimumWeight = maxWeight;
                }else {
                    if(secondNode == 0){
                        secondNode = idMaxWeight;
                        secondWeight = maxWeight;
                    }else {
                        if(Float.compare(secondWeight, idWeight) > 0){
                            if(thirdNode == 0){
                                thirdNode = secondNode;
                                thirdWeight = secondWeight;
                            }else {
                             if(Float.compare(thirdWeight, idWeight) <0){
                                    thirdNode = idNode;
                                    thirdWeight = idWeight;
                                }//else ignore
                            }
                        }else {
                            thirdNode = secondNode;
                            thirdWeight = idWeight;
                            secondNode = idNode;
                            secondWeight = idWeight;
                        }

                    }
                }
```

```
                    maxWeight = idWeight;
                    idMaxWeight = idNode;
                }else {
                    if(Float.compare(minimumWeight, 0.0f) == 0){
                        idMinWeight = idNode;
                        minimumWeight = idWeight;
                    }else if(Float.compare(idWeight,minimumWeight)<0){
                        if(secondNode == 0){
                            secondNode = idMinWeight;
                            secondWeight = minimumWeight;
                        }
                        idMinWeight = idNode;
                        minimumWeight = idWeight;

                    }else {
                        if(secondNode == 0){
                            secondNode = idMaxWeight;
                            secondWeight = maxWeight;
                        }else {
                            if(Float.compare(secondWeight, idWeight) > 0){
                                if(thirdNode == 0){
                                    thirdNode = secondNode;
                                    thirdWeight = secondWeight;
                                }else {
                                 if(Float.compare(thirdWeight, idWeight) <0){
                                        thirdNode = idNode;
                                        thirdWeight = idWeight;
                                    }//else ignore
                                }
                            }else {
                                thirdNode = secondNode;
                                thirdWeight = idWeight;
                                secondNode = idNode;
                                secondWeight = idWeight;
                            }

                        }
                    }
                }
            }
            switch(tipeData){
                case 0:
                    selectedNode = idMaxWeight;
                    if(selectedNode == 0 || TimeSimulator.getInstance().\
                     getNode(selectedNode).getTotalPacketDrop() >=2)
                        selectedNode = secondNode;
                    break;
                case 1:
                    if(realtimePath.size() > 3){
                        selectedNode = thirdNode;
                        if(selectedNode == 0 || TimeSimulator.getInstance().\
```

```
                        getNode(selectedNode).getTotalPacketDrop() >=2)
                            selectedNode = secondNode;
                    }else if(realtimePath.size() > 2)
                        selectedNode = secondNode;
                    else
                        selectedNode = idMinWeight;
                    break;
                case 2:

                case 3:
                    selectedNode = idMinWeight;
                    if(selectedNode!=0 && TimeSimulator.getInstance().\
                    getNode(selectedNode).getTotalPacketDrop() >=2)
                        selectedNode = thirdNode;
                    break;
            }
        if(selectedNode == 0 && idMaxWeight!=0) selectedNode = idMaxWeight;
            else if(thirdNode != 0) selectedNode = thirdNode;
            else if(secondNode !=0) selectedNode = secondNode;
            else if(idMinWeight!=0) selectedNode = idMinWeight;

            return selectedNode;
    }

}
```

4. **SASARR Algorithm**
   Like RULeARR, SASARR uses routing region approach but it implements Fuzzy logic for the adaptation of the routing region. Thus, in here, we attach the (a) fuzzy logic code and (b) the adjustment of BS to adapt the region based on the QoS values' updates rather than the routing region source code.

```
** * ********************************************************
Fuzzy Logic Implementation
***********************************************************

package org.fsi.enginegeneric;

import java.util.Hashtable;
import org.fsi.core.SituationInference;
import org.fsi.core.Situation;
import org.fsi.core.LinguisticVariable;
import org.fsi.core.FuzzyRule;
import org.fsi.core.ContextAttribute;
import org.fsi.datasource.DSGenerator;
import org.fsi.datasource.DSGeneratorFactory;
import java.util.Vector;
import org.fsi.datasource.RecordEvent;
***
***
***
```

