2007-12

Tunneled data transmission over wireless sensor networks

Yow, Thiam Poh

Monterey California. Naval Postgraduate School

http://hdl.handle.net/10945/3023

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TUNNELED DATA TRANSMISSION OVER WIRELESS SENSOR NETWORKS

by

Yow Thiam Poh

December 2007

Thesis Advisor:	John McEachen
Thesis Co Advisor:	Murali Tummala

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Tunneled Data Transmission over Wireless Sensor Network	**5. FUNDING NUMBERS**
6. AUTHOR(S) Yow Thiam Poh	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

A technique for terminal communication through transmission links established across a wireless sensor network is developed and tested. Using protocols established for conventional wireless communication networks as a guiding principle, different methodologies for link management, and segmentation and reassembly of information are explored. A protocol for sensor network encapsulation was designed and implemented across a network of terminals and wireless sensor motes. The study concludes with a discussion of the capabilities and limitations of this technique supported by results obtained through experiments under various scenarios.

14. SUBJECT TERMS Wireless communication, Sensor network communication, Data transmission	**15. NUMBER OF PAGES** 97
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

TUNNELED DATA TRANSMISSION OVER WIRELESS SENSOR NETWORKS

Yow Thiam Poh
Major, Singapore Army
Bachelor of Engineering (Hons), University of Glasgow, Scotland, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2007

Author: Yow Thiam Poh

Approved by: John C. McEachen
Thesis Advisor

Murali Tummala
Co-Advisor

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A technique for terminal communication through transmission links established across a wireless sensor network is developed and tested. Using protocols established for conventional wireless communication networks as a guiding principle, different methodologies for link management, and segmentation and reassembly of information are explored. A protocol for sensor network encapsulation was designed and implemented across a network of terminals and wireless sensor motes. The study concludes with a discussion of the capabilities and limitations of this technique supported by results obtained through experiments under various scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Recent developments in sensor technology and wireless communication devices provide a platform for deployment of wireless sensor networks in virtually any environment. The possibility of such deployments has prompted research in both the military and commercial world.

The intent of this research is to push the boundaries of employment of sensor networks for a variety of applications, and increase the complexity level of the tasks that can be executed by these networks. Possible applications for future wireless sensor networks could range from intelligence and targeting in a military context to monitoring disaster areas in a civilian context [1].

A wireless sensor network is constructed from a large number of sensor motes that are deployed within an area of interest. The adaptive nature of the motes allows for either random or a structured deployment of the sensor motes [2]. Being inexpensive and small in size, these motes can be readily deployed and an ad hoc network can be formed almost immediately upon placement of motes into the area of interest.

Flexibility in deployment comes with limitations such as operational lifespan and range of wireless transmission of each sensor mote. Deployment in large numbers is one way to mitigate the individual mote’s limitations through sheer numbers and redundancy. Such deployment methods will allow for motes to communicate with computer terminals that are out of range through relaying of packets over other motes, increasing the effective range of operation of the wireless sensor network [2].

Although wireless communication is the preferred means of communication today, all wireless communication systems are only as strong as their weakest link. Communication will not be possible once an individual receiver station is physically out of range of the transmitting station. Considerable research has been done on the techniques that can be employed within a wireless communication architecture to overcome this limitation. Techniques most common to all is the deployment of repeater stations within the area of interest to bridge any break in the communication link due to range problems.

The basis of this requirement motivated this study into the feasibility of re-programming sensor motes within a wireless sensor network to act as this repeater. An envisioned capability is to convert these sensor motes into repeater stations during times when sensory data collection is not required, allowing for the existing wireless sensor network architecture to be used for other forms of network communication as opposed to just its primary use for sensory applications.

Essentially, sensory data transmitted by sensor motes is encapsulated within a data packet before being transmitted to a data collection station either through a direct link or through neighboring motes within the network. To allow for non-sensor data transmission to be possible within a sensor network, four modules have been created, with specific functions for each module.

A client module which resides within the transmitting station will be responsible to prepare the data sequence into a suitable format for transmission. A server module, within the same station, will be responsible for broadcasting all conditioned data packets through the sensor base station. A listen module, residing within the receiving station, will be responsible for reconstructing the received sequence. Finally, a broadcast module residing within the individual motes will be responsible for repeating all new packets and dropping repeated packets. Together, these four modules form the basic operating components of what we called the Sensor Network Adaptation Interface Layer (SNAIL) concept.

The feasibility of this concept was tested on three types of topologies, namely, a link without any hops, a link with one hop and a link with two hops. There are a total of three experiments conducted for each topology. The first experiment analyzes how sequence interval affects link performance. The second experiment analyzes how packet interval affects the link performance. The final experiment is conducted to analyze how data length affects link performance.

The sensor motes used are Crossbow MICAz motes together with the MIB520 gateway base station. All motes have been configured with the lowest RF power of -25 dBm to reduce the total distance required for the experiment.

