Jimma University

School of Graduate Studies

School of Computing

Computer Networking Stream

Improving transport protocol for reliable data transfer in Wireless sensor network

A Thesis Submitted in Practical Fulfillment of the Requirement for the Degree of Master of Science in Computer Networking

By: Andualem W/giorgis

Advisor's:

1. Principal Advisor: Dr.Ing.Towfik Jemal(Phd)
2. Co-Advisor: Mr.Salahdin Seid (MSc)

May 2017

Jimma Ethiopia

Jimma University

School of Graduate Studies

School of Computing

Computer Networking Stream

Thesis title:

Improving transport protocol for reliable data transfer in Wireless sensor network

By: Andualem W/giorgis

APPROVED BY EXAMINING BOARD:

Name	Signature	Date
1. Dr.Ing.Towfik Jemal (Phd) (Principal Advisor)	_____	_____
2. Dr. Dereje Yohannes (Phd) (External Examiner)	_____	_____
3. Mr.Fissha Bayu (Phd Candidate) (Internal Examiner)	_____	_____
4. Mr. Salhadin Seid (MSc) (Co-Advisor)	_____	_____

May 2017

Jimma Ethiopia

Acknowledgment

First, I would like to say thank you to God who has given me the strength to endure all the difficulties and challenges of life.

Secondly, I would like to thank my principal advisor Dr.Ing.Towfik Jemal and Co-Adivisor Mr. Salahdin Seid for their endless patience and giving me insightful ideas and the constant motivation and support during my work.

I would also like to express my sincere gratitude to school of computing for giving chance to explore further on research areas.

Finally, I must thank school of computing staff members, Mesfin W/giorgis, Mebrat Bya, Zekarias Teferi, Gashaw Zemene, Brihanu Megersa, Ing.Ermias Girma and my friends who were being with me in those difficult situations. They are the greatest blessing in my life.

Table of Contents

List of Abbreviations (Or) Symbols

ACK……Acknowledgement

ADC……Analog - to - Digital converters

CCF……Congestion Control and Fairness

CODA……Congestion Detection and Avoidance

EACK……Explicit Acknowledgment

ESRT…… Event to sink Transport Protocol

FIFO…… Frist in first Out

GPS……..Global positioning system

IACK…… Implicit Acknowledgment

NACK……Negative Acknowledgements

PCCP…… Priority-based Congestion Control Protocol

RBC……Reliable Bursty Converge cast

RFID……Radio Frequency identification

SCTP…… Sensor Transmission Control Protocol
ART……Asymmetric and Reliable Transport

TCP…… Transport control protocol

UDP .....User data gram protocol

WSN……Wireless sensor network

List of Figures

List of Tables

Abstract

Wireless sensor network is an infrastructure contained of sensing (measuring), computing, and communication elements having capability of self-organizing, lightweight sensor nodes that are used to cooperatively monitor physical or environmental conditions. This network become an essential technology that applied in many applications areas such as in military, agricultural, health care, and many other areas. This thesis addresses the problem of reliable data transfer in wireless sensor network. In such networks, the principal challenge in the design of transport protocol is that the sensor's node deployed in harsh environment, due to harsh limitations on the nodes' hardware and power resources. There are also challenges in time sensitive applications for providing a reliable data delivery, such as unique network topology, diverse applications, small message size, resource constraints, frequent node failure, and congestion. The main objective of this thesis is to improve reliable data transport protocol for wireless sensor network. In order to achieve data reliability of packet by detecting and recovering lost packet, by making survey on existing reliable data transport protocols, approaches and identified Reliable Multi-Segment Transport Protocol (RMST) an optimal protocol for reliability and make a survey on queue theory. The novel approach is to develop hybrid Explicit Acknowledgments(EACK) and Negative Acknowledgments (NACK) based loss detection and recovery mechanism to grant reliability and introduce a novel queue management approach to avoid congestion in multi hop communication. Finally, new approach was tested, and evaluated with different metrics on Tinyos 2.1.x, with Five, Ten Micaz mote, meyer-heavy.txt full noisy file, in 50, 100 ms, and TOSSIM simulator on system application. In addition to system application, developed health care application in order to monitoring of patients biological states specifically on ECG and can generate early warnings if received signals turn from predefined personalized ranges. From the analysis result, the Energy Efficient Reliable Multi-Segment Transport Protocol (EERMST) end-to-end delay is 0:0: 0.00248333625 ms and NACK based RMST is 0:0: 0.0057716035 ms. The delivery ratio of ACK is somewhat closer with EERMST but has more packet delay. Generally, the EERMST has less end-to-ends delay and link delay than RMST protocol.

Keywords: Communication; Hybrid; Packet; Energy Efficient; Reliable; Acknowledgement ;Wireless sensor network

Chapter one

1. Introduction

In the current world of Technology, wireless communication technology places the major role that led us the innovative idea of using new technology for many applications. A Wireless Sensor Network is a wireless network technology consisting of spatially distributed autonomous devices using sensors. These network has the potential to change the way of living in many applications areas such as military, health care, agriculture, environmental monitoring, industry, natural disaster prevention, wildlife tracking system, building monitoring, space exploration, security, entertainment, seismic detection, care of the dependent people, and emergency management and, dam monitoring, traffic management, and many other areas [1,2,3].

In wireless sensor network, a large number of sensor nodes continually sense data from the environment and the critical event data need to be reliably delivered to the sink. The sink receives all the information from these sensor nodes, processes it, and sends them to the end user. Therefore, given the nature of error prone wireless links, presence of moving nodes and failing nodes, ensuring reliable transfer of data from resource constrained sensor nodes to the sink is one of the major challenges in wireless sensor networks. Therefore, the design of sensor network is application specific and different applications have different reliability requirements. Applications like habitat monitoring [4], periodic collection of environmental parameters like temperature, humidity etc. can tolerate a loss in data packets.

However, in event detection sensor networks critical information pertaining to the event has to be reliably transported to the central base station. Examples of event detection applications includes tsunami monitoring for detecting tsunamis in advance and issue warnings to prevent loss of life and damage. Border surveillance monitoring for detecting the attack of enemy forces, Health care monitoring to detect abnormal patient behavior, forest fire monitoring to detect and set an alarm if a fire starts somewhere in the monitored area. In addition, tracking and inventory management using sensors with RFID readers mounted on them to detect presence and location of objects, provides information regarding the condition of the object carrying the sensors [5].

Therefore, in such type of applications the sensor nodes are deployed in premeditated location. In these predetermined location the sensed data from the sensor node that having a critical piece of information to be reliably transported to the sink node for making decision on sensed data. Applications like these in which the sensors have a critical piece of information to be transported to the sink are the central focus of this thesis. In this thesis, the researcher follows the following methods. Make a survey analysis on wireless sensor network architecture, sensor node structure, application Areas, its characteristics, protocol stuck and current research works on reliability guaranteed protocols in transport layer, identifies a best protocol in which applications needs guaranteed reliability based on the transmission direction, based on loss detection, recovery mechanisms and energy efficiency. Then after identifying their opening of selected protocol, came up with a novel approach to solve the problem. Finally, design a framework that used for implementation and evaluate the performance by using different evaluation metrics.

1.1. Statement of the problem

Wireless sensor network has many unique characteristics such as unique network topology, diverse applications [6], small message size, traffic characteristics, and resource constraints usually having limited resources, including low computation capability, small memory size, low communication bandwidth, and finite, and un rechargeable battery [7]. Because of this resource constraint, using traditional transport protocols, which is TCP/UDP, is not implemented on wireless sensor network. The reasons are summarized in [8, 9, and 10]. So, in this resource constraint sensor, providing reliability is challenging.

Another challenge in providing a reliable data delivery protocol in wireless sensor networks is frequent node failure and congestion. Node failure in the sensor network can be the result of harsh environment, energy depletion, or system crashes and congestion occurs on each hop due to buffer overflow. As a multi-hop self-organized network, malfunction of several sensor nodes can cause significant topology changes and disrupt the normal functioning of the reliable protocol in a wireless sensor network. In addition, there is also accumulated delay on each hop. Therefore, reliable transport protocols are responsible to deliver data packet reliably from their sources (sensor) to their destinations (sink) without packet loss.

In order to achieve reliability of packet by detecting and recovering lost packet, there are many existing data transport layer protocol with reliability guarantee in wireless sensor network. The protocols focusing on reliable data delivery, which can be further classified according to their assumptions regarding the data transmission direction. It can be Sensors-to-sink or Sink-to-Sensors. The Sensors-to-sink protocols include RMST [11,12], RBC [13] and ERTP [14]. Sink-to-Sensors protocols include PSFQ [15], PALER [16], GARUDA [17], and HRS [18]. In critical application, the data transmission direction from Sensor -to- sink is most important to make a decision on sensed data.

1.2. Objectives of the study

1.2.1. General objectives

The main objective of this research is to improve reliable data transport protocol for wireless sensor network by comparing and contrasting different reliable transport protocols, select an optimized protocol, then to come up the shortcoming of selected protocols in terms of, reliability like loss detection and recovery, delay, scalability and energy efficiency and finally develop an algorithm for newly improved protocol.

1.2.2. Specific objectives

1.2.2.1. To explore Wireless sensor network, its sensor node structure, wireless sensor network architecture and its unique characteristics.

1.2.2.2. To study and analyze the problem of reliable data communication in wireless sensor networks.

1.2.2.3. To study queue theory in wireless sensor network, analyze the problem and develop a new de-queue algorithm for proposed system

1.2.2.4. To come up with a framework that is capable to describe the development for later implementation

1.2.2.5. Simulate, analyze, and evaluate the performance

1.3. Methodology and Tools

The methods used to accomplish the research objectives following steps have been followed.

1.3.1. Literature review

General literature review will be made on different related works to obtain an in-depth understanding of the area which include; previous studies, books, journals, websites, and published articles related to the subject.

1.3.2. Software and hardware tools

Different software and hardware required for implementation of the proposed architecture.

TinyOS-Platform for Practical Implementation

The system software is implemented in a TinyOS environment [19]. It is a lightweight open source operating system for wireless embedded sensors. It is designed to use minimal resources, and its configuration is defined at compile time by combining components from the TinyOS library and custom-developed components.

The nesC language

A TinyOS application is implemented as a set of component modules written in nesC [20]. The nesC language extends the C language with new constructs to facilitate the component architecture and multitasking. By adding direct language support for task synchronization and task management, it allows rapid development and minimizes resource usage.

TOSSIM-TinyOS Simulator

TOSSIM [21] is a discrete event simulator provided by TinyOS for high fidelity simulation of all of its applications. User can compile TinyOS application on PC framework of TinyOS instead of compiling them on real mote. Therefore, user can test, analyze, and debug developed algorithms in a controlled and repeatable environment. As it runs on PC, users can examine their code using debuggers and other development tools. TOSSIM primarily focuses on simulating TinyOS and its execution but not the real world.

1.3.3. Designing the Proposed model and simulate

The researcher will propose the desired requirements that should be addressed while conducting this work and designed its architecture.

1.3.3.1. **Wireless sensor node simulation design**

This is to achieve the objective of the research in reliable data delivery in wireless sensor network technology. To accomplish this task there are two alternatives. The first alternative is designing the sensor node or use on-the-shelf sensor. The other alternative is to model the wireless sensor node using simulation tools. Currently, since it is not possible to get (buy) the wireless sensor node (the hardware) in the local market, the second option of modeling the node using simulation tool is used.

1.3.3.2. **Evaluating the performance**

After designing wireless sensor nodes on simulation tool, it's performance is checked and analyzed by using the simulation tools with different values of parameters.

1.4. Significance of the Study

The proposed protocol will increase the lifetime sensor node, to improve reliable data delivery in sensor network with minimal delay and overhead. In which application that needs critical information applications like forest fire monitoring, health care monitoring, and battlefield surveillance also require high end-to-end reliability.

1.5. Scope and Limitation

The scope of this study is limited to propose reliable data transfer protocol in Wireless sensor network in data transport layer with single packet delivery.

Chapter Two

2. Literature Review and Related work

2.1. Overview of wireless sensor network

A sensor network is an infrastructure comprised of sensing (measuring), computing, and communication elements that gives an administrator the ability to instrument, observe, and react to events and phenomena in a specified environment. The administrator typically is a civil, governmental, commercial, or industrial entity. The environment can be any physical world in which sensors have deployed. Currently, this network has been widely used and remarkable interest among researchers because of their potential usage in a wide variety of applications. Sensor nodes are inexpensive portable devices with limited processing power and energy resources. Sensor nodes can be used to collect data from the environment, locally process this data and transmit the processed data back to the user.

2.1.1. Sensor Node Structure

A sensor node typically consists of four basic components: a sensing unit, a processing unit, a communication unit, and a power unit, which is shown in Figure 2.1 [23].

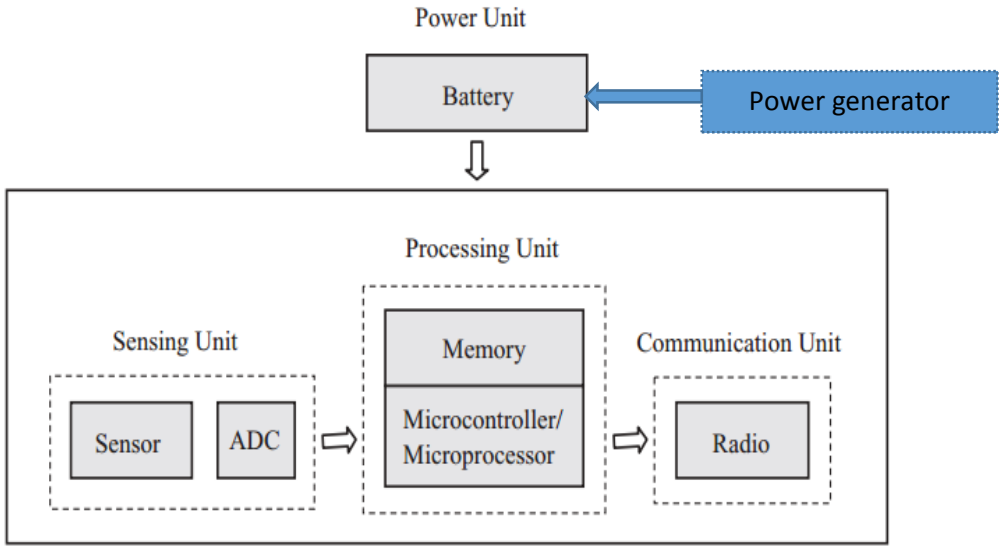


Figure 2.1.: Sensor node structure

The sensing unit usually consists of one or more sensors and analog - to - digital converters (ADCs). The sensors observe the physical phenomenon and generate analog signals based on the observed phenomenon. The ADC converts the analog signals into digital signals, which is then fed to the processing unit.

The processing unit usually consists of a microcontroller or microprocessor with memory, the microprocessor is responsible for managing the communication protocols, processing collected data from the on-board sensors, provides intelligent control to the actuator node and performing the power management. The Memory is used for storing programs and data. There are two types of memory units based on different needs for storage in a sensor node. The microprocessor itself contains some on-chip memory used to store system software. There is also typically flash memory available where users can store their own applications and data.

The communication unit consists of a short - range radio for performing data transmission and reception over a radio channel. The essential task is to convert a bit stream (or a sequence of bytes or frames) coming from a microcontroller and convert them to and from radio waves. For practical purposes, it is usually convenient to use a device that combines these two tasks in a single entity. Such devices are called transceivers.

The communication frequencies of the sensor nodes are between 433 MHz (in some early generations of sensor nodes) and 2.4 GHz (the most commonly used frequency) [22]. The communication unit has four operational states: transmit, receive, idle and sleep.

The power unit consists of a battery for supplying power to drive all other components in the system. In addition, a sensor node can also be equipped with some other units, depending on specific applications. For example, a global positioning system (GPS) may be needed in some applications that require location information for network operation, power generator that supplies additional power to actuator to maintain availability of data. A motor may be needed to move sensor nodes in some sensing tasks. All these units should be built into a small module with low power consumption and low production cost.

2.1.2. Network Architectures

A sensor network typically consists of a large number of sensor nodes densely deployed in a region of interest, and one or more data sinks or base stations that are located close to or inside the sensing region, as shown in Figure 2.2 [23].

The sink sends queries or commands to the sensor nodes in the sensing region while the sensor nodes collaborate to accomplish the sensing task and send the sensed data to the sink(s). Meanwhile, the sink(s) also serves as a gateway to outside networks, for example, the Internet. It collects data from the sensor nodes, performs simple processing on the collected data, and then sends relevant information (or the processed data) via Internet to the users who requested it or use the information.

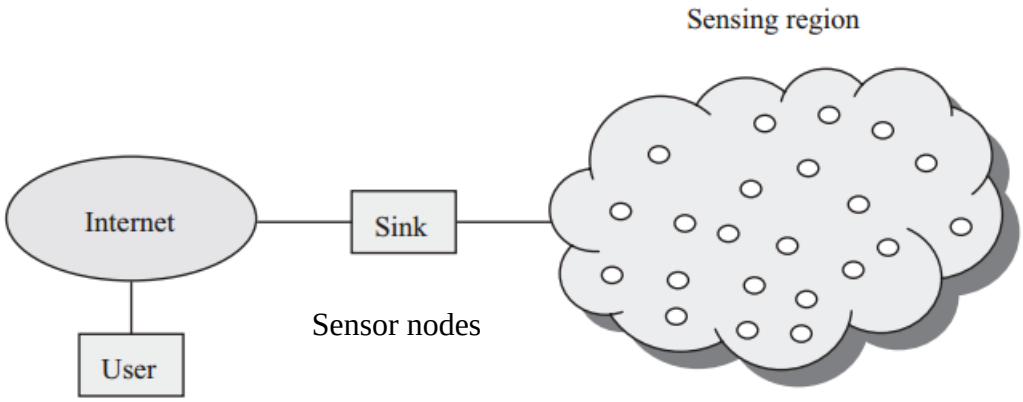


Figure 2.2.: Sensor network architecture [23]

2.1.3. Wireless Sensor Network Characteristics

Wireless sensor networks have the following unique characteristics and constraints [23]:

- ✓ **Dense sensor node deployment:-**
  Sensor nodes have usually deployed densely and can be several orders of magnitude higher than other network.
- ✓ **Battery-powered sensor nodes:-**
  Sensor nodes are usually powered by battery and deployed in a harsh environment, where it is very difficult to change or recharge its batteries.
- ✓ **Severe energy, computation, and storage constraints:**
  Sensors nodes are having highly limited energy, computation, and storage capabilities.

- ✓ **Self-configurable:** Sensor nodes are usually randomly deployed and autonomously configure themselves into a communication network.
- ✓ **Unreliable sensor nodes**: Since sensor, nodes are prone to physical damages or failures due to its deployment in harsh or hostile environment.
- ✓ **Data redundancy:** In most sensor network applications, nodes are densely deployed in a region of interest and collaborate to accomplish a common sensing task. Thus, the data sensed by multiple sensor nodes typically have a certain level of correlation or redundancy.
- ✓ **Application specific:** A sensor network is usually designed and deployed for a specific application. The design requirements of a sensor network change with its application.
- ✓ **Many-to-one traffic pattern:** In most sensor network applications, the data sensed by sensor nodes flow from multiple source sensor nodes to a particular sink, exhibiting a many-to-one traffic pattern.
- ✓ **Frequent topology change:** Network topology changes frequently due to the node failures, damage, addition, energy depletion, or channel fading.