```
public class MainFuzzy {
    SituationInference inferer;
    Thread t2;

    public MainFuzzy() {

    }

    public void startInferer(){
        t2 = new Thread (inferer);
        t2.start();
    }

    public void stopInferer(){
        t2.stop();
    }

    public void initParameters() {
//step1: define context attributes, their names, values 'll be set by data
        Vector attrs=new Vector();
        attrs.addElement(new ContextAttribute("delay",0));
        attrs.addElement(new ContextAttribute("energy",0));
        attrs.addElement(new ContextAttribute("DR",0));

//step2: define situation names and their importance regarding criticality
        Vector situations=new Vector();
  //      Situation s1=new Situation("verylow",1);
        Situation s1=new Situation("low",1);
     //   Situation s3=new Situation("upperlow", 1);
//        Situation s4=new Situation("lowermed", 1);
        Situation s2=new Situation("medium", 1);
    //    Situation s5=new Situation("uppermed", 1);
        Situation s3=new Situation("high", 1);
     //   Situation s5=new Situation("veryhigh", 1);
//        Situation s4=new Situation("hyper_threat",0.5);
//        Situation s5=new Situation("hypertension",0.9);
//        situations.addElement(s1);
        situations.addElement(s1);
        situations.addElement(s2);
        situations.addElement(s3);
    //    situations.addElement(s4);
     //   situations.addElement(s5);
   //     situations.addElement(s6);
   //     situations.addElement(s7);

        inferer = SituationInference.getInstance(attrs, situations);

//step 3: define linguistic variables based on context attribute names
//and break them into terms
       LinguisticVariable var1=new LinguisticVariable("delay",0,400);
   //var1.addTerm("verylow",0, 0,120,125);
```

```
    //Membership function("low",0, 0,95,100);
          var1.addTerm("low",0,0,175,185);
     //     var1.addTerm("upperlow",135,140,160,165);
          var1.addTerm("medium",175,185,295,305);
        //("medium",95,100,295,300);
    //      var1.addTerm("uppermed",185,190,210,215);
    //("high",295,300,500,500);
          var1.addTerm("high",295,305,390,400);
     //     var1.addTerm("veryhigh",350,355,495,500);

          LinguisticVariable var2=new LinguisticVariable("energy",0,100);
   //       var2.addTerm("verylow",0, 0,22,25);
        //Membership function("low",0, 0,95,100);
          var2.addTerm("low",0,0,35,40);
     //     var2.addTerm("upperlow",28,31,42,45);
          var2.addTerm("medium",35,40,65,70);//("medium",95,100,295,300);
     //   var2.addTerm("uppermed",60,63,70,73);//("high",295,300,500,500);
          var2.addTerm("high",65,70,95,100);
     //     var2.addTerm("veryhigh",86,89,98,100);

          LinguisticVariable var3=new LinguisticVariable("DR",1,100);
          var3.addTerm("low",0,0,40,45);
     //     var3.addTerm("upperlow",40,42,58,60);
          var3.addTerm("medium",40,45,75,80);//("medium",95,100,295,300);
    //   var3.addTerm("uppermed",71,73,80,82);//("high",295,300,500,500);
          var3.addTerm("high",75,80,95,100);

  //step4: register linguistic variables with inferer
          inferer.registerVar(var1);
          inferer.registerVar(var2);
          inferer.registerVar(var3);


          //step5: define fuzzy rules
          Vector<FuzzyRule> fuzzyRules=new Vector();

          //ps. situations can be defined based on the case

          FuzzyRule rule=new FuzzyRule("normal");
          rule.addCondition(var1,"low","0.45");
          rule.addCondition(var2,"high","0.2");
          rule.addCondition(var3,"high","0.35");
          fuzzyRules.addElement(rule);

          FuzzyRule rule1=new FuzzyRule("lightCongested");
          rule1.addCondition(var1,"medium","0.48");
          rule1.addCondition(var2,"medium","0.15");
          rule1.addCondition(var3,"medium","0.37");
          fuzzyRules.addElement(rule1);

          FuzzyRule rule2=new FuzzyRule("Congested");
```

```
        rule2.addCondition(var1,"high","0.45");
        rule2.addCondition(var2,"low","0.2");
        rule2.addCondition(var3,"low","0.35");
        fuzzyRules.addElement(rule2);

        //step6: registerRules with inferer
        for(FuzzyRule fuzzyRule : fuzzyRules){
            inferer.registerRule(fuzzyRule);
        }