Results from the experiments conducted on a base station to base station link show that link performance increases by either increasing the sequence interval or the packet interval. However, there is no change in link performance with different lengths of data sequences.

Similar conclusions can be derived from experiments conducted on a single hop and double hop link. However, an observation made from the single hop and double hop experiments is that, in steady state, link reliability decreases as the number of hops in a link increases. Analysis of all results obtained show that this decrease in reliability is due to a weakness in the error control protocol that was implemented.

Amendments were made to the error control protocol and the same sets of experiments were conducted once again on the three types of links. Results obtained from these experiments reflected a transmission link that is able to provide improved link performance with the amended protocol.

From the experiments conducted and results obtained, it can be concluded that tunneled data transmission over a wireless sensor network is a feasible concept. However, there are limitations to it. Limitations of the current development include, the only data that can be transmitted is text messages and error control is achieved through redundancy in packet transmission. Future development could look into the other types of data that can be transmitted through this mode and also develop a more efficient error control protocol.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

My heartfelt appreciation to my beloved wife, Shirley and son, Kellen for their kind understanding, patience and moral support demonstrated throughout the duration of my thesis research.

I am very grateful for the advice and support from Professor John McEachen, allowing me to keep focused in my research, without which, this thesis could not have been completed.

The time and advice given Professor Murali Tummala greatly helped in the development of my research.

Last but not least, I would like to extend my appreciation to all who have contributed to the completion of this thesis in one way or another.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Recent developments in sensor technology and wireless communication devices provide a platform for deployment of wireless sensor networks in virtually any environment. The environment of deployment ranges from heavily populated urban terrain in any modern city to densely vegetated primary forests.

The possibility of deployment in such terrain has prompted research in both the military and commercial worlds. The intent is to push the boundaries of employment, and increase the level of complexity of tasks that can be executed by these networks. Possible applications for future wireless sensor networks could range from intelligence and targeting in the military context to monitoring disaster areas in the civilian context [1].

A wireless sensor network is constructed from a large number of sensor motes that are deployed within an area of interest. The adaptive nature of the motes allows for either random or a structured deployment of the sensor motes [2]. Recent advances in technology have paved the way for the design and implementation of a new generation of sensor network motes packaged in a very small and inexpensive form with sophisticated computation and wireless communication abilities [3]. Being inexpensive and small in size, these motes can be readily deployed and an ad hoc network can be formed almost immediately upon placement of sensor motes into the area of interest.

However, with the physical advantage come limitations, such as operational lifespan and range of wireless transmission of each sensor mote. Swarming an area of interest with a large number of sensor motes is one way to mitigate the individual mote’s limitations through sheer numbers and redundancy. Such deployment methods will allow for motes to communicate with end user devices that are out of range through relaying of packets over other motes, increasing the effective range of operation of the wireless sensor network [2].

A. THESIS MOTIVATION

Wireless communication has been at the forefront of technological advancement in recent years. With the introduction of WiFi and WiMax technologies, reliance on wireless access for either academic or commercial applications has never been greater. However, all wireless communication systems are only as strong as their weakest link. Communication will not be possible once an individual receiver station is physically out of range of the transmitting station. There has been considerable research done on the techniques that can be employed within the wireless communication architecture to overcome this limitation. Techniques most common to all are the deployment of repeater stations within the area of interest to bridge any break in the communication link due to range problems.

Although deployment of a repeater station will overcome this problem, there are situations where deployment of a repeater for a conventional wireless architecture will not be possible, due to the size of the repeater station or simply the inability for a mobile entity to provide the power the repeater station requires for reliable performance. Therefore, it would be an ideal situation where these repeater stations are small enough to be readily deployable, yet robust enough to withstand the harsh conditions presented by the environment of deployment.

The basis of this requirement motivated a study into the potential capability of sensor motes within a wireless sensor network to act as this repeater. With the amount of technology that has been invested into the motes to generate reliable and robust network architecture, it is reasonable to assume additional tasking can be performed by these sensor motes in these situations. A potential capability envisioned is to convert these sensor motes into a distributed repeater station in times when sensory data collection is not required, allowing for the existing wireless sensor network architecture to be used for other forms of network communication and elevate the capability of this network to another level, as opposed to just its primary use in sensory applications.

The presence of a wireless sensor network creates a wireless communication architecture that is adaptive and flexible within the area of interest. Currently, the sole purpose of this ad hoc network is to perform passive data collection and monitoring of environmental changes within the area of interest. In times of environmental inactivity, these motes will go into sleep mode, to conserve their limited power supply. With the presence of such communication architecture within the area of interest, it is a wasted resource to allow the motes to go into sleep mode.

This study aims to investigate and lay the foundation for further research into the following areas:

1. Multi-tasking Sensor Motes as Repeater Stations

Typically, sensor motes deployed in the area of interest will go into sleep mode upon inactivity. Without output generated from these motes, these motes are untapped resources, but have the ability to provide some form of wireless communication through the adhoc network formed upon deployment. Notwithstanding operational frequency and bandwidth as its limitation, these motes can be programmed to be repeater stations for any wireless communication stations.