2.1.4. Wireless Sensor Network Applications

Recent advance in the integration and miniaturization of physical sensors, embedded microcontrollers and radio interfaces on a single chip; wireless networking; and micro-fabrication have provided a new generation of wireless sensor networks suitable for many applications. Wireless sensor networks can be used for military applications, habitat monitoring, machine health monitoring and guidance, traffic pattern monitoring and navigation, health care monitoring, Agriculture monitoring, and infrastructure monitoring. We briefly describe these applications and for more details in [23].

Military Applications

Wireless sensor networks can be a part of military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting systems. The rapid deployment, self-organization, and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military applications. Some of the military applications of sensor networks are monitoring friendly forces, equipment and ammunition; battlefield surveillance;

reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical attack detection (recently is considered as one of the critical types of attacks) and reconnaissance.

Environmental Applications

Another important area of wireless sensor networks used in environmental applications which includes tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock [24, 25].

Home Applications

Home automation: As technology advances smart sensor nodes and actuators can be incorporated into appliances, such as vacuum cleaners, microwave ovens, and refrigerators. These sensor nodes inside the devices can communicate with each other and with the external network via the Internet or satellite. They allow end users to control home devices locally and remotely. Accordingly, wireless sensor networks enable the interconnection of various devices at residential places with convenient control of various applications at home [26].

Commercial Applications

Wireless sensor are used in commercial applications for monitoring material fatigue, building virtual keyboards, managing inventory, monitoring product quality, constructing smart office spaces, environmental control in office buildings, robot control and guidance in automatic manufacturing environments. Such as interactive toys; interactive museums; factory process control and automation; monitoring disaster area; smart structures with sensor nodes embedded inside; machine diagnosis; transportation; factory instrumentation; local control of actuators; detecting and monitoring car thefts; vehicle tracking and detection; as well as instrumentation of semiconductor processing chambers, rotating machinery, and wind tunnels.

Healthcare Applications

Some of the health applications of sensor networks involve providing interfaces for the disabled, integrated patient monitoring, diagnostics, drug administration in hospitals, tele-monitoring of human physiological data, and tracking and monitoring doctors patients inside a hospital.

2.2. Protocol Stack for Wireless Sensor Network

The protocol stack for wireless sensor networks consists of five protocol layers: the physical layer, data link layer, network layer, transport layer, and application layer, as shown in Figure 2.3 [23].

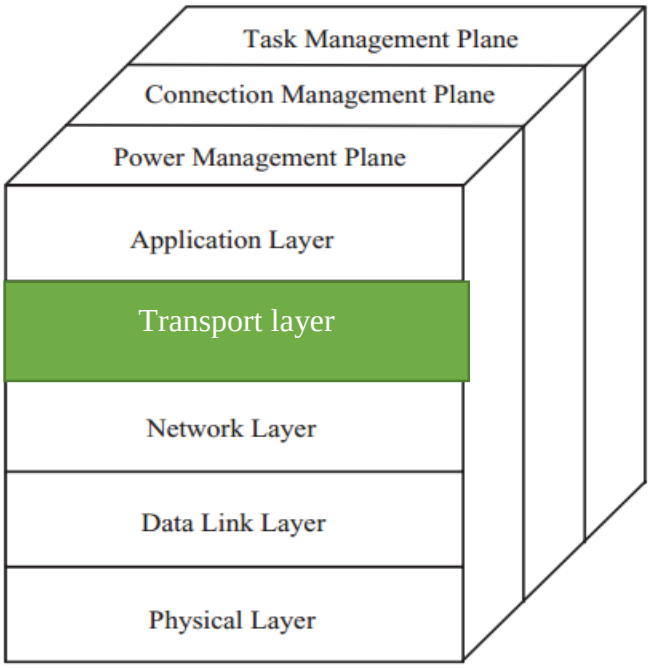


Figure 2.3.: Protocol stack for sensor networks

The application layer contains a variety of application layer protocols to generate various sensor network applications. The transport layer is responsible for reliable data delivery required by the application layer. The network layer is responsible for routing the data from the transport layer. The data link layer is primarily responsible for data stream multiplexing, data frame transmission and reception, medium access, and error control. The physical layer is responsible for signal transmission and reception over a physical communication medium, including frequency generation, signal modulation, transmission, and reception, data encryption, and so on.

On the other hand, the protocol stack can be also divided into a group of management planes across each layer [27], including power, connection, and task management planes. The power management plane is responsible for managing the power level of a sensor node for sensing, processing, and transmission and reception, which can be implemented by employing efficient power management mechanisms at different protocol layers. For example, at the MAC layer, a

sensor node can turn off its transceiver when there is no data to transmit and receive. At the network layer, a sensor node may select a neighbor node with the residual energy as its next hop to the sink.

The connection management plane is responsible for the configuration and reconfiguration of sensor nodes to establish and maintain the connectivity of a network in the case of node deployment and topology change due to node addition, node failure, node movement, and so on. The task management plane is responsible for task distribution among sensor nodes in a sensing region in order to improve energy efficiency and prolong network lifetime.

2.3. Transport Layer

In this section the author presents an overview of general reliability issues in the data transport protocol for wireless sensor networks. In general, the transport layer is responsible for reliable end - to - end or hop-by-hop data delivery between sensor nodes and the sink(s).

There are two major functions in the transport protocol of wireless sensor networks, that are reliable data delivery and congestion control. Reliable data delivery requires when packets are lost in a multi-hop wireless sensor network, some or all of the lost packets can be detected and the lost information recovered by appropriate mechanisms [28]. Congestion occurs when many sensor nodes send data to the sinks and the amount of the data traffic exceeds the network capacity. In the case of congestion in a wireless sensor network, nodes start to drop packets or the delay of the packets. This frequent dropping of packets leads to waste of energy and offsets to achieve reliability.

Due to unique nature and specific requirements of different applications of sensor networks, due to the low energy, computation, and storage constraints of sensor nodes designing reliable data transport protocol may faces challenges. Because of these unique characteristics, traditional transport protocols cannot be applied directly to sensor networks without modification. For example, the conventional end - to - end retransmission based error control and the window - based congestion control mechanisms used in the transport control protocol (TCP) cannot be used for sensor networks directly because they are not efficient in resource utilization.

On the other hand, sensor networks are application specific and usually deployed for specific application areas. Different applications areas may have different reliability requirements, which

have a big impact on the design of transport layer protocols. For example, consider a sensor network deployed in a chemical plant to detect harmful gas, health care monitoring, and battlefield surveillance which have crucial for sensor nodes reliably transport every sensor reading data back to the sink. On the other hand, some applications may not require simple 100% guaranteed transmission of data packets [29].

The reliable data delivery in sensor networks primarily occurs in two directions: upstream and downstream. In the upstream, the actuator nodes transmit their sensed data to the sink(s), while in the downstream the data originated from the sink(s). For example, queries, commands, and programming binaries are sent from the sink(s) to the source sensor nodes.

Reliable protocols in wireless sensor networks are protocols that can reliably deliver packets from their sources to their destinations without packet loss, providing congestion control, and energy efficiency.

In the section, 2.3.1 presents the general design consideration of transport protocol, in 2.3.2 presents general issues reliable data delivery and in section 2.3.3, presents the basic approaches in reliable data delivery in wireless sensor network, in section 2.4 presents the operating system used in wireless sensor network. In section 2.5 presents a variety of related work in transport layer protocols for wireless sensor networks.

2.3.1. Data Transport Protocol with Design Consideration in Wireless Sensor Network

Unlike the traditional TCP/IP network, each sensor node in a wireless sensor network resource constraint which has very limited power, bandwidth and storage space and has to cope with a loss wireless channel. The reliable data transport protocols that are widely used in the internet such as TCP and UDP are not suitable for wireless sensor networks [23].

When designing reliable data transport layer in wireless sensor network should have to consider the following:

I. The reliable data transport protocol should be able to provide robustness to the network and be able to adapt to different scenarios, such as node failure and route changes.

II. Since, a wireless sensor network is an energy-constrained multi-hop network, a reliable data transport protocol should try to avoid any packet drop unless absolutely necessary. This is because data packets normally have to travel many hops before they reach their destinations. If a packet is dropped during the transmission, all the energy, and bandwidth that have already been spent on the packet in the previous hops are completely wasted.

III. Fairness may be another consideration in the reliable data transport protocol design. In wireless sensor network, most of the data flows are transmitted from many sensor nodes that deployed in physical environment to a sink node. In such a multi-hop many-to-one, routing structure nodes deployed far from the sink are not equal in message delivery, which can often result in unfairness. The packets delivery of nodes far away from the sink have a higher possibility to get lost during transmission than packets from closer sink nodes. Such unfairness for different nodes can cause problems in some applications and thus may need to be considered when designing a reliable data transport protocol [29].

2.3.2. General Issues in Reliable Data Delivery

In general, a reliable data transport protocol should cover the following dimensions:

- **Communication type:** Reliable protocols should provide reliable delivery of a single packet, blocks of packets or streams of packets [30]. Streams of packets are a continuous data stream. Periodic event monitoring is an example application type using streams of packets. Blocks of packets are segments of a complete data stream. A block of packets consists of a fixed number of data packets. Reliably delivering a single packet can be very important for queries or highly aggregated data, while delivery of blocks of data is necessary for many wireless sensor network applications such as remote network reprogramming. The cases of delivering a single packet and delivering blocks of packets can use very different underlying protocol mechanisms. The primary approaches for single packet delivery are ACK-based retransmission and transmission of multiple redundant packets. A wider variety of options exists for reliable delivery of blocks of data or streams of data. NACK-based approaches and multi-paths approaches are commonly used in such protocols.

- **Reliability Requirement:** Reliability requirements vary across different wireless sensor network applications. For sensors-to-sink delivery, the reliability requirement is either 100% guaranteed data delivery (or as close to this possible) or a percentage or probabilistic delivery requirement (for example 75% reliability). For sink-to-sensors delivery, the reliability requirement can be classified into four categories:

  I. Delivery to the entire network;
  II. Delivery to sensor nodes in a sub-region of a network (location based delivery);
  III. Delivery to the core members of the network that are able to cover the entire sensing field; and
  IV. Delivery to sensor nodes with a probabilistic reliability requirement

- **Upstream and downstream delivery**: In wireless sensor networks, it can be assumed that most communications are not between arbitrary peer nodes. As a data collecting network, the data flow in wireless sensor networks is normally from sensor nodes towards a single sink/gateway node [29].

  Most research works in wireless sensor network is to granting packet reliability from sensor-to-sink transmission direction, the protocols categorized under this transition direction are RMST [11,12], RBC [13], and ERTP [14]. However, in some scenarios, a reliable protocol for downstream communication is also important. Reliable downstream protocols include PSFQ [15], PALER [16], GARUDA [17], and HRS [18].

2.3.3. Basic Approaches in Reliable Data Delivery

There are three basic approaches to achieving reliability in wireless sensor networks. The first one is End-to-End vs Hop-by-Hop Error Recovery approach [15], the second one is ACK vs NACK approach [32] and, the third Sender Retransmission vs Forward Error Correction [30].

2.4. Operating System for Wireless Sensor Network

Wireless sensor nodes are resource constraint in their nature with low memory size, low bandwidth, and low computational capacity. Although these nodes have different characteristics, their basic hardware components are the same: a physical sensor, a microprocessor or microcontroller, a memory, a radio transceiver, and a battery. Therefore, these hardware components should be organized in a way that makes them work correctly and effectively without a conflict in support of the specific applications for which they are designed. Each sensor node needs an operating system (OS) that can control the hardware, provide hardware abstraction to application software, and fill in the gap between applications and the underlying hardware. In traditional OS, system software operates between application software and hardware, which often designed for workstations and PCs with plenty of resources. This is usually not the case with sensor nodes in wireless sensor networks. There are several sensor-based operating systems used today; some of them are TinyOS, Mantis OS, Mate, and EYESOS, SenOS etc [31]. In the following Section 2.8.1, the author will be reviewing TinyOS, since it is the de-facto standard and very mature Operating System for wireless sensor networks [19].

Tiny Operating System (TinyOS)

TinyOS, as its name implies, can be described as a miniature framework designed for embedded systems that require very aggressive resource management due to the highly constrained nature of their resources such as power and available memory [19]. It implements the hardware abstraction layer and scheduler of a conventional operating system, allowing generic programs that may have no knowledge of the intricate details of the operations supported by the underlying hardware components (such as sensors) to use well-defined interfaces to interact with these components. Its C-language-like framework provides an interface to core system components, allowing a programmer to manage various services of the system.

TinyOS provides software abstraction for hardware components such as its communication, routing, sensing, and storage subsystems. A software component in TinyOS refers to abstractions of specific services provided by either another software component or a hardware component. A software component consists of any number of the following: Modules and Configurations.

Modules are the lowest form of abstraction provided by the TinyOS operating system. They implement program logic that directly addresses a software component and can also provide a particular set of services, thereby enabling the reuse of software components. Other software components can interact with a module through its defined interface, which specifies a set of operations/services to a particular module. On the other hand, abstractions of multiple modules and other configurations grouped together form a newly abstracted component referred to as a configuration.

A configuration can be visualized as a super component consisting of several subcomponents to provide a single unified interface. Configurations wire a set of components defined in a component signature. That allows two or more components to communicate with each other.

2.5. Related works

2.5.1. Transport Protocol for Wireless Sensor Network

Several transport protocols have been designed for wireless sensor networks. A list of relevant related works on transport protocol in wireless sensor network has shown in Figure 2.4.

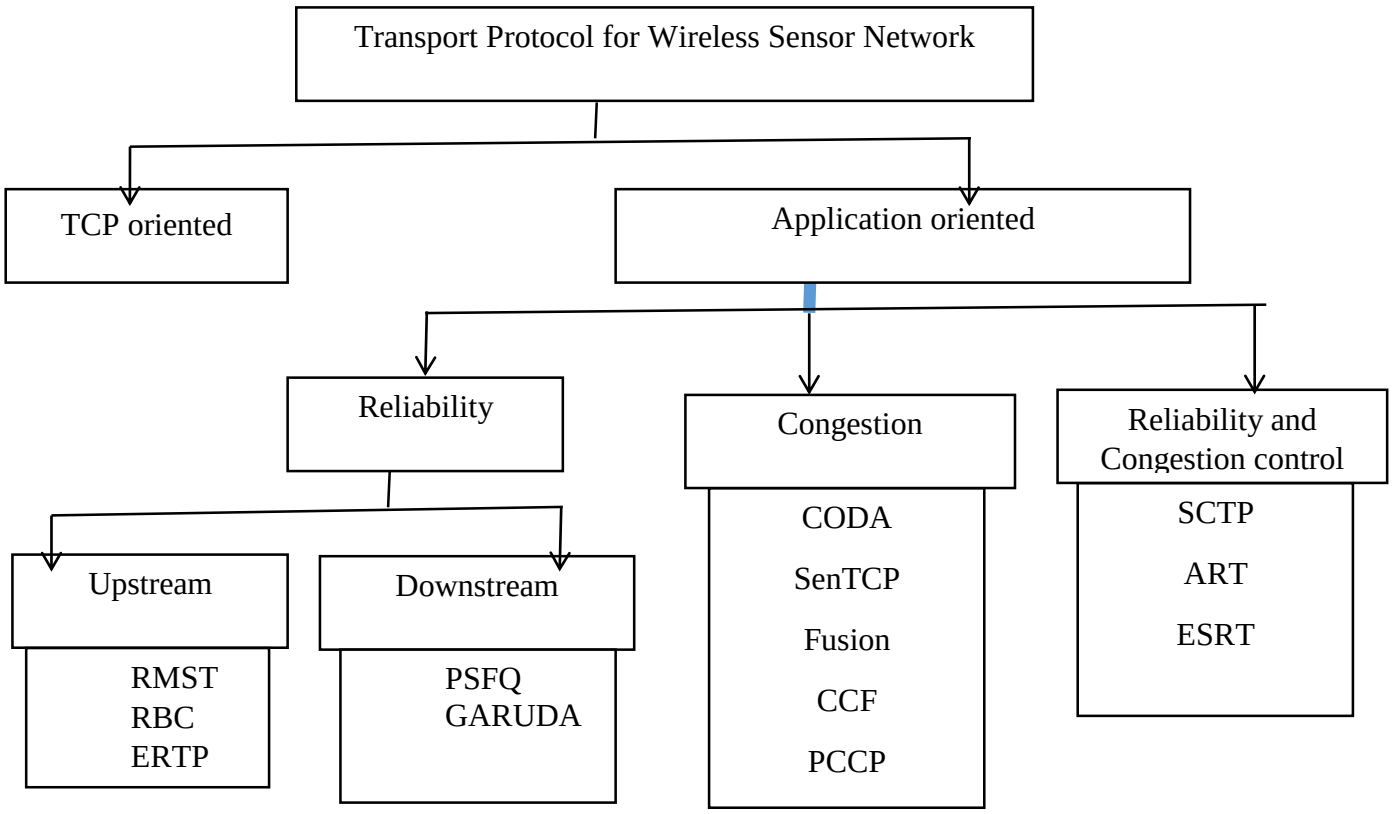


Figure 2.4: The existing transport protocol in wireless sensor network

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are two traditional transport protocols that are currently used for the internet. TCP is a reliable end-to-end transport protocol, which is widely used for data services and very efficient for wired networks. The basic idea of TCP congestion control is that TCP send probe to the network for available resources and increase the transmission rate until packet losses are detected. TCP takes packet loss as indication network congestion and trigger appropriate congestion control scheme.

However, TCP is a connection oriented that cannot be directly implemented and may not suitable for data transport in wireless sensor networks because wireless sensor network has unique characteristics and applications specific. The reasons have been summarized in [8, 9, 10]. In TCP, Setting up a connection using the three-way handshake is costly, slow, and not suitable for many urgent tasks in wireless sensor networks. Besides that, TCP has degraded throughput in wireless system, especially in situations with a packet loss rate because TCP assumes that packet loss is due to congestion and triggers rate reduction whenever packet loss is detected. Another reason is preprocessing or aggregation of data in intermediate nodes is desirable and often necessary in wireless sensor networks. Packet can be combined or change before they reach their destination. Since TCP is end-to-end protocol, it cannot handle this scenario. Lastly, TCP is not light weight due to the large amount of run time and header overhead and may not suitable for implementation in low-cost sensor nodes with limited processing, memory and energy resources. As described in Figure 2.4, there are different protocols used in wireless sensor network, based on reliability, congestion control and both. Providing reliability can be in upstream or downstream by transmission direction. Three transport protocols provide reliability in upstream data transmission direction such as RMST [11,12], RBC [13], and ERTP [14].

According to the author in [11,12] propose Reliable Multi-Segment Transport (RMST) protocol with NACK-based approach that has primarily timer-driven loss detection and repair mechanisms. It is designed for relatively long-lived data flows from source nodes to a sink node, although it could be applied to other contexts as well. This protocol supports both cache and non-cache mode. In cache mode of RMST, all-intermediate nodes on a path as well as the sink maintain a cache that stores all segments being sent. When a node detects a missing segment, it creates a NACK packet that includes the missing segment's identifier and sends it back along the path to the source. When an intermediate node receives a NACK, if it has all of the missing segments listed in the NACK in its cache, it will forward them towards the sink and drops the NACK. If the node has only some or none of the missing segments, it locates and resends the missing segments it has (if any) and forwards the NACK again along the path to the source until either all missing segments are recovered or the NACK reaches the original source node.