        //generate data
        DSGenerator dsGen;
        dsGen = DSGeneratorFactory.createRandomDS(attrs.size()) ;
        dsGen.addSubscriber(inferer);
        dsGen.setGeneratingRate (100);
        Thread t1 = new Thread (dsGen );
        Thread t2 = new Thread (inferer);
        t1.start();
        t1.yield();
        t2.start();
        t2.yield();

        try{
        Thread.sleep (10000);

        }catch(Exception e){}

        inferer.stop();
        dsGen.stop();

    }

    public Vector notifyUpdate(RecordEvent recordEvent) {
        inferer.update(recordEvent);
        return inferer.doInference();
    }

    public Hashtable getHighest() {
        return inferer.getHighest();
    }

}
```

****

BS's adjustment according to routing region adaptation

****

```
public void finish(){
        //mainFuzzy.stopInferer();
        recordEvent = null;
    }
```

```
public double adjustRegionHInit(double h_init , double avgDelay,\
double energy, double deliveryRatio){
        if(Double.compare(hInitPreset,0.0d) == 0)
            hInitPreset = h_init;
        double newHInit = hInitPreset;
        //Set_region can be adjusted
 //       double verylow = 0.4d * hInitPreset;
        double low = 0.5d * hInitPreset;
//        double upperlow = 0.7d * h_init;
//        double lowermed = 0.85d * h_init;
        double medium = 1.0d * hInitPreset;
//        double uppermed = 1.25d * h_init;
        double high = 2.0d * hInitPreset;
//        double veryhigh = 2.0d * h_init;

   rec = new DSRecord(avgDelay/1000, energy*100, deliveryRatio*100);
        recordEvent.update(rec);
        Vector result = mainFuzzy.notifyUpdate(recordEvent);
        System.out.println(result.toString());
        //Hashtable highest = mainFuzzy.getHighest();
        double value = 0;
        double totalSumOfConfidence = 0;
        System.out.println("Fuzzy Result = " + result.size());
        //mengkalikan bobot hasil decision fuzzy dengan value h_init baik 
       for (Iterator<Hashtable> it = result.iterator(); it.hasNext();) {
            Hashtable maps = it.next();
       if(maps.containsKey("normal")){
      //1. hypolow
      value += low * Double.valueOf((String)maps.get("normal"));
       totalSumOfConfidence += Double.valueOf((String)maps.get("normal"));
        }else if(maps.containsKey("lightCongested")){
         //3. low
        value += medium * Double.valueOf((String)maps.get("lightCongested"
        totalSumOfConfidence += Double.valueOf((String)maps.get\
       ("lightCongested"));
        }else if(maps.containsKey("Congested")){
        //4. upperlow
        value += high * Double.valueOf((String)maps.get("Congested"));\
    totalSumOfConfidence += Double.valueOf((String)maps.get("Congested"));
            }
   {

public static void main(String args[]){
        BSFuzzyAdjust bsFuzz = new BSFuzzyAdjust();
        double h_init = 30;
        double avgDelay = 250;
        double energy = 30.3d;
        double deliveryRatio = 25.2d;
        System.out.println("Current h_init " + h_init);
        h_init = bsFuzz.adjustRegionHInit(h_init, avgDelay, energy, \
```

```
            deliveryRatio);
            System.out.println("After adjustment h_init " + h_init);
            bsFuzz.finish();
        }

        public void updateAverageDelay(int srcNode, long delay, int seqNum) {
            boolean exist = totalDelay.containsKey(srcNode);
            if(!exist){
                Vector dt = new Vector();
                dt.add(delay);
                dt.add(seqNum);
                dt.add(1);
                totalDelay.put(srcNode, dt);
            }else {
                Vector dt = totalDelay.get(srcNode);
                dt.set(0, ((Long)dt.get(0)).longValue() + delay);
                dt.set(1, seqNum);
                dt.set(2, ((Integer)dt.get(2)).intValue() + 1);
            }

        }

        public double adjustRegionHInit(double hInit, long timestamp) {
            double[] d = getAverageDelay();
            System.out.println("Adjusting H-INIT energy : " \
             + getAverageEnergy());
            System.out.println("delay " + d[0]);
            System.out.println("ratio " + d[1]);