In RMST's non-cached mode, only the sink and the source node have the ability to maintain such a cache. Thus, when a missing segment is identified, the NACK travels from the sink all the way back to the source node and the source will resend the missing segment in a timely manner.

The problems that was not address in RMST protocol are lack of congestion control, it cannot handle when all packet in the communication is lost, it cannot provide reliable delivery of single/last packets in addition to energy efficiency. The following algorithm shows the current NACK or a retransmission works to grants reliable [32].

```
Initially parent = p, children = C, seq_no = 0, no de_id = i, Buf = 0

1:   upon sensing an event E :

2:    begin

3: create a data message

M=(seq_no, node_id, data) with the event information
E;

4: Add a copy of M to message buffer

5: call SendMsg (M, p);

6: seq_no ++;

7:end

8: procedure SendMsg(M, p)

9: begin

10: send M using radio transceiver;

11: end

12: upon receiving a message:

13:  begin

14:  if message is M then

15: call SendMsg(M, p);

16: if message is NACK then

17: look for the copy of the message in the buffer Buf
;

18: if message is not found in Buf

19: call Look ForMsgInChild ( ) to find the
message in the child's buffer;

20: else

21: M = msg in Buf;

22: call ParentSelection ( ) to find a new parent;

23: call SendMsg ( M )

24: end

25:procedure LookForMsgInChild (src, seq_no)

26: begin

27: call SendMsg (NACK) to send message to child

28: end
```

Algorithm 1, NACK algorithm for achieving reliability

The author in [13] propose Reliable Bursty Converge cast (RBC) to provide real-time and reliable data transport under conditions of high-volume busty traffic. RBC improves typical network efficiency by using a window-less block acknowledgement scheme to carefully schedule packet retransmission. In the network each sender divides its packet queue into M+2 separate queues, indexed 0 through M+1 where M is the maximum number of retransmissions allowed at each hop. Packets in queue j have transmission priority over packets in queue j+1. Queue M+1 is used for free packet buffer. When the sender sends a packet to the receiver, the ID of the buffer holding the packet, as well as the ID of the buffer storing the packet to be sent next, is included with the data packet. When the receiver receives a packet from the sender, by comparing the buffer's ID with the expected buffer's ID piggybacked in the previous packet, the receiver can decide whether there is packet loss or not. However, the problem that does not addressee here is, when transmitting data packet, the sender over hears the channel to detect forwarding of the same packet by the next node. This over hearing requires additional energy and it cannot also work for last hop. In addition, the packet loss can potentially trigger window-based flow. This will reduce the transmission rate unnecessarily when packet lost may have occurred as a result of link error.

The study in [14], propose ERTP, which is designed for data streaming that ensure statistical reliable delivery of sensor data packets delivered to the sink. These protocols do not handle accumulated delay during multi hop transmission, it does not consider energy efficiency and consume more energy by over hearing the channel, it uses end-to-end Implicit Acknowledgement and retransmission when necessary. This end-to-end process have longer response time between source and destination. This will result large amount of packet loss, the packet loss result in energy waste for retransmission.

On the other hand, if some events suddenly happen after a long quiet time, a large amount of data can be generated and fill all the buffer space of the intermediate nodes in a short period. In this case, the effectiveness of ERTP protocol is compromised [16].

In Table 2.1, shows the summery of existing reliable protocols in wireless sensor network.

Protocols	Approaches							
	Design Focus	Direction	Loss Detection	Loss Recovery	Reliability	Communication Type	Unique Design	Evaluation
RMST [11,12]	Reliability	Sensors-to-sink	Selective NACK	End-to-end and hop-by hop	Packet	Block of packets	Cross-layer design (Transport and MAC layer)	NS-2
RBC [13]	Reliability	Sensors-to sink	Implicit ACK	Hop-by-hop	Packet reliability	Block of packets	Window-less queue management	49 Mica2 motes experiment
ERTP [14]	Reliability	Sensors-to sink	Implicit ACK	End-to-end	Packet reliability	Block of packets	-	TOSSIM
PSFQ [15]	Reliability, energy efficiency and scalability	Sink-to sensors	NACK	Hop-by-hop	Packet reliability	Block of packets	Pump slowly and fetch quickly	NS-2 and Rene2 motes
PALER [16]	Reliability	Sink-to sensors	Inclusive NACK	Hop-by hop	Packet reliability	Block of packets	Single inclusive NACK	Jist/Swans
Flush [35]	Reliability and time efficiency	Sensors-to sink	Selective NACK	End-to-end	Packet reliability	Block of packets	NACK and rate control design	100 MicaZ motes

Table 2.1: Summary of Existing Reliability Guaranteed Protocols

From the Figure 2.4, there are five transport protocols that provides only congestion control mechanism which are Congestion Detection and Avoidance (CODA)[33], SenTCP[34], Fusion, Congestion Control and Fairness (CCF)[35] and Priority-based Congestion Control Protocol (PCCP)[36] .However, none of these protocols has any reliability mechanism.

In addition in the Figure 2.4, there are three protocols that provides both reliability and congestion control which are Event to sink Transport Protocol (ESRT) [37], Sensor Transmission Control Protocol (SCTP) [38] and Asymmetric and Reliable Transport (ART) [39]. In these except ART protocol, the other two ESRT and STCP are upstream data transmission direction but the communication type is end-to end and use NACK and ACK approaches. Except ESRT, the other does not consider energy efficiency that consumes more energy. In Table 2.2, shows existing reliable guaranteed protocol with their energy consumption.

Table 2.2: Reliability and energy efficient comparison

Protocol	Category	Direction	Type	ACK	NACK	IACK	Sequence number out of order	Time out	Increase source sensing rate	Packet retransmission	Energy efficiency
RMST[11, 12]	Packet	Upstream	Hop-Hop	-	✓	-	-	✓	-	✓	Good
RBC[13]	Packet	Upstream	Hop-Hop	-	✓	✓	-	-	-	-	No
ERTP[14]	Packet	Upstream	End-End	-	-	✓	-	✓	-	✓	Good
PSFQ[15]	Packet	Downstream	Hop-Hop		✓	-	✓	✓	-	✓	No
GAURDA	Packet	Downstream	Hop-Hop		✓	-	✓	-	-	✓	No
ESRT[37]	Event	Upstream	Event-sink	-	-	-	-	✓	✓	-	Fair
STCP[38]	Event/packet	Upstream	End-End	✓	✓	-	-	✓	-	✓	No
ART[39]	Event/query	Both	End-End	✓	✓	-	-	✓	-	✓	No

In these Table 2.1 and Table 2.2 the authors uses a receiver feedback and sender retransmission mechanisms in order to detect and recover lost packets. They use two most commonly used mechanisms in wireless sensor network are ACK-based approach and NACK-based approach.

As clearly shown in the Table 2.1 and Table 2.2, no authors considered the energy efficiency and scalability with minimum delay in their works when data is transmitting from sensor –to sink for

providing reliability and no authors address to solve the common problem of the NACK error recovery mechanism. In addition, there is also accumulated delay in multi-hop communication. To address these accumulated delay problems only the author in [12], uses queuing theory to solve it. The author uses a priority queue based approach, which gives priority to routed data in the queue.

Therefore, this thesis addresses all packet in the communication loss problem, providing a delivery of single or last packet problem, address the problem wastage of energy as a result of retransmission, considering wastage of energy by over hearing the channel, to address the problem of congestion control and the problem accumulated delay in Multi-hop transmission. Then proposing a hybrid (EACK and NACK) based loss detection mechanism by adopting Hop-by-Hop loss recovery method using end-to-end sequence number. In addition, introducing new queueing theory approach to handle congestion.

2.5.2. Queue Management Approaches in Wireless Sensor Networks

Overview of the Queuing Model

Queuing or waiting lines can be found everywhere. Queuing for buying food at supermarket, to withdraw money from bank or a telephone waiting to be placed through to receiver are day-to-day examples. Queuing happen because the demand for a particular service is higher than the server can cope with it. The basic queuing system can be illustrated as customers arriving for service, waiting for service if the server is busy and leaving the system after service being completed. The basic queuing model is shown in Figure 2.5.

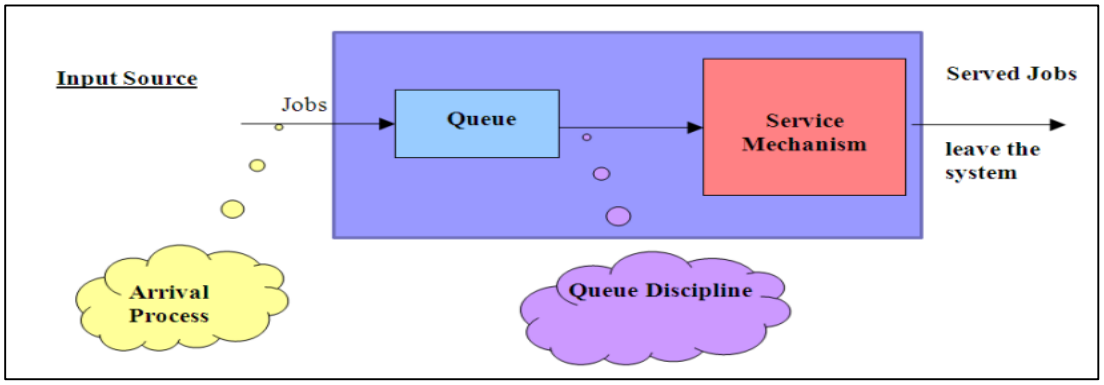


Figure 2.5: Basic Queuing Model

The basic queuing model depicted in Figure 2.5 can be identified some basic elements of the system as [40]:

Input process: describe the input process in terms random variables representing either

the number of arriving during a time interval or the time interval between successive arrivals.

Service Mechanism: it involves the number of servers, the number of message being served at any time, and the duration of service and its modes. In network of queues more than one servers arranged in series or parallel combinations. Random variables are used to characterize the service times, and the number of servers.

Queuing: the number of message waiting for service is important point of consideration.

The waiting room or queue length can be considered infinite.

Queue discipline: representing the way in which queue is organized. The rules, which are developed, consist of inserting or removing packet from the queue. FIFO, Priority, Fair Queuing are three basic queue disciplines.

In FIFO, All arriving packets are placed in a queues and the transmission of packets in order of arrival. The Packets are discarded when buffer is full and delay and loss of packets depends on inter-arrival times and packet lengths, random drops due to malicious monopolization: as one flow sends packets at a high rate and fills the buffer. Fairness among packet is not achieved with packets from higher priority class are buffered as long as there is space discarding the packets from lower priority packets. The Following Figure 2.6, illustrates the FIFO queue discipline labelled in [40].

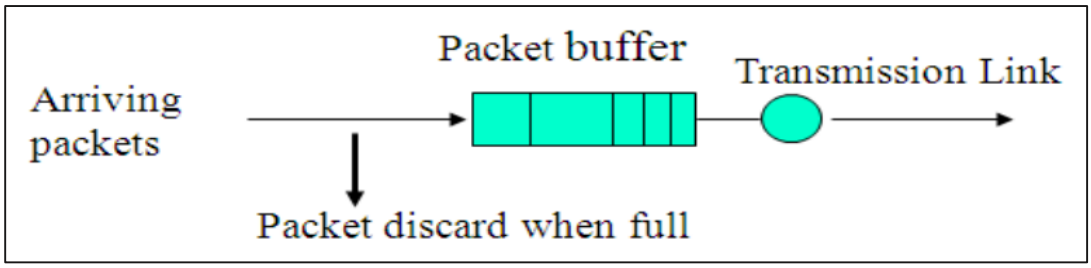


Figure 2.6: FIFO Queuing Model

In Priority Queuing: Packets are served according to priorities set by system and the packets are separated and placed in different buffers according to the priority classes. High priority classes are given to serve first than low priority. The following Figure 2.7 illustrates the Priority Queuing Model queue discipline in [40].

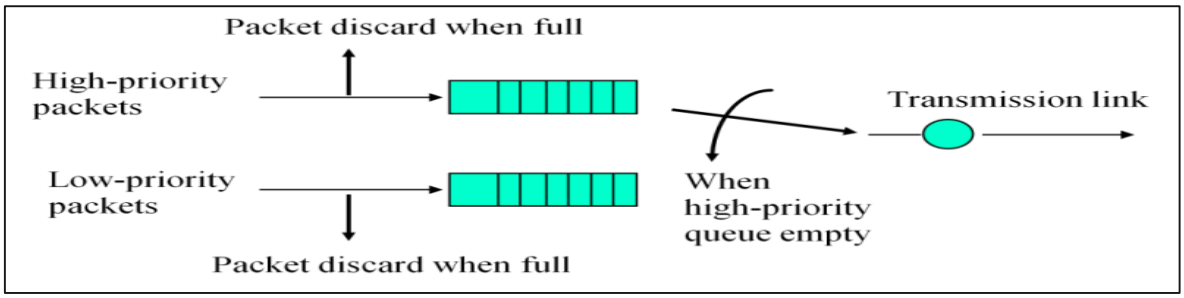


Figure 2.7: Priority Queuing Model

Having this overview of queuing model, deferent applications needs these queening model based on their requirement in order to deliver originating packet to required destination. Because of unique characteristics of wireless sensor network, it needs special queuing management.

There are different researcher's uses different approaches to manage queue efficient in wireless sensor network. The authors in [12, 41] proposes an approach for queue management in wireless sensor network based on application priority. The queue is developed based on the data type and the number of queues in a node depends on the application requirements. The base station dynamically assigns individual priority for each type of data. During forwarding heterogeneous data towards the base station, each sensor node transmits route data of its children nodes as well as its own generated data. Therefore, at any given time a sensor node may act as both a source node as well as forwarding node. When a sensor node transmits its data to the upstream direction, then it is called a child node and its immediate upstream node is called its parent. Figure 2.8 illustrates their proposed system.

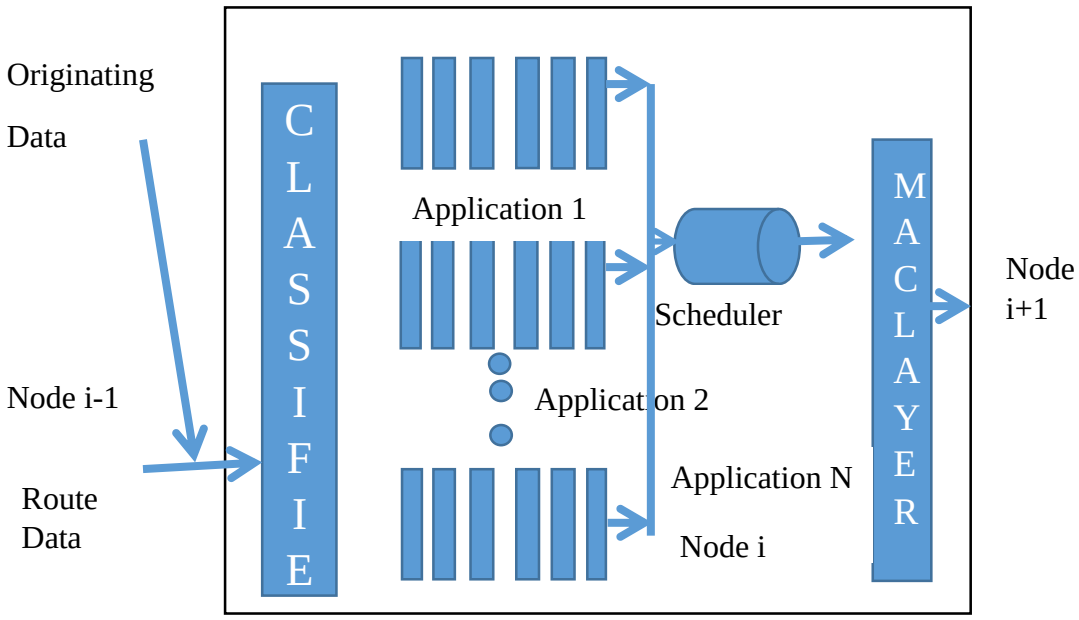


Figure 2.8: Queue management based on their application

As shown in Figure 2.8, the classifier is to classify heterogeneous traffic generated either by the same node or by incoming from different nodes. Based on the type of data, they are placed in the queue. The scheduler is defined as how many packets the scheduler schedules per unit time from the queues and forwards the packets to the MAC layer from which the packets are delivered to the next node (i.e., i+1) along the path towards the base station. The base station assigns the priorities for heterogeneous traffic that come from different application.

The scheduler schedules the queues according to the priority assigned by base station, decides the service order of the data packets from the queues, and manages the queues according to their priorities. This ensures the data with higher priority to get higher service rate. In this proposed approach the authors' uses route data has more priority than originating data.

The authors in [42] propose similar approaches by using a priority queue as illustrated in Figure 2.8. The approaches uses queuing discipline is First in First out (FIFO) principle where packets are processed according to their arrival time: a packet that arrives first will be processed first as soon as the processor is free. When a packet arrives at a node i, and thanks to its type of service field, node's i classifier allocates this packet to a virtual queue at level that is appropriate to the packet's class of traffic. The packets with higher priority are served first, the lower second, and so on, according to FIFO discipline, which is illustrated in Figure 2.9.

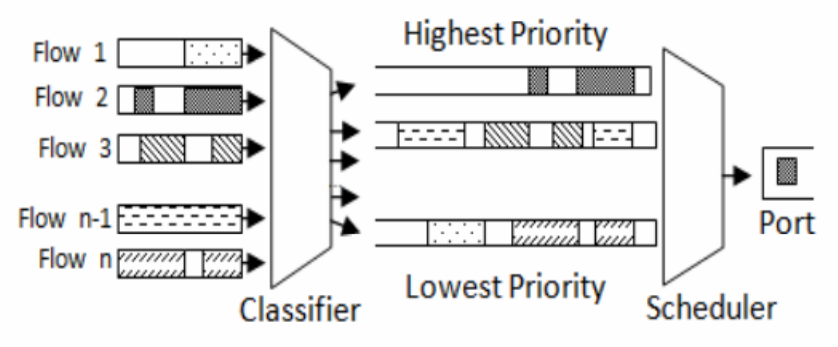


Figure 2.9: Priority queue

According to the authors in [43,44] has similar approach as stated earlier which is designed for data queuing and transmission procedure using Beacon Advertisement-based Time Division Multi-Access (Beacon-ATMA) method. Their Proposed approaches are used for deployment of home automation applications.

The protocol is capable of differentiating different traffic from heterogeneous sensors, which is essential to the data delivery requirements for targeted application. This protocol implements data prioritization in its queuing and scheduling algorithms. In this, Once data is generated from a sensor, it is encapsulated by the wireless node housing sensor into a data frame that containing information such as node ID, time of generation, and priority level of the source sensor. The data frame is then inserted into the transmit queue of the node according to decreasing priority. If there should be any existing data on the queue that are of equal priority, then the data of the same level are ordered according to generation time and any data frames that are of lower priority level are pushed further down the queue.

However, in delay sensitive application like target detection, health care monitoring, Tsunami monitoring, border surveillance, nuclear plant radiation monitoring, and forest fire monitoring the sensed critical data must be delivered as quickly as well as reliably as possible. In order to handle these opening, the researcher also proposes a novel approach. In this approach, high priority data and low priority data are valuable for monitoring applications. Therefore, researcher assumed that, either the originating data or fresh sensed data and route data has equal value of importance but the packet that needs retransmission have higher priority. The reason behind this is that, as routed data that have already traversed some hop(s), their loss would cause more wastage of network resources than that of the originating (i.e., source). On the other hand, low priority data may be delayed or dropped during high traffic situations , it also is possible that low priority data may be forced to wait for extended periods, and in extreme conditions, may never reach their destination at all.

2.6. Data Aggregation

In wireless sensor network, sensor nodes are usually deployed in large or even huge number for continuous monitoring of specified application area. Because of the dense pattern of sensor deployment, neighboring sensor nodes may sense similar data on a specific phenomenon, which is referred to as spatial correlation. Since sensor nodes are run by battery power, it is critical to perform every operation in energy efficient manner. For this purpose, it is desirable for a sensor node to remove the redundancy in the data received from its neighboring nodes before transmitting the final data to the Base Station. Data aggregation is an effective technique for removing data redundancy and improving energy efficiency in WSNs.

The basic idea is to combine the data received from different sources so that the redundancy in the data is minimized and the energy consumption for transmitting the data is reduced in the aggregation process [45]. There are different data aggregations techniques are available for energy efficiency for different network architecture. It can be Flat, Hierarchical, or Tree Based Network architecture. The data aggregations categorized under Flat network architecture are Directed Diffusion (DD) [46] and sensor protocol for data via negotiation (SPIN) [47]. In this network, each sensor node plays the same role and is equipped with approximately the same battery power.

In Hierarchical network, in which data aggregation data has to be done at special nodes, with the help of these special node it can reduce the number of data packet transmitted to the sink. Therefore, with this network architecture it improves the energy efficiency of the whole network. In this hierarchical data-aggregation, it can be Cluster-Based Networks data aggregation or Chain Based Networks Data Aggregation. For example, LEACH (Low Energy Adaptive Cluster Hierarchy) [48] and HEED (Hybrid Energy Efficient Distributed) [49] categorized under cluster based network. All regular sensors can sense and send their data packet to a cluster head (local aggregator). The cluster head aggregates data packet from all the regular sensors in its cluster and sends the concise digest to the base station. With the help of the local aggregator node, we save the energy of the sensors. In Chain Based Networks, Power Efficient Data Gathering Protocol for Sensor Information Systems (PEGASIS) [50] technique is used for data aggregation. In this network, each sensor sends data to the closer neighbor and all sensors are structured into a linear chain for data aggregation.

In Tree Based data aggregation technique, data is transferred from leaves node to sink node and parent nodes do aggregation process. For example, TAG (Tiny Aggregation) technique is used under this category that performs the data aggregation process with the help of queries process and provides service for aggregation in distributed, low-power, wireless environments [51]. Figure 2.10, illustrates that data aggregation is the process of aggregating the sensor data using aggregation approaches.

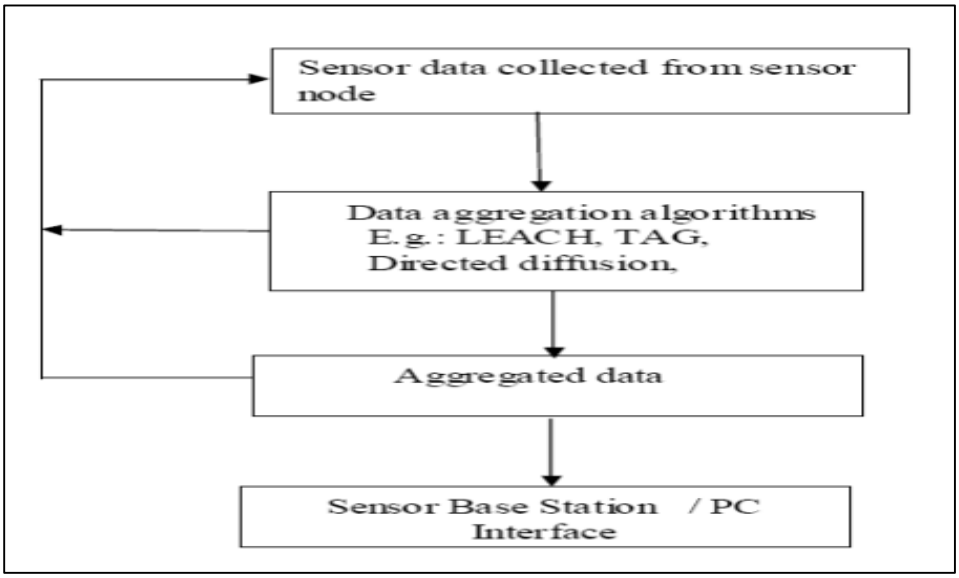


Figure 2.10: Architecture of data aggregation

In our proposed solution, the author has recommended to use a hierarchal cluster based data aggregation algorithm, which is described under Chapter 3 in section 3.7.

Chapter Three:

3. Proposed solution

From the related work of literature review, it is understood that a number of research has been done in reliable data delivery protocol in transport layer for wireless sensor network. In this study the author have contribute to improve and evaluate reliable data transfer protocol based on hop-by-hop loss detection and hybrid based loss recovery mechanism. Thus, the goal of this protocol is to provide high reliability of packet delivery from sensor to sink communication with low system overhead and network delay. In this chapter, the author presents design considerations in section 3.1, Design scenario for proposed solution in section 3.2, proposed framework in section in 3.3. The Protocol used in our proposed approach, hop-by-hop loss detection and recovery scheme and overview of Automatic Repeat request in section 3.4.2, and proposed transport protocol approach and proposed queue management approach in section 3.5 and 3.6.

3.1. Design Considerations:

Wireless sensor networks should be designed with an eye to energy conservation, congestion control, and reliability in data dissemination, security, and management. These issues often involve in one or several layers of the hierarchical protocol. It also studied either separately in each layer or collaboratively in cross layers. For example, congestion control may involve only the transport layer, but energy conservation may be related to the physical, data link, network, and perhaps all other high layers. Generally, transport control protocols' design includes two main functions: congestion control and reliability. In this thesis, the author is focusing on reliability, in which critical applications that need reliable transmission of each packet, and to provide packet-level reliability have considered. The following listed points are considered for proposed protocol:

1. The protocol should be able to provide robustness to the network and be able to adapt to different scenarios, such as node failure and route changes.
2. Fairness may be another consideration in the reliable data transport protocol design. In wireless sensor network, most of the data flows are transmitted from many sensor nodes that deployed in physical environment to a sink node.

In such a multi-hop many-to-one, routing structure nodes deployed far from the sink are not equal in message delivery, which can often result in unfairness. The packets deliveries of nodes far away from the sink have a higher possibility to get lost during transmission than packets from loser sink nodes. Such unfairness for different nodes can cause problems in some applications and thus may need to be considered when designing a reliable data transport protocol [29].

3. Energy Efficiency, a sensor node usually has limited energy. For this reason, it is most important to be considered here for designing a transport protocol to keep high-energy efficiency in order to prolong the network lifetime. In multi-hop network structure, the transport protocol should be minimizing the sensor energy consumption. In addition, it should try to avoid any packet drop unless necessary. This is because data packets normally have to travel many hops before they reach their destinations. If a packet has dropped during the transmission, all the energy, and bandwidth that have already been spent on the packet in the previous hops are completely wasted. However, there are cases where packet dropping is inevitable. Since sensor nodes have limited storage space, when the buffer is full of data packets and a new packet arrives, a data packet earlier or old data in the transmission must be dropped.

   In this study, the author assumed that the older data packet and newer data packet are given equal importance, except the missing packets that needs retransmissions are given higher priorities over the transmissions of newer packets. As a result, when retransmission occurs, data packets that are already stored in the transmission queue have to wait until retransmissions have to send and the acknowledged packet has to be dropped from the individual node.

3.2. Design scenario

This section describes the scenario of proposed protocol, in which the applications that needs critical information sensed by sensor node to be delivered with minimal packet loss to receiver. For application like real-time monitoring, Safety applications, Health care monitoring, Battle filed monitoring that sensors nodes to be reliably delivers critical packets to its destination and all data collected from sensor have the same importance.

For such applications single packet delivery is important when nodes send a critical piece of information to the sink node. Likewise, the neighboring sensor nodes may sense similar data on a specific phenomenon to be aggregated and transferred to base station. The proposed protocol is applicable in multi-hop network architecture, in which a sensor node transmits its sensed data toward the sink via one or more intermediate nodes, in order to maximize the packet delivery and reduce the energy consumption for communication. The data aggregation process takes place on intermediate node.

The packet loss in wireless sensor network is usually due to the quality of the wireless channel, sensor failure, congestion and buffer overflow. To guarantee reliability, it is important to recover lost packets in order to transmit correct information and minimize energy consumption, recover lost segments; managing buffer of the node. Therefore, the author uses Hop-Hop loss recovery mechanism and hybrid-based (NACK and EACK) loss detection mechanisms. In addition, the author introduces a new queue management approach in order to minimize accumulated delay in multi-hop network at each hop.

3.3. The Proposed Network Framework

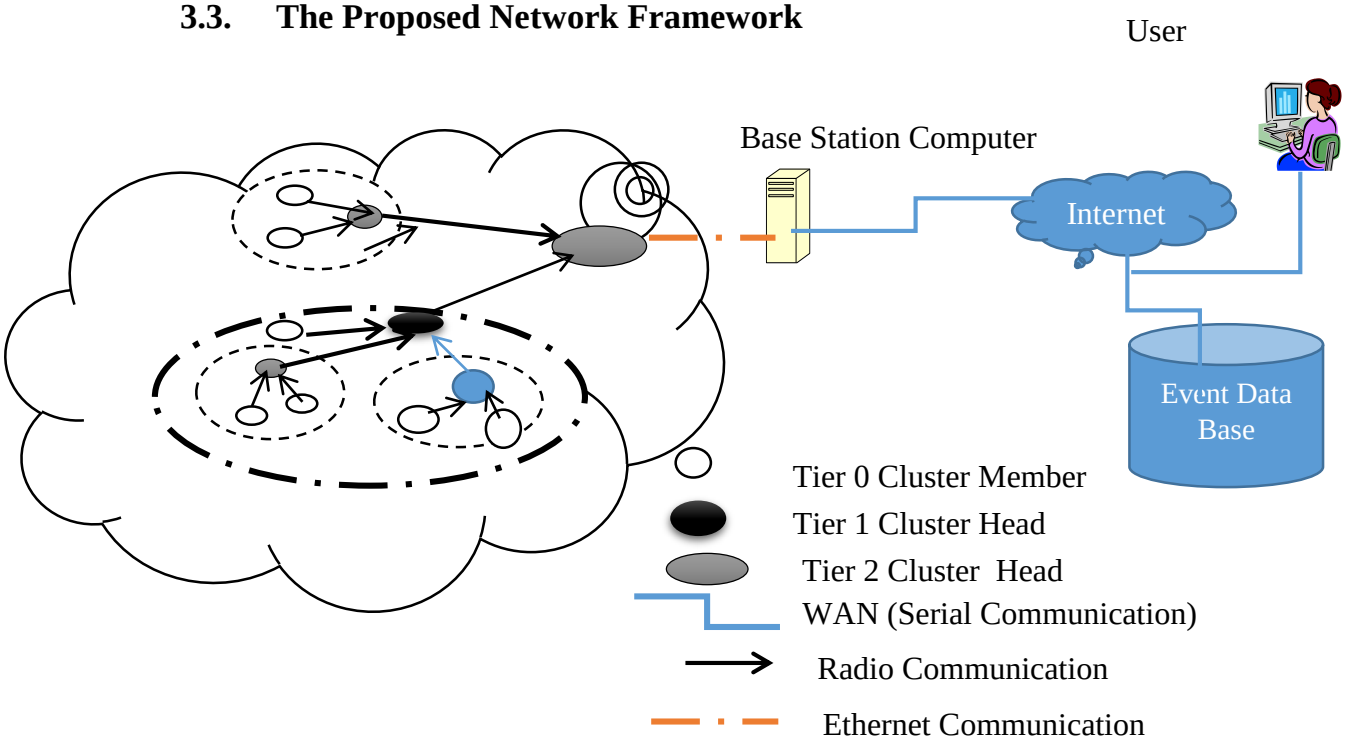


Figure 3.1.: Proposed network framework for reliable data delivery

This diagram shows the interactions between each major component in the network. Beginning with the network of sensors, information travels to the base station through the Tire 2 cluster head. The Tier 2 cluster head appoints a secondary cluster head (Tier 1) with in its cluster to acts as a backup manager and collects aggregated information from Tier 1 cluster head. In addition, has responsible for coordinating activates with in the cluster and forwarding information between secondary cluster head. Then sends its information to its base station. Notice that, Tire 2 and Tier 1 are known to their cluster members. Tire 1 has also responsible for coordinating activates with in the cluster, collects information from cluster members and forwarding information to Tire 2 cluster head. Tire 0 are all wireless sensor nodes that has a capable to collect or sense physical phenomena and has a responsible to transmit their sensed data directly to Tire 1 cluster head or if there are not cluster heads it directly sends to sink node.

Once the data has been parsed by the base station, it is sent to the database. Finally, users are able to view the status of the sensor network by using an application that queries event information from the data storage.

Base Station:

Our base station will be a computer connected to the Tire 2 cluster head, the node to which all other nodes will be programmed to send their packets. This base node will need to parse sensor data, and format the data for transmission into an appropriate storage medium. The communication between base station computer and sensor networks is by using Ethernet communication.

Data Storage:

For effective implementation of the sensor network, they require a centralized storage mechanism allowing for multiple concurrent clients reading and modifying the system at a given time. This storage mechanism must be able to hold the readings of an arbitrarily large number of sensor nodes, effectively organizing it for efficient read and write access.

User Interface:

Given that at this point, all of our information will be housed in a central medium, they require an appropriate user interface to parse and display this data in a user-friendly manner. At the minimum, this interface needs to display the status of each node, along with a description of past node readings. Additionally, users should be able to specify their own names for nodes, forgoing the default behavior of identifying via unique node numbers.

3.4. Transport Protocols used in our proposed approach

The main purpose of transport layer protocol in wireless sensor network is to achieve reliable data transport and to perform flow control and congestion control with low energy consumption in order to extend the lifetime of a wireless sensor network. The transport protocol depends on the network scenario and the application where the node is deployed. In this, study the researcher selects the best transport protocol from the reliability scheme in wireless sensor network based on transmission direction, loss detection, recovery mechanisms, and energy consumption.

There are some reliable transport layer protocols available for wireless sensor network as it is clearly described in section 2.5. Reliable Multi-Segment Transport (RMST) is well-known reliable transport protocol under upstream (sensor to sink) data transmission in wireless sensor network.

For the proposed approach any sensor node is deployed anywhere for application like health care monitoring, safety detection, battle filed monitoring etc. in which the sensed data from the sensor to be reliably transported to sink with minimal loss of packet. Reliable Multi-Segment Transport is considered as the main optimal since unlike other protocols does not consider energy efficiency. This protocol is designed for relatively long-lived data flows from source nodes to a sink node. This also uses NACK based hop-by-hop loss detection repair mechanisms by using timer-driven [12].

As RMST is highly improves the delivery of large blocks of data in multiple segments from a source node (cluster members) to a sink node, which is preferred for the proposed protocol. In RMST, uses cache mode and non-cache mode for detecting a missing segment in such a way that, all intermediate node and sink node maintain a cache that store all segments being sent. The authors in [52] present the overall process for reliable data transfer and congestion control at transport layer. Specifically, the authors in [12] are obviously present how the RMST protocol works.

However, RMST protocol result in wasteful use of network resource like memory, CPU, wastage of energy, delay and overhead. Because in order to recover lost packets, intermediate nodes (Tire1 and Tire 2) or sink node has to cache all the incoming packets, also lack congestion control and increases buffer size when sudden happening of events. it does not also consider single /last packet or all packets in the communication lost. The approach in this thesis is design to handle last/single packet and loss of all packets in the communication delivery problem by using hybrid method, which is EACK, and NACK method used to detect and repair packet loss for regular data packet. These methods are clearly described in section 3.4.2.

In addition, these theses also solve buffer overflow when in some cases events suddenly happen after a long quiet time, a large amount of data can be generated from Tire 0 cluster members and fill all the buffer space on intermediate nodes (Tier 1 and Tire 2) and also in sink nodes within a short period. To deal with the shortcomings of RMST, new queue management approach is introduced for reliable data transport protocol; the detail approach is described in section 3.6.

3.4.1. Hop-by-Hop Error Recovery

As described in section 2.5, there are different protocols that use Hop-by-hop error recovery mechanisms such as RMST [12], RBC [13], PSFQ [16], PALER [17], and GAURDA [18]. Hop-by-hop error recovery mechanisms have become a widely accepted recovery mechanism in sensor networks. The basic design idea of hop-by-hop error recovery is that the intermediate nodes, rather than just the final node, perform loss detection and recovery.

To be specific, the whole multi-hop forwarding operation is divided into a series of single-hop processes. By ensuring reliable transmission between every two neighbor nodes in the transmission path, overall reliability can be achieved.

The biggest advantages of hop-by-hop loss recovery can occur quickly, and progress made in early hops. The short out coming of this mechanism is clearly described under [16].

3.4.2. Overview of Automatic Repeat Request Mechanism (ARQ)

Automatic repeat request mechanisms (ARQ) are reliability technique used in hop-to-hop and end-end retransmission based recovery mechanisms. ARQ mechanisms in wireless sensor networks make use of three different acknowledgement mechanisms to trigger the retransmission of a lost packet [41]:

- ✓ **Explicit acknowledgments:** In mechanism, the receiver with a short notification message directly acknowledges every packet that have successfully received data packet. If a sender is not able to recognize the expected notification message, then the packet is retransmitted. This mechanism offers the highest reliability guarantee. It is used by protocols providing packet reliability on hop-to-hop as well as on end-to-end level.
- ✓ **Negative acknowledgments:** Negative acknowledgment mechanisms use sequence numbers to detect packet loss. A node detects the failed transmission of the packet with sequence number n after having received the subsequent packet with sequence number n+1. Now, this node sends a negative acknowledgment to the sender to indicate the loss of packet n. During periods with low packet loss, this mechanism requires a lower amount of individual notification messages than explicit acknowledgments. Negative acknowledgments are used by protocols providing packet reliability on a hop-to-hop as well as an end-to-end level.

- ✓ **Implicit acknowledgments:** After transmitting the data packet, the sender overhears the channel to detect the forwarding of the same packet by the next node. This mechanism does not require any additional notification messages. Implicit acknowledgments show some drawbacks in wireless sensor networks. For example, overhearing of the forwarded packet requires additional energy and overhearing does not work on last hop. This mechanism is only used by protocols providing hop-to-hop reliability.

3.4.3. Brief description of ACK and NACK

ACK is a reliability technique used in TCP/IP as well as wireless sensor network in which control packet is sent by the receiver, if it has successfully received the data packet from the sender. Normally, ACK-based loss recovery schemes are timer-driven. That is, if the sender does not receive the ACK from the receiver within a predefined period, the sender will consider the data packet to be lost during the transmission and will resend the previous packet.

NACK-based loss recovery schemes work in a different way. If the receiver does not receive the data packet within a given time, it will send back a NACK packet to the sender to request retransmission. ACK-based approaches seem to be more reliable than NACK-based approaches since they verify the transmission of every single packet.