            //adjust region when delay dan DR not change
            boolean last3Data = DataLogger.getInstance().getLastEventHelloBS()
            boolean sameLast3DR = DataLogger.getInstance().check3LastDR();
            if(last3Data){
                double newHInit = hInit * 1.5;
                if(Double.compare(newHInit, 70.0) >= 0)
                    newHInit = 70.0;
                Nodes bs = TimeSimulator.getInstance().getNode\
                (TimeSimulator.getInstance().getBSNodeId());
                bs.setRadiusPower(bs.getRadiusPower() + 10.0);
                DataLogger.getInstance().addBSHelloEvent(timestamp,\
                getAverageEnergy(), d[0], d[1], hInit, newHInit);
                return newHInit;
            } else if(sameLast3DR){
                double newHInit = hInit;
                DataLogger.getInstance().addBSHelloEvent(timestamp, \
                getAverageEnergy(), d[0], d[1], hInit, newHInit);
                return newHInit;
            }else if((Double.compare(d[0], 0) == 0) || \
            (Double.compare(d[1], 0) == 0)) {
                DataLogger.getInstance().addBSHelloEvent(timestamp, \
               getAverageEnergy(), d[0], d[1], hInit, hInit);
```

```
                return hInit;
            } else {
                double newHInit = adjustRegionHInit(hInit, d[0], \
                getAverageEnergy(), d[1]);
                DataLogger.getInstance().addBSHelloEvent(timestamp, \
                getAverageEnergy(), d[0], d[1], hInit, newHInit);
                return newHInit;
            }
        }

        public double adjustRegionHInitNoFuzzy(double h_Init, long timestamp)
            double[] d = getAverageDelay();
            if(Double.compare(hInitPreset, 0.0d)==0)
                hInitPreset = h_Init;
            double newHInit = hInitPreset;

            System.out.println("Adjusting H-INIT energy : " + \
            getAverageEnergy());
            System.out.println("delay " + d[0]);
            System.out.println("ratio " + d[1]);
            double delay = 0;
                if(Double.compare(d[0], 175000) <=0){
                    delay = 1.0d;
                }else {
                    delay = 175000/d[0];
                }

            //
    boolean last3Data = DataLogger.getInstance().getLastEventHelloBS();
    boolean sameLast3DR = DataLogger.getInstance().check3LastDR();
    if(last3Data){
    newHInit = h_Init * 1.5;
    if(Double.compare(newHInit, 70.0) >= 0)
    newHInit = 70.0;
    Nodes bs = TimeSimulator.getInstance().getNode(TimeSimulator.getInstance().
    bs.setRadiusPower(bs.getRadiusPower() + 10.0);
    DataLogger.getInstance().addBSHelloEvent(timestamp, getAverageEnergy(), \
    delay, d[1], h_Init, newHInit);
                return newHInit;
            }else if(sameLast3DR){
                newHInit = h_Init;
              DataLogger.getInstance().addBSHelloEvent(timestamp, \
              getAverageEnergy(), delay, d[1], h_Init, newHInit);
                return newHInit;
            }else if((Double.compare(d[0], 0) == 0) || (Double.compare(d[1],\
             0) == 0)) {
                DataLogger.getInstance().addBSHelloEvent(timestamp, \
                getAverageEnergy(), delay, d[1], h_Init, newHInit);
                return hInitPreset;
            } else {
                double val = (delay + d[1] + getAverageEnergy())/3;
```

```
            System.out.println("Value : " + val);
            if(Double.compare(val, 0.4) <= 0){
                newHInit = 2.0*hInitPreset;
            }else if(Double.compare(val, 0.4) > 0 && Double.\
                compare(val, 0.8) <= 0) {
                newHInit = 1.0*hInitPreset;
            }else if(Double.compare(val, 0.8) > 0 ) {
                newHInit = 0.5*hInitPreset;
            }
            DataLogger.getInstance().addBSHelloEvent(timestamp, \
            getAverageEnergy(), delay, d[1], h_Init, newHInit);
            return newHInit;
        }
    }
```

Publication

Publications arising from this thesis include:

Rizal, M. N. Gondal, I., P. D. and Qiu, B . (2012), Priority Based Expansion of neighbourhood Size for Heterogeneous Traffic Routing in WSN. In *The 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, ACM.* Paphos, Cyprus Island.