However, ACK-based schemes suffer from two major drawbacks when used in sensor networks. The first problem is that, considering the limited bandwidth and energy of sensor nodes, the overhead of sending an ACK for every data packet may be unacceptable, especially when the size of each data packet is relatively small. The second problem is the well-known ACK implosion problem [16]. That is, when a node is broadcasting data packets in a dense network, the requirement of sending an ACK in response to the receipt of a packet for all the receivers may cause serious channel congestion and packet collisions. NACK-based is more effective than ACK-based and can be a better option for sensor networks because it only generates an extra packet when data loss occurs. However, when designing a NACK based loss recovery scheme, several issues still need to be carefully considered. Similarly as with ACK-based schemes, there is a potential NACK implosion problem. When the network connectivity is poor and the sender is broadcasting too many receivers, the sender can be flooded with NACK packets.

Retransmissions may lead to more serious congestion in the network, while ignoring errors can reduce overall network reliability. Another typical NACK problem is the loss of all data packets. In a NACK-based scheme, the receiver can detect and report packet loss only if it is aware of the incoming packet. Thus, a NACK-based scheme cannot handle the unique case where all packets in a communication are lost.

3.5. Proposed transport protocol approach

In order to achieve the three-design consideration that state in section 3.1 and the problem stated in selected protocol that described in section 1.1. To solve these problems the researcher has proposed an improved protocol, which is named as EERMST (energy-efficient reliable multi-segment transport protocol). Beside this, the novel method that uses hybrid (Explicit Acknowledgement and Negative acknowledgement) lost packet recovery mechanism is introduced. The Explicit Acknowledgment cheeks the last packet in the transmission is successfully delivered or not. If the packet is lost within predetermined period the sender retransmits the lost packet. These mechanisms to handle single or last packet lost problem and to solve all packet in the communication lost which provides highest reliability guarantee in addition to Negative acknowledgement. In addition, a novel new queue management approach is introduced to solve the accumulated delay at each hop and to limit the congestion due to sudden happening of an event. This queue management approach is by giving higher priority for retransmission packet rather than new incoming data or old data (routed data) in the queue. Thus, it improves the energy consumption, improves network performance, higher reliability guaranty, increase the network lifetime, and reduces congestion.

In Figure 3.3 shows, the interaction of transport protocol with others layer. The flow of process is start from flow number one and finish at flow number six. The process is start from application layer and going down to transport layer. Transport layer is a layer that provides transport protocol that has a main function of reliable data transmission to ensure data will successfully receive at sink node. Next is network layer that provide route selection. With the best route selection, the data will be sending from source to destination. However, there may have a congestion during data transmission and transport protocol will play the role to solve this problem occur.

Thus, transport layer will interact with the MAC layer and do a process of congestion detection, congestion notification, and rate adjustment. After adjust the congestion rate, the data will go back to transport layer to ensure hop-to-hop reliability guarantee and achieve high reliability.

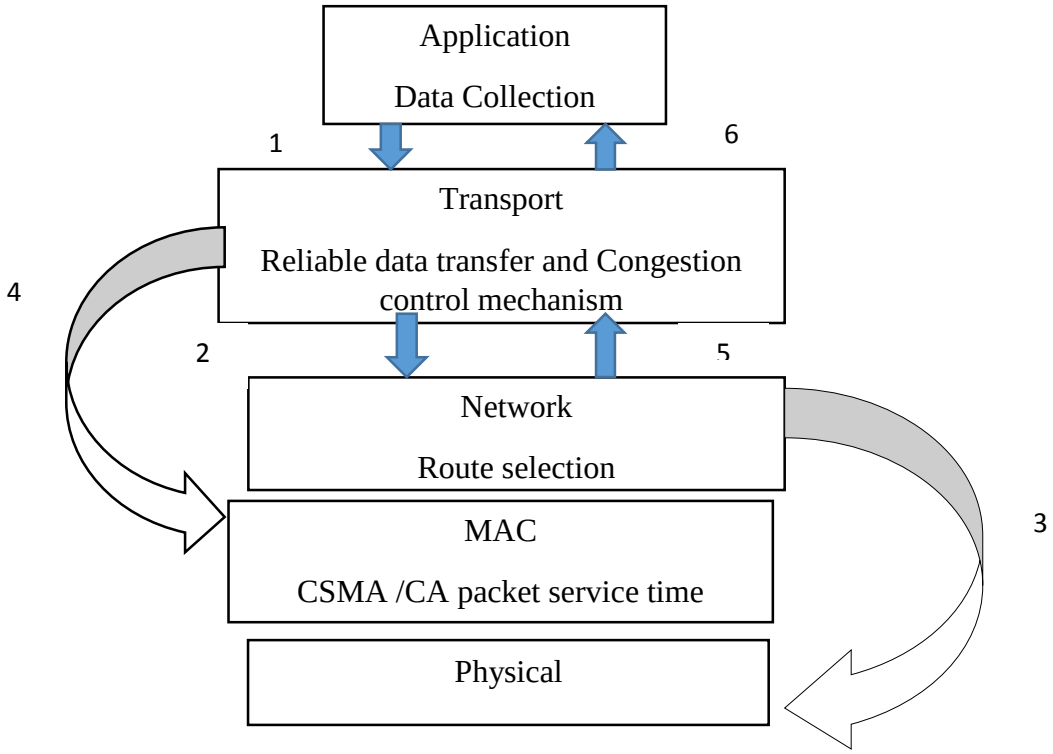


Figure 3.2.: The referenced reliable transport protocol architecture [52]

The state diagram as illustrated in Figure 3.4, describes how proposed hop-hop reliable data delivery protocol works and the following steps shows how protocol proceeds.

1. Sense and send physical phenomenon
2. For new packets set EACK=NACK=0
3. Intermediate nodes stores packet and process data aggregation
4. On successful reception, sink (intermediate nodes) sends EACK=1, NACK=0 towards in reverse direction to sender and the sender deletes packet from its buffer
5. On unsuccessful reception, sink (intermediate) sends EACK=0, NACK=1 towards in reverse direction to sender with lost packet_id, sequence number. The sender resends the lost packet from its buffer.

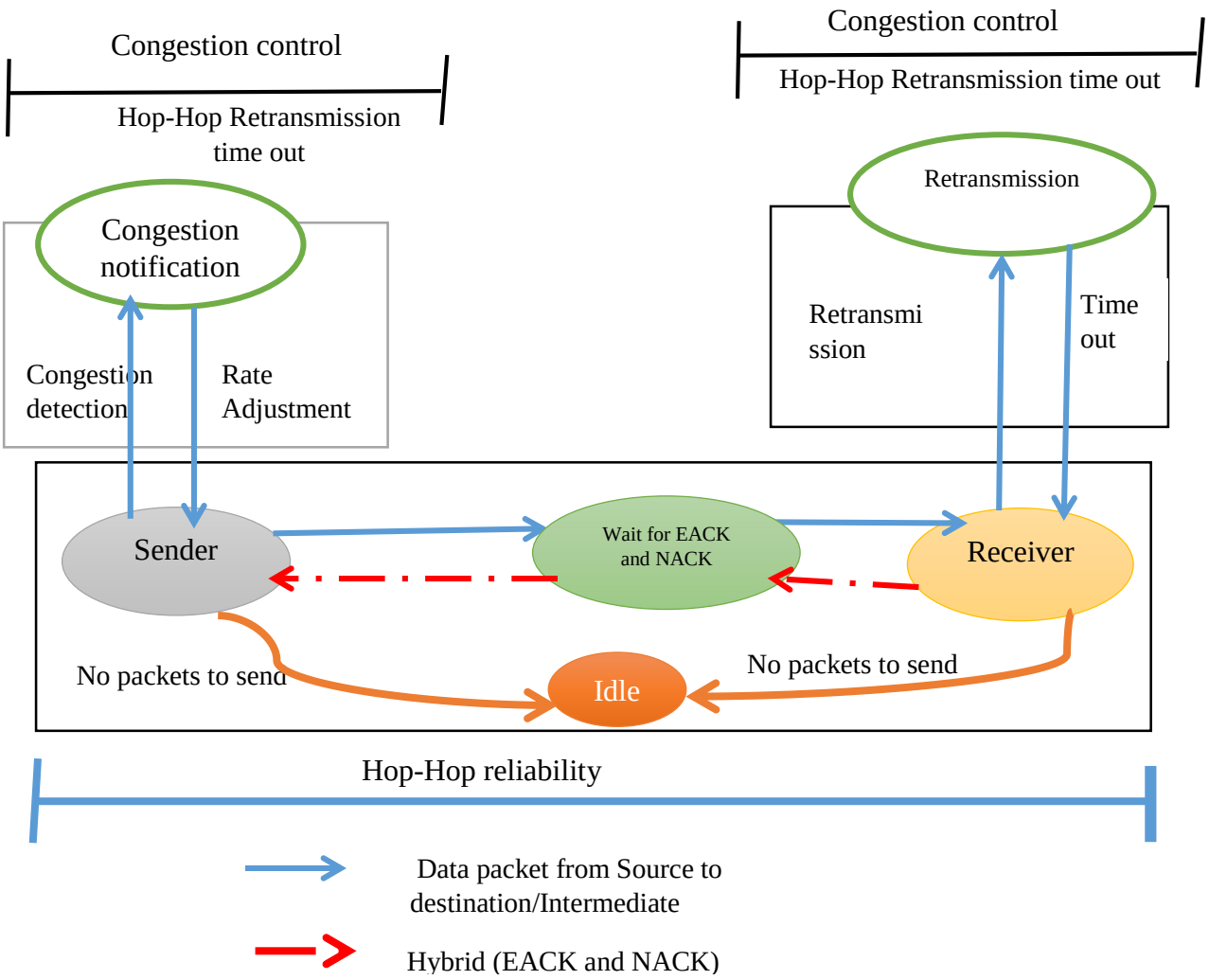


Figure 3.3.: State diagram for proposed protocol

Congestion control

Congestion can occur in wireless sensor networks due to several reasons: interference between concurrent data transmissions, the addition or removal of sensor nodes in the network, or bursts of messages because of the occurrence of some events [38][39]. Congestion in the network can lead to two serious outcomes. As congestion spreads, buffer drops will increase quickly and become the dominant reason for packet loss. Significant delay can also be observed when congestion occurs. Another consequence of congestion is the growing expenditure of resources per packet. Fewer packets can be transmitted with the same amount of energy as before. Thus, alleviating congestion can be helpful in achieving reliable data delivery.

In order to alleviate and handle congestion control different options are used option 1: Dropping packets, Option 2: Control sending rate of individual node, Option 3: Control how many nodes are sending, Option 4: Aggregation. Thus, the author in this study introduces a queue management approach and uses the best data aggregation protocol. Likewise, the use of this queue theory is for handling congestion by dropping semantically less important packet. The mechanism to drop the packet has clearly stated in section 3.6.

Congestion Detection

Used to detect congestion in sensor networks based on the observation that congestion can result in excessive queuing and based on the number of times the channel is sensed busy, a utilization factor can be calculated to deduce the congestion level of the network.

Congestion Notification

When network congestion is detected, the congestion notification information needs to be conveyed from the congested nodes to their neighbors or to the source nodes or destination nodes.

Rate Adjustment:

A straightforward way of alleviating congestion is to simply stop sending packets into the network, or to send at a lower rate. The rate adjustment decision can be made by the congested nodes themselves, by a node outside the congested area (sink node), or by a predetermined policy. When a single CN bit is used to notify congestion, one option is for nodes to adjust their sending rate according to an additive increase multiplicative decrease (AIMD) scheme.

Sender Operation

The sender (sensor node) sends any sensed packet to receiver, before sending the packet, it first examines the existence of retransmission packet in the queue. If there is retransmission packet, sender chooses the packet into higher priority in order to retransmission first. If the packet is being transmitted for the first time, and the sender knows that it will be sending new packet to receiver until end of the last end packet and sends EACK for last packet and wait for EACK and NACK. In the latter case, the sender will not send any new data packet until it gets an EACK back for last packet. If NACK value=1, it resends the packet.

Receiver operation

When a packet is received from the sender, the receiver accepts the packet and stores the packet in to his or her own cache. if the packet is successfully received, it acknowledges NACK=0 and EACK=1 value to back ward direction, on other hand, if there is one or more missed packet, it acknowledges NACK=1 and EACK=0 with missed packet_id and sequence number. The packets that have acknowledge EACK=0 were dropped from the sender's queue.

Algorithm 2, Modified algorithm to improve reliable transfer of multiple messages by using hybrid EACK/NACK based loss detection and recovery.

Notation:

BS *=base station;* ***Seq_no****= Sequence number =1;* ***Buf****=Buffer ;****P****=Parent;* ***C****=Children ;****Node_Id****=Node identification ;****EACK****=Implicit acknowledgement ;****NACK****=Negative Acknowledgement; Initially EACK=NACK=0;*

Up on sensing event /physical phenomena **E**:

Begin

Create data message **M**= (seq_no, node_id, packet_id, Data)

Add a copy **M** to message buffer

Do send message to parent node

While packet_id is not equal to last data packet_id

Sequence ++;

Send message **M**;

Packet_id ++;

End while

Send EACK for last data packet_id;

End Do

Up on receiving message:

IF M is successfully received;

Send (**NACK**=0) and (**EACK**=1) to reverse direction

Add a copy **M** to its own buffer

IF Node_Id is not equal to BASESTATION/Sink

Call ParentSelection ( ) to find a new parent Until Node_Id equal to base station;

Send M to next parent

Send EACK for last data packet_id;

End both **IF**

Else

Look for copy of message in its buffer;

IF message is not found in the buffer;

Sends (**EACK**=0 and **NACK**=1) towards reverse direction with seq_no and missed data packet_id;

Call LookForMesgInchild ( ) //to find message **M** in childes buffer

End If

End Else

Procedure ParentSelection ( )

IF the current node_id is **BS** end parent selection

Else select next parent

Loop

End IF

Procedure LookForMesgInchild ( )

IF data packet_id **AND** seq_no found in its buffer

Resend the data packet

Else, select next Child **until** found missed data packet

End IF

3.6. Proposed queue Management approach

From the section 2.5.2 of related work of literature review, it is assumed that several research has been done in recommending queue management in wireless sensor network for resource constraint nodes using various approach. To en-queue or de-queue the message based on some priority based mechanism. The goal of the presented work is to reliably deliver and transmit sensed data packets as quickly as possible from source to destination. A novel queue management approaches are designed for better scheduling retransmissions and it acts as congestion control mechanism. The scenario considered in this study is that the originating (sensed data); routed data (which is placed in the queue) are equal importance except retransmitted packet in each hope which has given higher priority.

En-queue Policies:

The individual node has to maintain a transmission queue structure, whose responsibility is to temporarily store packets and manage the transmission. For every new data packet includes EACK /NACK bit and the sender set value to Zero. Then, the sender caches the entire packet, sends the packet to the receiver, and starts a retransmission timer. At the same time, the sender can retransmit any missing packet in parallel with the regular data packet transmitting process. When a source node receives a NACK with value one for a given packet, it locates the missing packet in its transmission queue if present and adds into header of the queue to transmit first otherwise go reverse direction until the missed packet found.

De-queue Policy:

In this section, describes how packet can be removed from the queue. Since any event that can be sensed by sensor nodes can receive and generate data packets constantly, while their storage capacity is quite limited due to size, cost and power limitations, an appropriate de-queue policy is necessary in order to manage the buffer space more efficiently. In the proposed policy, packets are de-queued in the following scenarios:

- If the transmission queue not reaches its maximum capacity and a new packet is received, the node adds the newer packet at the tail of the queue. Since all packets in the queue move in the sequence from tail to head.

- The ACK or NACK feedback packet has higher priority; received ACK/NACK Packet ID has read and compared with the IDs of packets in the transmission queue. Since the Packet_ID of ACK or NACK gives the latest in-order packet received by the node, which is responsible to retransmit packet first. The packet X with matching Packet_ID in previous hop has to retransmit the missed packet until received by receiver. The node has to drop (de-queue) all packets except the packet that needs retransmission.

Algorithm 3, De-queue policy

Notation:

*Let **Q** be the output queue with capacity **k;** Let **p'** be the new packet to insert; Let **Pi** be the existing packets in **Q***

Begin

IF Q size is less than **K** then

Insert **p′** to the tail of queue

Else

IF P′ data type is feedback (NACK=1) **Then**

For each packet **Pi** in queue **Q Do**

Replace Feed Back (NACK) with **Pi**

Move replaced one to head of the **Q**

Send to next hope/Sink

End For

End If

IF data type is feedback (ACK=0) **Then**

For each packet **Pi** in queue **Q do**

Drop the packet

End for

End If

IF P' data types routed (data in the **Q**) **Then**

For each **Pi** in the Queue **do**

Insert to head of the Queue

Send to next hop (sink)

End for

End if

IF P' data types originating (sensed fresh data) **Then**

For each **Pi** in the Queue **do**

Insert to **Tail** of the Queue

End for;

End if;

End If

3.7. Data aggregation in Hierarchical based networks

In energy constrained sensor networks of large size, it is inefficient for sensors to transmit the data directly to the sink. In such scenarios, sensors can transmit data to a local aggregator or cluster head which aggregates data from all the sensors in its cluster and transmits the concise digest to the sink. This results in significant energy savings for the energy constrained sensors. Thus, the scenario that described in section 3.2 and the architecture presented in section 3.3 shows cluster based network, in which different sensors nodes (cluster members) are grouped in to different cluster heads. The cluster heads can communicate with the sink directly via long range transmissions or multi hopping through other cluster heads. In our study, the sensor nodes are deployed in predetermined region in which event detection applications applied. For future it is beneficial to use multi-hop communication among the nodes in the cluster members to reach the cluster head. We have recommended to use the most popular and attractive hierarchical protocol of LEACH [48] for data aggregation and clustering technique. In [53] also describes the modified LEACH approach in which clustering process, selection of cluster head process, and parent selection process takes place. Clustering is a process of dividing sensor nodes into groups on the basis of various parameters and selecting a group leader from each group. The groups are called clusters and group leaders are called cluster heads(CHs) of the cluster. The parameter for forming the cluster includes distance between cluster head and its member, intra cluster communication cost, residual energy of sensor nodes, location of nodes with respect to base station etc.

In LEACH protocol the distributed sensor nodes organize themselves into clusters for data fusion. A designated node (cluster head) in each cluster transmits the fused data from several sensors in its cluster to the sink. This reduces the amount of information that is transmitted to the sink. The data fusion is performed periodically at the cluster heads.

LEACH is suited for applications, which involve constant monitoring and periodic data reporting.

There are two main phases involved in LEACH, setup phase, and steady state phase. The setup phase involves the organization of the network into clusters and the selection of cluster heads.

The steady state phase involves data aggregation at the cluster heads and data transmission to the sink. In the set up phase, the clusters are organized and cluster heads selected based on the suggested percentage cluster head in the network and the number of times the node has been a cluster-head so far in each sensor category. This decision is made by each node n choosing a random number between 0 and 1. If the number is less than a threshold T (n), the node becomes a cluster-head for the current round. The threshold T (n) is set as follows:

$$T(n) = \begin{cases} \dfrac{p}{1 - p * (r \bmod \frac{1}{p})} & if \quad n \in G \\ 0 \end{cases}$$

Where **p** is the desired cluster-head percentage in each sensor category which is given by the user, **r** is the current round and **G** is the set of nodes in each sensor category that have not been cluster-heads in the last $1/P$ rounds.

As a general LEACH is the most popular clustering algorithm and can be applicable in network layer for routing protocol in wireless sensor network. Therefore, this protocol is behind our scope.

Chapter Four:

4. Prototype Implementation and performance Evaluation

4.1. Experimental Environment Setup

In our experiment the proposed protocol is implemented in the network embedded systems C (nesC) programming language and the TinyOS operating system [21]. nesC is a component based event-driven programming language based on the C programming language. TinyOS is an open source component-based operating environment written in nesC and is optimized and designed for embedded systems such as wireless sensor networks.

TOSSIM is a discrete event simulator for TinyOS sensor networks and it is a short form of TinyOS simulator. In TinyOS, applications are being compiled for a mote. By using TOSSIM, the application can be compiled into its framework virtually on a PC which is allowing the user to test and analyze algorithms in a controllable environment.

MicaZ wireless sensor motes are used for our experiments. Each MicaZ mote has an ATMEL 7.37 MHz ATMega128L, low power 8-bit micro-controller with 128 KB of program memory, 512 KB measurement serial flash data memory, and 4 KB EEPROM. The MicaZ mote uses Chipcon CC2420 radio, a single-chip IEEE 802.15.4 compliant radio frequency transceiver operating at 2.4 GHz, and is capable of transmitting at 250 kbps.

4.2. Experimental Design

In order to demonstrate the impact of different parameters to the performance of the proposed approach, all of the experiments in this study are conducted by varying network topology ,internal noise level as well as external. In experimental design, the protocol is runs on a network configured on notepad application with 5, 10, 15 number of sensor nodes and one sink node that is shown in Figure 4.1. Each of the sensor nodes is programmed to create data packets and send them as well receive packet from its upstream neighbor, to its downstream neighbor. Each sensor node creates a new data packet every 1000 ms. The detail of simulation parameters for the simulation scenario is shown in Table 4.1..

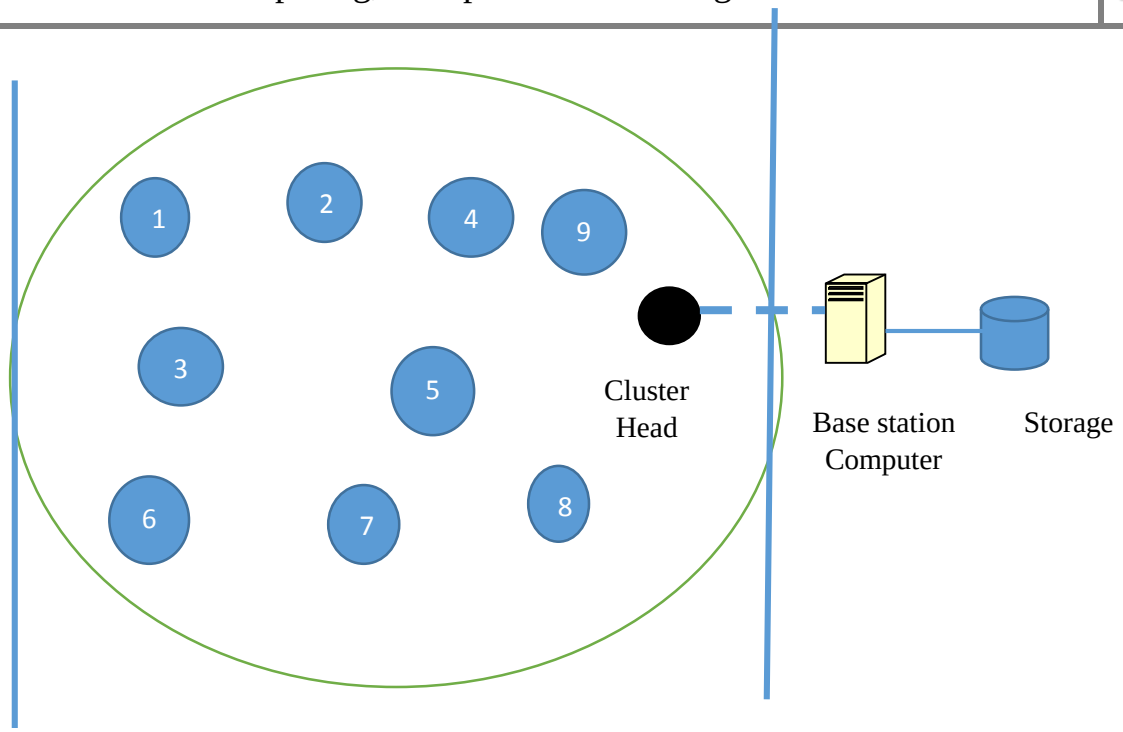


Figure 4.1.: Experimental design

For better understand of this experimental design handshaking is involved between sink node and nodes.

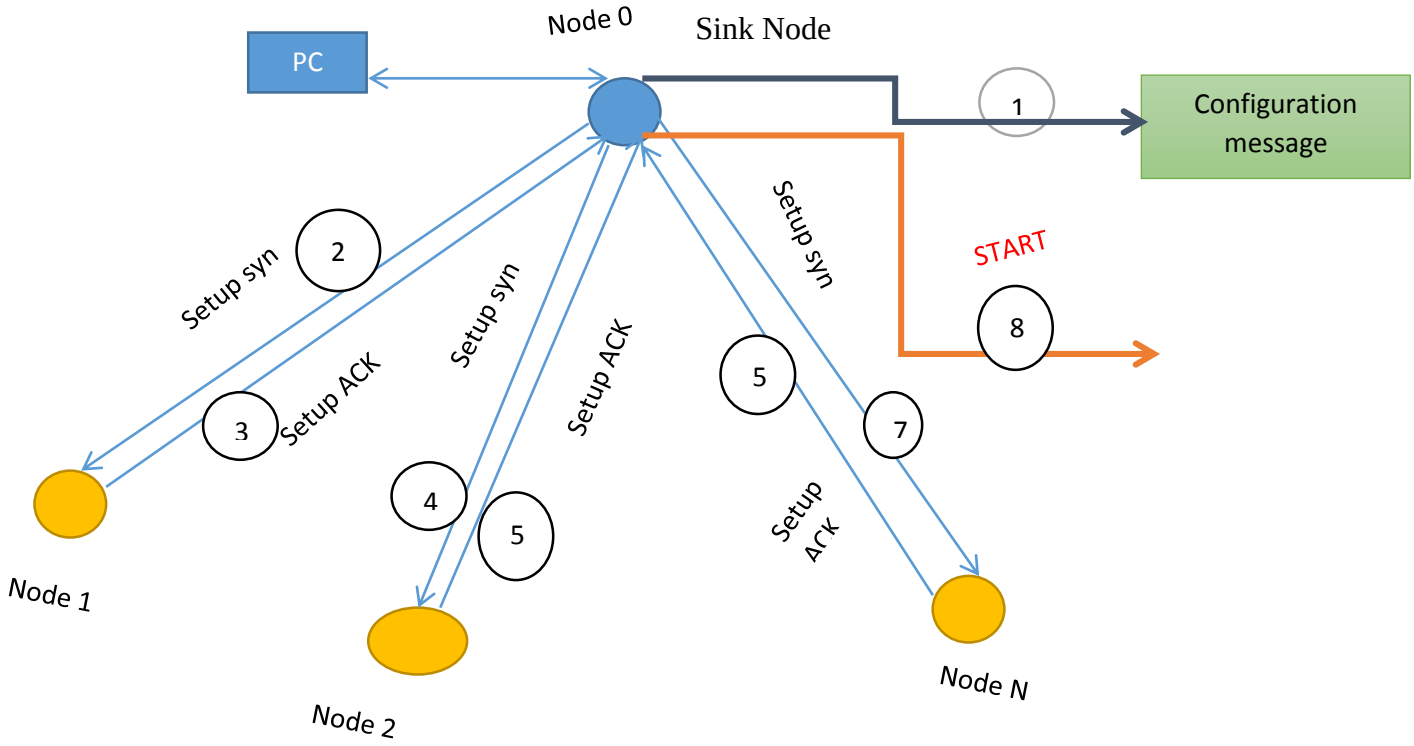


Figure 4.2: Configuration stage

First, the configuration message is emitted in a broadcast way, so that each mote within the Base Station reception range receives it. Then, the PC application iterates through all motes and requests setup synchronization, to which motes should answer with a setup acknowledgement. Until this acknowledgement is not received, the synchronization message is resent. After a successful configuration, a 'start' message is injected into the network in a broadcast way which signals the start of the Data monitoring section.

Simulation Parameters	Values of Simulation Parameter
Number of Sensor Nodes	5,10,15
Number of Sink Nodes	1
Sending Interval	1000 ms
Message Size	2 Byte (default)
Data rate	250 kbps
Active Message Type	6
Simulation runs	TinyOS 2.1.x

Table 4.1.: Summary of experimental parameters

For experiments conducted in this thesis, there are 5, 10 MicaZ motes used and all sensor nodes are deployed in predetermined environment in health care. This designed topology is saved in a separate text file. To get this file configuration of topology in the simulation and the radio connectivity graph can be scripted by python language by specifying a network topology file. The application is created on notepad application and the created python script can loaded in to the simulation.

The created topology is saved as topology.txt, topology1.txt file in the same folder in TOSSIM application, and the script will read such file. These are 8 lines of the topology.txt, which is shown in Figure 4.3. The node (mote) Zero which is sink node communicated with other motes one, two, etc and with a gain value 30 and 40 that means the signal strength between two nodes.

0	1	-30.0
0	2	-30.0
0	3	-30.0
0	4	-30.0
1	0	-30.0
2	0	-30.0
3	0	-30.0
4	0	-30.0

0	1	-40.0
0	2	-40.0
0	3	-40.0
0	4	-40.0
1	0	-40.0
2	0	-40.0
3	0	-40.0
4	0	-40.0

Figure 4.3.: Network topology

In this case, the units of node with single hop and one sink created in TOSSIM. As shown the topology in Figure 4.2 the value having 0 (zero) is sink node. When TOSSIM is started, no node can communicate with each other and the default radio model is based on signal-strength. TOSSIM simulates RF noise and interference a node hears, both from others and outside sources which uses Closest Pattern Matching (CPM) algorithm. To configure CPM, a noise trace needed to be feed. This can be achieved by calling addNoiseTraceReading on a Mote object. The Author assumed, the motes are deployed in a very harsh environment or in a very noisy place and recommended to use the meyer-heavy.txt, which is the very noise trace file having 196604 lines. For a sample, the following Figure 4.4 shows 45 lines of noise file.

-39	-98	-97	-96	-98
-98	-98	-98	-98	-86
-98	-99	-97	-98	-90
-98	-98	-98	-98	-91
-99	-98	-98	-98	-87
-98	-98	-98	-98	-87
-94	-98	-98	-99	-98
-98	-98	-91	-86	-98
-98	-98	-98	-97	-98

Figure 4.4: Sample noise text file

The following piece of codes is written down to give a node a noise model from a node trace file. The script code reads the whole 196604 lines. Likewise, this value indicated that the node have a huge amount of external noise or wave that affects the reliability to deliver their data.

```
noise = open(“meyer-heavy.txt”, “r”)
lines = noise.readlines()
for line in lines:
str = line.strip()
if(str != “ ”):
val = int(str)
for i in range(1, num_node+1):
t.getNode(i).addNoiseTraceReading(val)
```

4.3. Simulation process

After the environment setting, setup set and defined as described in the section 4.1 and 4.2, conducting the simulation experiment using Virtual machine, which run Ubuntu 12.04 operating system and installing wireless sensor network application. The Figure 4.5 shows the simulation process of TOSSIM simulator.

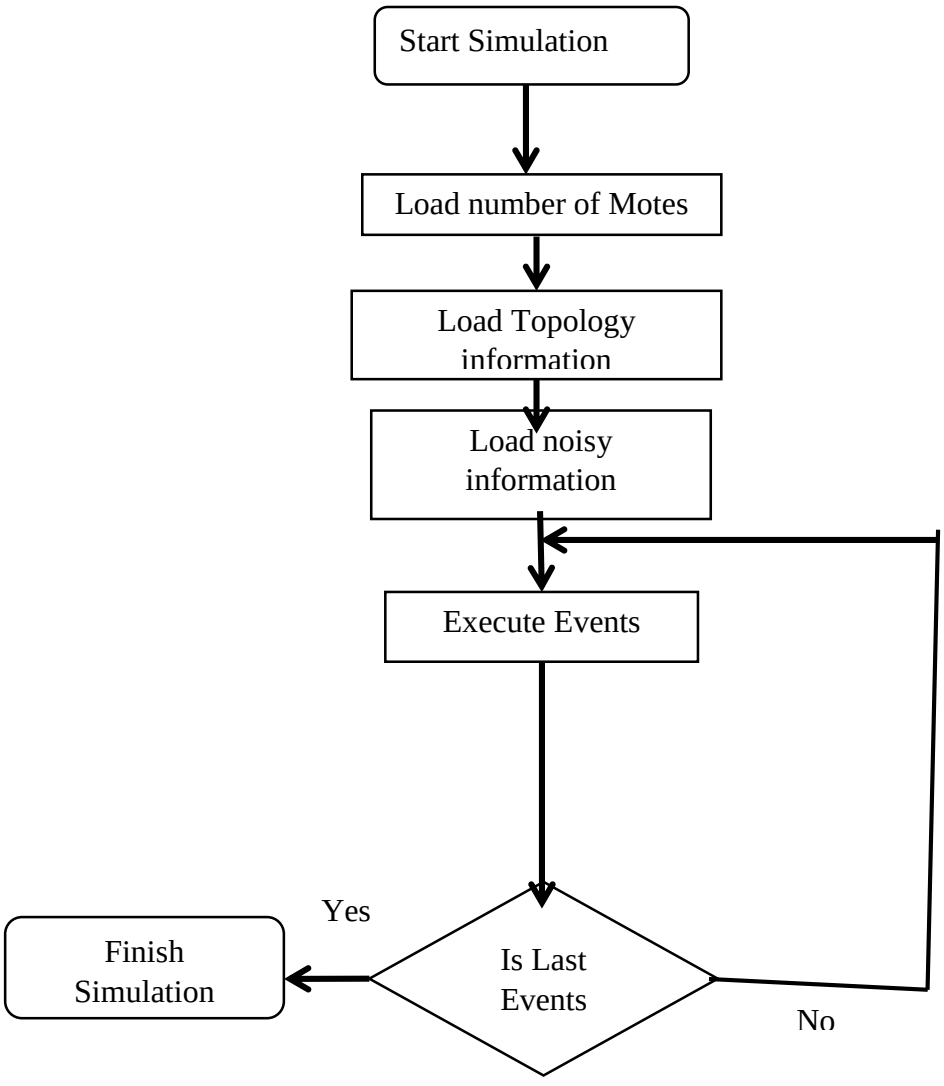


Figure 4.5. : Simulation Process Flow

4.4. Reporting and Visualization:

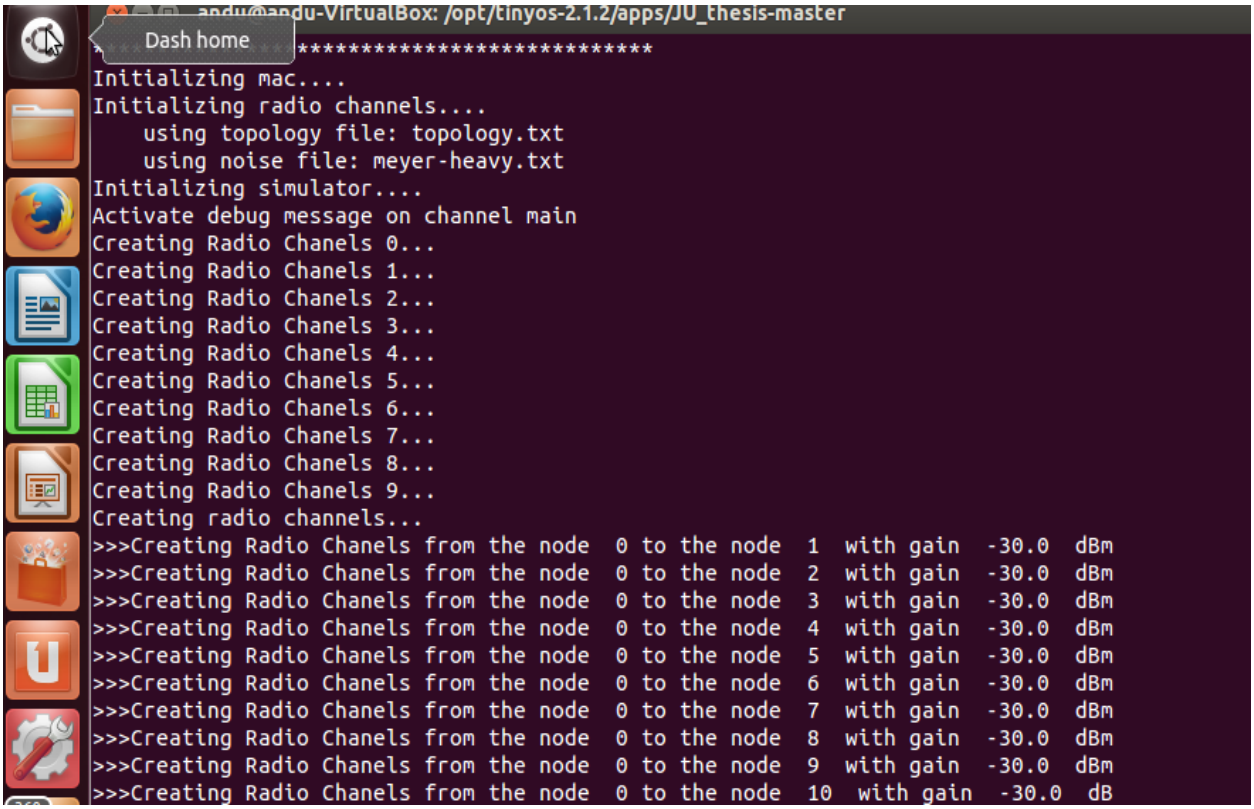
In this section, we describe the way in which the results of the simulation were visualized. In TinyOS 2.1.x, it only supports CLI (command line interface) and simulation result is visualized by command line interface. At this time, the implementation of proposed EERMST protocol is implemented on system application, which would sense temperature and developed health care monitoring application for monitoring the heart activities of the patient in hospital. As shown in the Figure 4.1 shows that nine sensor nodes are deployed for people who are positioning at intensive care room. At this room, ECG signals of every patient are measured. Notice that ECG (Electrocardiogram) is the process of recording the electrical activity of the heart over a period. The measured data are reliably transmitted to nursing room for remote monitoring. In addition, it connected to any networked device and database for further manipulation of data. There are 200 samples of heartbeat data are used for each node and determined the rate of heartbeat. The determined heartbeat may be at high rate, normal rate and low rate based on estimated target heart rates for different ages [54]. The application detects the movement of heartbeat and provides feedback to help and maintain an optimal health status which is crucial for further treatment. The role of EERMST has implemented at the time of transportation of data over network. From the simulation, each Mote communicated with each other based on topology and has a set of radio range between them. If the radio strength between the motes is stronger than noise, data is delivered from source to destination. Otherwise, the packet is going to be lost or takes more time to deliver. In addition, in real environment there are many noises caused by interference or multipath effect. For example, waves can be reflected in building can be a noise have been included in the simulation result. From the result, all motes are powered on at the same time and the sink node enforces to start data monitoring session at the same time. There are 200 samples of data given for monitoring section for normal nodes (motes). The Figure 4.6 illustrates the sample of data and finally the motes respond the monitored data to sink node. The Figure 4.7 displays simulation result.

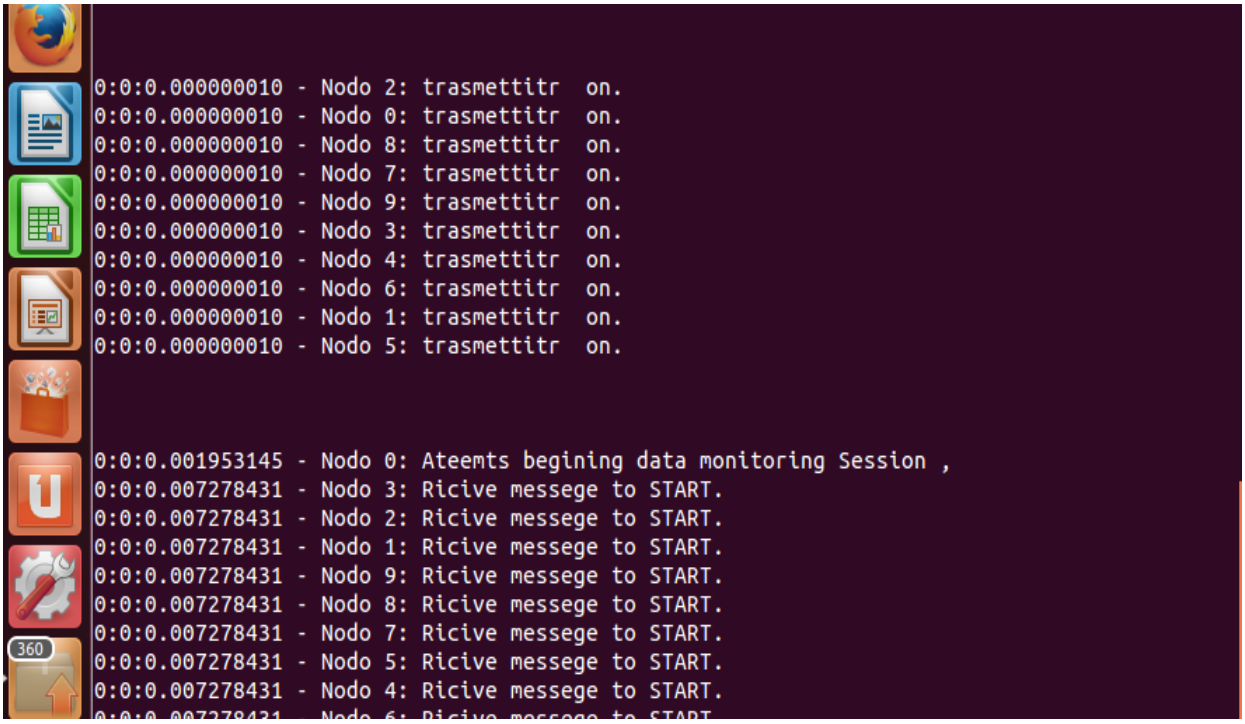
```
#define SAMPLE_DATA_SIZE 2400

uint16_t DATA1[SAMPLE_DATA_SIZE]= {352, 202, 480, 565, 209, 857, 284, 225, 345, 115,
885, 989, 475, 41, 468, 134, 81, 976, 923, 929, 440, 791, 470, 68, 320, 678, 476, 199, 193, 516,
582, 752, 700, 531, 1930, 943, 213, 495, 898, 122, 67, 158, 209, 205, 567, 20, 108, 411, 716, 727,
566, 51, 796, 730, 4, 1004, 504, 243, 148, 732, 714, 21, 362, 721, 12, 56, 978, 486, 696, 540, 403,
90, 126, 82, 501, 919, 932, 320, 116, 128, 930, 191, 130, 40, 527, 457, 547, 160, 141, 764, 752,
69, 473, 1694, 437, 123, 54, 1267, 288, 176, 551, 582, 66, 357, 199, 393, 321, 485, 122, 127, 401,
550, 88, 703, 630, 81, 942, 285, 130, 792, 906, 541, 904, 693, 307, 179, 720, 182, 460, 345, 232,
540, 34, 333, 185, 22, 874, 739, 229, 196, 15, 172, 1147, 187, 545, 642, 758, 350, 12, 632, 341,
13, 23, 523, 668, 87, 468, 886, 147, 386, 193, 556, 734, 655, 85, 202, 824, 567, 116, 519, 873,
1527, 950, 219, 256, 296, 85, 120, 312, 565, 892, 298, 556, 219, 362, 681, 732, 826, 433, 44, 128,
143, 199, 15, 34, 541, 1001, 306, 137, 410, 120, 675, 622, 201, 89, 338, 171, 617, 106, 96, 349,
1102, 574, 317, 467, 770, 126, 180, 1003, 163, 697, 51, 809, 150, 987, 291, 102, 181, 171, 291,
257, 820, 621, 486, 333, 210, 8, 985, 300, 814, 699, 612, 230, 19, 120, 1026, 55, 1051, 64, 27,
481, 20, 152, 279, 244, 155, 1326, 480, 1461, 221, 655, 91, 387, 182, 277, 95, 309, 795, 43, 428,
214, 710, 659, 427, 168, 619, 866, 606, 0, 36, 597, 390, 230, 819, 609, 157, 452, 427, 781, 1056,
16, 1277, 669, 565, 1, 46, 1629, 381, 1437, 1520, 52, 622, 270, 444, 562, 453, 294, 66, 541, 351,
588, 86, 96, 660, 1392, 331, 833, 251, 425, 541, 665, 309, 90, 191, 198, 262, 609, 116, 1397, 751,
412, 789, 166, 936, 14, 367, 1427, 746, 1636, 289, 916, 167, 108, 490, 179, 1033, 315, 215, 436,
56, 577, 212, 514, 1183, 1218, 396, 68, 369, 74, 593, 307, 82, 969, 564, 514, 202, 153, 693, 186,
57, 963, 126, 539, 62, 220, 593, 189, 229, 103, 679, 589, 629, 207, 281, 679, 1231, 616, 516, 574,
210, 366, 204, 374, 672, 596, 138, 1113, 398, 77, 458, 157, 191, 907, 65, 82, 48, 24, 58, 861, 218,
429, 830, 314, 554, 62, 109, 58, 207, 452, 26, 447, 974, 291, 273, 198, 508, 457, 399, 857, 309,
872, 669, 38, 38, 182, 330, 170, 47, 165, 808, 67, 767, 728, 122, 253, 67, 103, 65, 783, 622, 145,
357, 212, 236, 39, 456, 483, 433, 347, 536, 281, 39, 118, 800, 157, 174, 768, 334};
```

Figure 4.6: Sample data

Note: 200 samples of data are used


andu@andu-VirtualBox: /opt/tinyos-2.1.2/apps/JU_thesis-master
Dash home
Initializing mac....
Initializing radio channels....
using topology file: topology.txt
using noise file: meyer-heavy.txt
Initializing simulator....
Activate debug message on channel main
Creating Radio Chanels 0...
Creating Radio Chanels 1...
Creating Radio Chanels 2...
Creating Radio Chanels 3...
Creating Radio Chanels 4...
Creating Radio Chanels 5...
Creating Radio Chanels 6...
Creating Radio Chanels 7...
Creating Radio Chanels 8...
Creating Radio Chanels 9...
Creating radio channels...
>>>Creating Radio Chanels from the node 0 to the node 1 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 2 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 3 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 4 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 5 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 6 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 7 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 8 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 9 with gain -30.0 dBm
>>>Creating Radio Chanels from the node 0 to the node 10 with gain -30.0 dB


0:0:0.000000010 - Nodo 2: trasmettitr on.
0:0:0.000000010 - Nodo 0: trasmettitr on.
0:0:0.000000010 - Nodo 8: trasmettitr on.
0:0:0.000000010 - Nodo 7: trasmettitr on.
0:0:0.000000010 - Nodo 9: trasmettitr on.
0:0:0.000000010 - Nodo 3: trasmettitr on.
0:0:0.000000010 - Nodo 4: trasmettitr on.
0:0:0.000000010 - Nodo 6: trasmettitr on.
0:0:0.000000010 - Nodo 1: trasmettitr on.
0:0:0.000000010 - Nodo 5: trasmettitr on.
0:0:0.001953145 - Nodo 0: Ateemts begining data monitoring Session ,
0:0:0.007278431 - Nodo 3: Ricive messege to START.
0:0:0.007278431 - Nodo 2: Ricive messege to START.
0:0:0.007278431 - Nodo 1: Ricive messege to START.
0:0:0.007278431 - Nodo 9: Ricive messege to START.
0:0:0.007278431 - Nodo 8: Ricive messege to START.
0:0:0.007278431 - Nodo 7: Ricive messege to START.
0:0:0.007278431 - Nodo 5: Ricive messege to START.
0:0:0.007278431 - Nodo 4: Ricive messege to START.

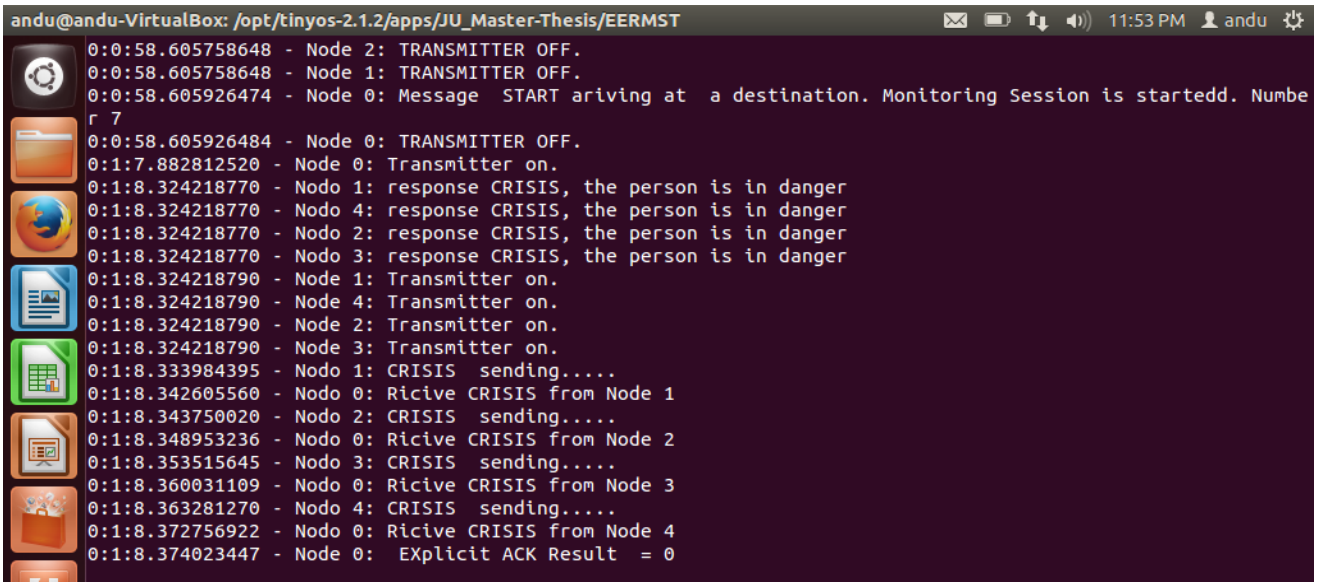


Figure 4.7: The screen shot of TOSSIM simulator (CLI)

The result shows that, from the given 200 samples of data for each sensor node determines the patient's heartbeat rate (crises rate), normal movement rate (normal condition) and low rate (the person recommended to exercise) which delivers their result to sink node (nursing room). The protocol determines reliable delivery of the given data to sink node. In section 4.5 shows the evaluation metrics to analyses the protocol.

4.5. Performance Evaluation Metrics

Our main aim of the proposed protocol is to improve reliability in transport protocol and reduce overhead as well as latency by implementing a hop-by-hop loss recovery and hybrid based loss detection method. In the experiments, the following metrics are considered when analyzing the performance of the proposed protocol:

4.5.1. End-to-End Delay: The end-to-end delay is measured as the interval between the generation of a data packet at its source and the reception of that packet at the sink. The end-to-end delay shows the average amount of time it takes for the network to deliver a data packet from a particular source node to the sink.

$$\text{End-to-End-Delay} = \sum \frac{Packet\_Arival\_Time - Sent\_Packet\_Time}{Total\_number\_of\_Connection\_Pair} \quad \textbf{(1)}$$

4.5.2. Link Delay: The link delay measures the interval from when a packet is created at the sender to the time it is received at the next hop receiver.

4.5.3. Delivery Ratio:

It is defined as the ratio of the number of unique packets successfully received at the sink to the number of packets forwarded by the sources. This illustrates the level of delivered packet to the destination. Equation (2) shows the delivery ratio calculation.

$$\textbf{Delivery Ratio} = \frac{\sum \text{Number of packet received}}{\sum \text{Number of packet send}} \qquad \textbf{(2)}$$

4.5.4. Resend Rate:

As the name indicates, the resend rate is a measure of the frequency of retransmissions by a node. The resend rate for each sensor node is calculated as the number of resent data packets divided by the total number of data packets sent by the node. A higher resend rate indicates that more of the senders' transmissions at the link are unsuccessful. Since retransmitting packets may cause higher waiting time in the transmission queue, the resend rate has significant impact on both the end-to-end delay and the link delay.

4.6. Result and Discussion:

This section presents the detail analysis of the simulation using charts, based on the results generated from simulation. In this experiment, all sensor nodes turned on or started at equal time and have the same transmission off time. The sink node exerts for all other nodes in order to starts data monitoring session. The monitored data from nodes returned to sink node and the sink node cheeks the delivery of data, detects lost packet and perform recovery operations. Hop-by hop detection method and hybrid recovery method was applied for our approach.

End-to-End Delay: This metric is important to know how long time for each packet to reach from its source to destination. The proposed algorithm (EERMST) which is the combination of two loss recovery approaches compared with RMST_NACK based loss detection, and recovery mechanism.

RMST_NACK and EERMST analysis result

Table 4.2.: RMST_NACK and EERMST analysis with 30 gain value with in three rounds

On the first round with gain value of 30 dBm

Node ID	Node on time	Node 0 Broadcast received time	Data monitoring section started by node 0	Sending time	Receive time(RMST)	Receive Time (EERMST)
1	0.000000010	0.009429908	0:0:0.001953145	0:0:9.737304707	0:0:9.740493775	0:0:9.744567831
2	0.000000010	0.009429908		0:0:9.747070332	0:0:9.751419062	0:0:9.747177126
3	0:0:0.000000010	0.009429908		0:0:9.756835957	0:0:9.763412457	0:0:9.757492064
4	0.000000010	0.009429908		0:0:9.766601582	0:0:9.775573698	0:0:9.768508902

On 2nd round with gain value of 30 dBm **analysis result**

Node ID	Node on time	Node 0 Broadcast time started	Sending time	Receive time in (RMST)	Receive Time in (EERMST)
0	0:0:19.057617207	0:0:19.548828145			
1	0:0:19.498046915	0:0:19.550720223	0:0:19.507812520	0:0:19.518646197	0:0:19.511672948
2	0:0:19.498046915	0:0:19.550720223	0:0:19.517578145	0:0:19.52615353	0:0:19.513671895
3	0:0:19.498046915	0:0:19.550720223	0:0:19.527343770	0:0:19.532791125	0:0:19.523437520
4	0:0:19.498046915	0:0:19.550720223	0:0:19.537109395	0:0:19.547607380	0:0:19.540267923

Node ID	Node on time	Node 0 Broadcast time started	Sending time	Receive time (RMST)	Receive Time (EERMST)
0	0:0:19.057617207	0:0:19.548828145			
1	0:0:19.498046915	0:0:19.550720223	0:0:29.27832033	0:0:29.28752133	0:0:29.273590095
2	0:0:19.498046915	0:0:19.550720223	0:0:29.28808595	0:0:29.28984071	0:0:29.285430903
3	0:0:19.498046915	0:0:19.550720223	0:0:29.29785158	0:0:29.30856319	0:0:29.300872765
4	0:0:19.498046915	0:0:19.550720223	0:0:29.30761721	0:0:29.31571958	0:0:29.311019857

On 3rd round with gain value of 30 dBm **analysis result**

EERMST analysis result

Table 4.3.: EERMST analysis with 50 gain value of three rounds

Frist round with gain value of (50 dBm**)**

Node ID	Node on time	Node 0 Broadcast time started	Sending time	Receive time	Link delay
0	0:0:0.000000010	0:0:0.001953145			
1	0:0:0.000000010	0:0:0.003662119	0:0:9.731445332	0:0:9.734786987	0.003341655
2	0:0:0.000000010	0:0:0.003662119	0:0:9.741210957	0:0:9.750396694	0.009185737
3	0:0:0.000000010	0:0:0.003662119	0:0:9.750976582	0:0:9.758270241	0.007293659
4	0:0:0.000000010	0:0:0.003662119	0:0:9.760742207	0:0:9.763153082	0.002410875

Second round with gain value of 50 dBm

Node ID	Node on time	Node 0 Broadcast time started /received	Sending time	Receive time	Link delay
0	0:0:19.042968770				
1	0:0:19.484375040	0:0:0.003662119	0:0:19.494140646	0:0:19.502334566	0.008193926
2	0:0:19.484375040	0:0:0.003662119	0:0:19.503906270	0:0:19.510818461	0.006912191
3	0:0:19.484375040	0:0:0.003662119	0:0:19.513671895	0:0:19.519866927	0.006195032
4	0:0:19.484375040	0:0:0.003662119	0:0:19.523437520	0:0:19.530822730	0.00738521

Third round with gain value of 50 dBm

Node ID	Node on time	Node 0 Broadcast time started	Sending time	Receive time	Link delay
0	0:0:28.816406270				
1	0:0:29.256835977	0:0:29.305740341	0:0:29.266601582	0:0:29.273559549	0.006957967
2	0:0:29.256835977	0:0:29.305740341	0:0:29.276367207	0:0:29.280075071	0.003707864
3	0:0:29.256835977	0:0:29.305740341	0:0:29.286132832	0:0:29.294647187	0.008514355
4	0:0:29.256835977	0:0:29.305740341	0:0:29.295898457	0:0:29.298736575	0.002838118

In the ACK based approach, it is obvious that each data packet requires acknowledgement and the new data packet cannot be sent until the previous sent was acknowledged by receiver, this assures each packet delivery, however it has more delay than other does. In RMST_NACK protocol, only the lost data packet requires acknowledgment. The experiment tests the performance of proposed hybrid based approach (EERMST) and RMST_NACK approach with the same 50 and 100 ms timer in order to manage the expiry of time with the same gain value of -30.0, -40.0 and -50.0 dBm for four nodes. The results show that, with the same 30 and 40 gain value the end-to-end delay of RMST_NACK is closer to 0:0: 0.0057716035 ms.

The End to End delay result of new (EERMST) approach is closer to 0:0: 0.00248333625 ms. The RMST_NACK have more delay with the value 0:0: 0.00328826725 ms.

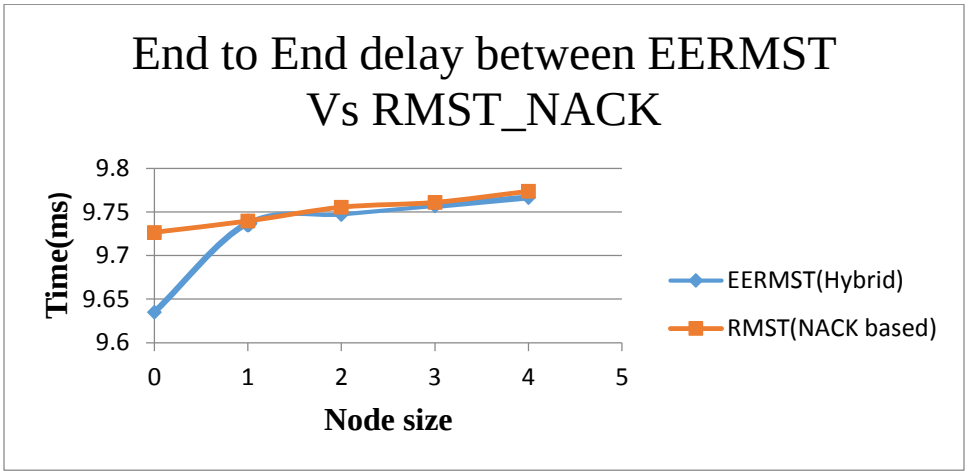


Figure 4.8: End-to-End delay between EERMST Vs RMST_NACK

Resent Rate: The resent rate of EERMST and RMST_NACK is more or less comparable with internal noise of 30, 40 dBm. However, resend rate of RMST with 50 dBm have lost all packets and all packet needs retransmission. In EERMST approach shows that, each sensor node is started monitoring session twelve times roundly. From these rounds, only two nodes required resending the missing packet with 50 dBm and the other nodes delivers their packet effectively.

Therefore, this shows that it performs best outstanding performance in 50 dBm and more. Notice that from these two approaches, the new data packet cannot be sent until the receiver acknowledged previous sent packet with in sending interval.

Link Delay: This metric used to know how long time the packet to reach from source node to next hope. To evaluate the performance of our protocol, the experiment tested on fixed network size with varying the internal noise level having 30, 50, and more. The average delays to deliver a packet from source to next hop have computed and the results that obtained from the experiment clearly shown in Table 4.2 and Table 4.3. In the first three rounds of data monitoring section the result of link delay between hybrids based EERMST and RMST_NACK protocol shows that EERMST is less delay than RMST. In addition, the delay increase as noise level increase. As would be expected, more packet loss will happen when increasing noise and it will extend the delivery delay.

Delivery Ratio: As stated in chapter two, many applications need success especially in critical based events, which depends upon reliable delivery of events. Delivery ratio represents an important metric to estimate reliability. The packet delivery ratios for a network size of 5, 10, and 15 nodes have demonstrated and it shows that similar delivery ratio result except with varying noise level. Each node has sent 200 numbers of data sent and the sink node can received from the sent packet as shown in Figure 4.11. The proposed EERMST achieves good packet delivery ratio in all scenarios with highest noise level than NACK based RMST protocol. They also have a best packet delivery ratio with light noise.

			EERMST			RMST		
		Packet sent	Packet resent	Packet reached at sink node	Packet delivery ratio	Packet reached at sink node	Packet delivery ratio	Packet resent
Node 1	Round 1	200	3	199	0.995	196	0.98	4
	Round 2	200	4	197	0.985	193	0.965	4
	Round 3	200	3	194	0.97	190	0.95	3
Node 2	Round 1	200	1	197	0.985	195	0.975	2
	Round 2	200	2	197	0.985	198	0.99	4
	Round 3	200	1	193	0.965	192	0.96	4
Node 3	Round 1	200	2	196	0.98	196	0.98	3
	Round 2	200	4	198	0.99	189	0.945	1
	Round 3	200	0	195	0.975	199	0.995	2
Node 4	Round 1	200	3	197	0.985	183	0.915	5
	Round 2	200	1	198	0.99	195	0.975	0
	Round 3	200	2	196	0.98	194	0.97	4

Table 4.3: Number of packet sent, reached packet to sink, packet retransmitted, and its Packet Delivery ratio sensor nodes

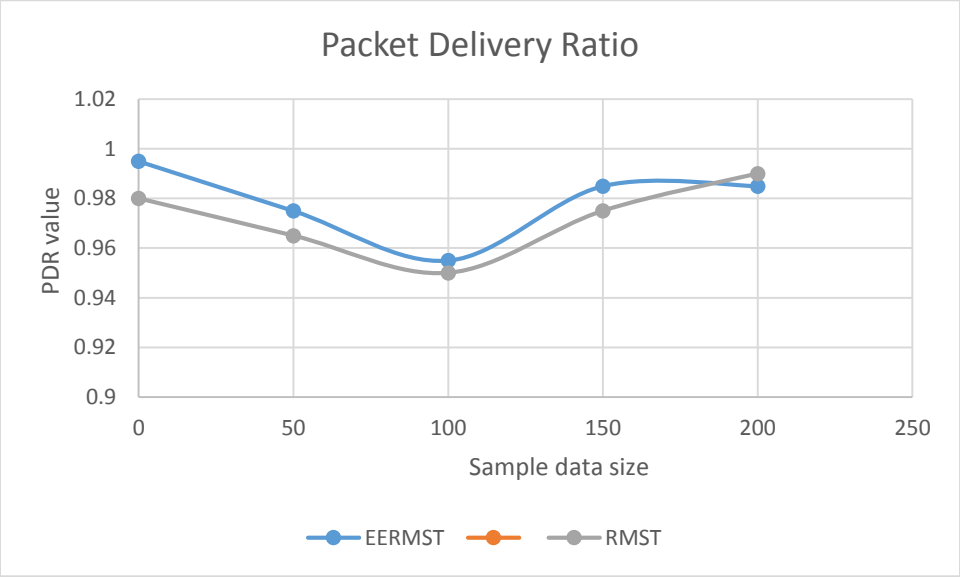


Figure 4.9.: delivery ratio of EERMST and RMST protocols

4.7. Future Work

As clearly described in section 2.6 and 2.7 of study the major functions of transport control protocols for wireless sensors networks are congestion control, guaranteeing of reliability and energy conservation. There are different existing protocols studied to achieve theses functionalities. The existing protocols focuses are only either congestion or reliability guarantee in uni-direction (upstream or downstream), and none of them settles congestion control and reliability simultaneously in both directions. Moreover, some protocols such as [33-36] only focusing on congestion control have decreased reliability. However, some applications in wireless sensor networks require both functions in both directions (upstream or downstream), for example, re-tasking needs for motes in critical time-sensitive monitoring applications, border surveillance monitoring applications. Therefore, an adaptive mechanism is required to support packet reliability and congestion control in both directions.

The developed queue algorithm was not implemented, for future it needs to be implementing the improved queue algorithm method and evaluation of this algorithm by congestion control protocols metrics. The additional line of improvement is regarding testing it on physical sensors to assess its practical significance.

The last, but definitely not the least important section is where we disclose our benchmark results on pre-defined networks, which answer commonly asked questions concerning wireless communication in wireless sensor networks. Therefore, we do not believe that network design is generic enough. The routing aspect are not considered in our design since it was beyond the scope of this work. The design can be enhanced in such a way that routing of wireless sensor networks is taken into account.

4.8. Conclusion

This thesis studded the general overview of wireless sensor network, application areas, network structure, protocol stuck, operating system used for wireless sensor network. Specifically the thesis focuses on reliable data deliver issues in transport layer of protocol stack have been discussed. In addition, general issues in designing a reliable data transport protocol for wireless sensor networks have also discussed. The survey is conducted on existing data transport protocols focusing on reliability, congestion control protocols and general issues in designing a reliable data transport protocol for wireless sensor networks. The challenges for providing reliable data delivery are unique network topology, diverse applications, small message size, resource constraints, frequent node failure, and congestion. We identify the best approach for loss detection and recovery mechanism for reliable packet delivery. The hybrid based loss detection and recovery mechanism is designed to provide a solution to last or single packet delivery problem in hop-by-hop recovery mechanism and NACK loss detection approach. The novel approach is to introducing timer based Explicit ACK approach to the NACK approach to handle the problem.

In addition, a new queue management method is also introduced in order to alleviate and handle congestion control by using priority based queue method by dropping unwanted packet from the communication. Finally, new approach was tested, and evaluated with different metrics on Tinyos 2.1.x, with Five, Ten and fifteen Micaz mote, meyer-heavy.txt full noisy file, in 50, 100 ms, and TOSSIM simulator on system application as well as developed healthcare application with 200 sample of data.

From the analysis result, The End to End delay result of new (EERMST) approach is closer to 0:0: 0.00248333625 ms. and NACK based RMST is 0:0: 0.0057716035 ms. Therefore, RMST has more delay of 0:0: 0.00328826725 ms than EERMST. Generally, we conclude that the EERMST shows the best outstanding performance in terms of end to end delay, resent rate, packet delivery ratio and link delay with given network topology.

Reference

[1] I. Khemapech, I. Duncan, and A. Miller, "A Survey of Wireless Sensor Networks Technology," in 6th Annual Postgraduate Symposium on the Convergence of Telecommunications,Networking and Broadcasting, Liverpool, UK, 2005.

[2]. I. F. Akyldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," Computer Networks (Elsevier), vol. 38, pp. 393-422, 2002.

[3] D. Estrin, "Embedded Networked Sensing Research: Emerging System Challenges," in NSF Workshop on Distributed.

[4] Mainwaring, Alan. Polastre, Joseph. Szewczyk, Robert. Culler, David. Anderson, John. Wireless Sensor Networks for Habitat Monitoring. First ACM Workshop on Wireless Sensor Networks and Applications. September 28, 2002. Atlanta, GA, USA

[5] McKelvin, M. L., Williams, M. L., and Berry, N. M. 2005. Integrated radio frequency identification and wireless sensor network architecture for automated inventory management and tracking applications. In Proceedings of the 2005 Conference on Diversity in Computing(Albuquerque, New Mexico, USA, October 19 - 22, 2005). TAPIA '05. ACM Press, New York, NY, 44-47.

[6] I. F. Akyildiz , W. Su , Y. Sankarasubramaniam , Y. , and E. Cayirci , " Wireless sensor networks: A survey " , Computer Networks (Elsevier) Journal, vol. 38 , no. 4 , Mar. 2002 , pp. 393 – 422 .

[7] I. F. Akyildiz , T. Melodia , and K. R. Chowdhury , " A survey on wireless multimedia sensor networks " , Computer Networks (Elsevier), vol. 51 , no. 4, Mar. 2007 ,pp. 921 – 960 .

[8] C. Wang and K. Sohraby, "A Survey of Transport Protocols for Wireless Sensor Networks," IEEE Network, 2006.

[9] Q. Pang, V. W.S. Wong and V. C.M. Leung, "Reliable Data Transport and Congestion Control in Wireless Sensor Networks," Int. Journal Sensor Networks, Vol. 3, No. 1, 2008.

[10] C. Wang, K. Sohraby, B. Li, W. Tang, "Issues of Transport Control Protocols for Wireless Sensor Networks," Conference on Communication, Circuit and Systems, 2005.

[11] F. Stann, and J. Heidemann. RMST: Reliable Data Transport in Sensor Networks. In Proc. SNPA '03, Anchorage, AK, June 2003, pp. 102--112.

[12] Pushkar Chavan, Sachin Pusadkar , Aditya Haralikar, Suraj Patil .RMST: Reliable Multi-Segment Transport Protocol in Sensor Networks, International Journal of Computer Science and Network, Volume 4, Issue 2, April 2015

[13] H. Zhang, A. Arora, Y.-R. Choi, and M. Gouda. Reliable Bursty Convergecast in Wireless Sensor Networks. In Proc. MobiHoc '05, Urbana-Champaign, IL, May 2005, pp. 266--276.

[14] T. Le, W. Hu, P. Corke and S. Jha, "ERTP: Energy-efficient and Reliable Transport Protocol for Data Streaming in Wireless Sensor Networks," Journal of Computer Communication 32 (2009), 1154-1171, 2009.

[14] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: A Reliable Bulk Transport Protocol for Multihop Wireless Networks.In Proc. ACM SenSys '07, Sydney, Australia, Nov. 2007, pp. 351--365.

[15] C.-Y. Wan, A. Campbell, and L. Krishnamurthy . Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks. In Proc. WIRELESS SENSOR NETWORKA '02, Atlanta, GA, Sept. 2002, pp. 1--11.

[16] C. Miller and C.Poellabauer. PALER: A Reliable Transport Protocol for Code Distribution in Large Sensor Networks. In Proc. SECON '08, San Francisco, CA, Jun. 2008, pp. 206--214.

[17] S. Park, R. Vedantham, R. Sivakumar, and I. Akyildiz. A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks. In Proc.MobiHoc '04, Tokyo, Japan, May 2004, pp. 78--89.

[18] H. Lee, Y. Ko, and D. Lee. A Hop-by-hop Reliability Support Scheme for Wireless Sensor Networks. In Proc. PERCOMW '06, Pisa, Italy, Mar. 2006, pp. 431--439.

[19] TinyOS: An open-source operating system for sensor networks, Available at: http://www.tinyos.net, Accessed June 2005.

[20] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, "The nesC Language: A Holistic Approach to Networked Embedded Systems," in Proceedings of Programming Language Design and Implementation (PLDI) 2003, June 2003.

[21] TinyOS http://www.tinyos.net 2002

[22] CrossBow Technology. http://www.xbow.com/

[23] John Wiley & Sons, Inc., Hoboken, New Jersey, "Wireless Sensor Networks: A Networking Perspective", a book published by A John & Sons, Inc, and IEEEE, 2009.

[24] D. Steere, A.Baptista, D. McNamee, C. Pu, and J. Walpole, "Research Challenges in Environmental Observation and Forecasting Systems", in proc. ACM/IEEE MOBICOM'00, Boston, August, 2000.

[25] Ian F. Akyildiz, and Mehmet Can Vuran, "Wireless Sensor Networks", USA, 2010.

[26] Bharathidasan, A., Anand, V., and Ponduru, S., "Sensor Networks: An Overview", Department of Computer Science, Technical Report, University of California, Davis, 2001.

[27] L. B. Ruiz , J. M. Nogueira , and A. A. F. Loureira , " Sensor network management" , SMART DUST: Sensor Network Applications, Architecture, and Design (edited), CRC Press , Boca Raton, FL , 2006 .

[28] B. D. Quang and H. W. Joo, "Trade-off between Reliability and Energy Consumption in Transport Protocol for Wireless Sensor Network," IJCSNS International Journal of Computer Science and Network Security, Vol. 6, No. 8B, August 2006.

[29] E. Felemban, C. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. In Proc. IEEE INFOCOM '05, Miami, FL, Mar. 2005, pp. 2646--2657.

[30] L. Wang, and S. Kulkarni. Proactive Reliable Bulk Data Dissemination in Sensor Networks. In Proc. PDCS '05, Phoenix, AZ, Nov. 2005, pp. 773--778.

[31] A. Sinha, A. P. Chandrakasan, ''Operating System and Algorithmic Techniques for Energy Scalable Wireless Sensor Networks,''Proceedings of the 2nd International Conference on Mobile Data Management, Hong Kong, Jan. 2001, pp. 199–209.

[32] Purushotham BV, Prakasha S, and Dr. K Ganesan,"Study of Reliable Data Communication in Wireless Sensor Networks," Proceedings of the International MultiConference of Engineers and Computer Scientists , Hong Kong ,March, 2008.

[33] C. Y. Wan, S. B. Eisenman and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," in Proceeding of ACM Sensys'03, 2003.

[34] C. Wang, K. Sohraby, and B. Li, "SenTCP: A hopby-hop Congestion Control Protocol for Wireless Sensor Networks," in Proceeding of IEEE INFOCOM, 2005.

[35] C. T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," in Proceeding ACM Sensys'04, 2004.

[36] C. Wang, K. Sohraby, V. Lawrence, B. Li and Y. Hu, "Priority-based Congestion Control in Wireless Sensor Networks," in Proceeding of IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC'06), 2006.

[37] O. B. Akan, I. F. Akyildiz, "Event-to-sink Reliable Transport in Wireless Sensor Network", IEEE/ ACM Transaction on Networking, Vol. 13, No. 5, October, 2005.

[38] Iyer, Y.; Gandham, S. and Venkatesan, S. "STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks". Proceedings of IEEE ICCCN 2005, San Diego, CA, USA, October 2005.

[39] N. Tezcan and W. Wang, "ART: an asymmetric and reliable transport mechanism for wireless sensor network," International Journal of Sensor Network, 2007.

[40] Hsien-Po Shiang, van der Schaar, M. ,"Queuing-based dynamic channel selection for heterogeneous multimedia applications over cognitive radio networks" IEEE Trans. Multimed. (USA), vol.10, no.5, pp.896-909, 14-18 August 2008

[41] M.Mostafa, Md. Obaidur,"Congestion Control Protocol for Wireless Sensor Networks Handling Prioritized Heterogeneous Traffic.

[42] O. Alaoui Fdili, Y. Fakhri and D. Aboutajdine," Impact of queue buffer size awareness on single and multi-service real-time routing protocols for WIRELESS SENSOR NETWORKs, International Journal of Communication Networks and Information Security," Vol. 4, No. 2, August 2012.

[43] C. Cantillas, P. Ching, et al.,"Queuing Schemes for Wireless Sensor Nodes Transmitting Prioritized Data," Research Congress in De La Salle University, Manila, Philippines, March 2-4, 2015.

[44] Ong, A. V. & Cu, G. ," Data Collection with Prioritization for Wireless Sensor Networks," Proceedings of Workshop on Computation: Theory and Practice (WCTP2013) (2014).

[45] Ms. Dharani R, Ms. Lavanya S,"Hybrid data aggregation techniques in wireless sensor network," International Research Journal of Engineering and Technology (IRJET), Vellore, TamilNadu, India, Aug-2015.

[46] Kiran Maraiya, Kamal Kant, Nitin Gupta "Architectural Based Data Aggregation Techniques in Wireless Sensor Network: A Comparative Study", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 3 Mar 2011.

[47] Vaibhav Pandey, Amarjeet Kaur and Narottam Chand "A review on data aggregation techniques in wireless sensor network", Journal of Electronic and Electrical Engineering, ISSN: 0976–8106 & E-ISSN: 0976–8114, Vol. 1, Issue 2, 2010

[48] Erfan. Arbab, Vahe. Aghazarian, Alireza. Hedayati, and Nima. Ghazanfari Motlagh, "A LEACH-Based Clustering Algorithm for Optimizing Energy Consumption in Wireless Sensor Networks", ICCSIT, 2012.

[49] Harneet Kaur, Ajay sharma, "Hybrid Energy Efficient Distributed Protocol for Heterogeneous Wireless Sensor Network", IJCA,2010.

[50] Stephanie Lmdsey and Cauligi S. Raghavendra "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", IEEE 2002.

[51] Samuel Madden, Michael J, and Joseph M. Hellerstein, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks", OSDI, December, 2002.

[52] Farizah Yunus, Nor-Syahidatul N. Ismail, Sharifah H. S. Ariffin, A. A. Shahidan, Norsheila Fisal, Sharifah K. Syed- Yusof," Proposed Transport Protocol for Reliable Data Transfer in Wireless Sensor Network (WSN)," UTM-MIMOS Center of Excellence, Malaysia, July 2016

[53] Salahadin .S," Dam Safety Monitoring Using Wireless Sensor Networks," M.S. thesis, Dept. Computer Science., AA Univ., Ethiopia, February 2013.

[54]. http://www.heart.org/HEARTORG/HealthyLiving/PhysicalActivity/FitnessBasics/Target-Heart-Rates_UCM_434341_Article.jsp#.WJFxqU197IU