2. Terminal to Terminal Communication Capability

Utilizing the technology that these sensors have been using to transport sensory data back to the data collection terminal, other forms of information could be transported within the wireless sensor network. Data such as text messages could be transferred between terminals through the links established by these motes. This hypothesis forms the basis for text messaging within a wireless sensor network.

3. Hybrid Network Comprising Application Systems and Sensor Motes

With mesh technology in the majority of the wireless sensor networks, a flexible yet robust network coverage within the area of deployment of the motes could be easily formed. Utilizing this network could improve range of coverage for application systems requiring wireless access to data collection or transmission.

B. THESIS OBJECTIVE

As wireless sensor networks progress are widely deployed, it is almost imperative that useful applications and capabilities should be developed. As such, there is a need to study the capabilities and limitations of current wireless sensor network technology and explore possible applications that can be developed for wireless sensor networks, such as achieving other forms of wireless communication through the sensor network.

In view of the interest in future development of capabilities for sensor motes, this thesis aims to lay the foundation by conducting an investigation into the possibility and feasibility of multi-tasking a wireless sensor network to perform other tasks apart from just collection of data results generated by the individual sensor boards mounted on the motes. Specific research tasks to be addressed in this thesis are current and potential capability of a wireless sensor network, structure of packets being delivered, error control protocols for data packets, and reliability issues for communication based on a wireless sensor network.

Findings from the above tasks will assist in formulating a current set of boundaries that should be taken into consideration when developing any potential application. In addition, this study will allow developers and end-users to better understand the capability and limitations of a wireless sensor network and provide more value for future wireless sensor network research.

C. RELATED WORK

Wang [2] studied the possibility of determining the topology that has been implemented based on traffic characteristics. The study includes an investigation into the behavior of the traffic generated by different topologies of a wireless sensor network. He has also performed analysis of the traffic collected for self similarity and statistical distribution.

Jumpot [4] studied the performance of congestion control mechanisms in Asynchronous Transfer Mode unspecified bit rate (ATM-UBR) services for Transmission Control Protocol delivery. His studies include performance of the control mechanism in services with and without variable bit rate (VBR) background traffic and in both the local area and wide area networks. The result of his study is based on the efficiency and fairness of the protocol in the various types of networks that his simulation is based upon.

D. THESIS ORGANIZATION

Chapter II provides an overview of the theory of wireless sensor networks. The current capabilities and limitations of a wireless sensor network in its current state will also be discussed briefly in this chapter together with the hardware and the software part of the sensor motes. The envisioned capability of transporting information between end user devices using a wireless sensor network will also be discussed in detail in this chapter.

Chapter III covers mainly the concept of how a network protocol could be developed and implemented to fulfill the proposed capability. An overview of the proposed concept will be discussed in this chapter. Details on the capability, such as packet structure, error control protocols and limitations will be discussed in this chapter as well.

Chapter IV will describe the experimental setup with details on the specifications and configurations. The chapter will also provide discussions on the results and findings obtained from the experiments conducted. Advantages and limitations of the proposed capability will also be discussed in detail in this chapter.

Chapter V provides the conclusion to the thesis. In this chapter, future work and recommendations will also be provided.

THIS PAGE INTENTIONALLY LEFT BLANK

II. WIRELESS NETWORKS

With advances in technology, higher achievable data rates and reduction in the cost of wireless communication devices, wireless communication is becoming more common as a primary form of communication between users in both commercial and military environments.

To better understand the capabilities of wireless communication, this chapter will provide an overview on the different types of wireless communication technologies and discuss on the advantages and limitation of such a network.

A. INTRODUCTION TO WIRELESS NETWORKS

There are several ways wireless communication can be made available. The most common way is to construct a permanent communication infrastructure, complete with base stations and relay stations, to allow for communication links to be established between two terminal stations through the base stations and relay stations.

The second way to establish a wireless network is by means of adhoc networking, where relay stations are deployed within an area without wireless coverage, in an attempt to allow for on-demand and possibly unconstrained inter-connectivity between terminal stations [5].

Adhoc and rapid deployment is a capability that is especially useful in situations where wireless communication is required but the construction of an infrastructure for the network is not possible. Unfortunately, with current repeater modules of wireless networks coming in the form of building infrastructure or terminal stations, such deployment capabilities are not feasible.

To study further into the performance requirements of a wireless network, outlined below are several metrics recommended by the Arizona State Public Information Network [6] to determine the measure of performance of a wireless network.

1. Range

Range is an important parameter to look into when one is evaluating the capability and effectiveness of a wireless network. Depending on the type of wireless network and the type of network adapter that is being evaluated, a typical operating range could be between 200 ft to 650 ft indoors or up to 3280 ft in open space.

2. Throughput

Throughput is determined by the amount of data that can be processed and transmitted through a wireless connection per unit time between two terminals. Generally, a network with a higher throughput would be more desirable. However, an increase in network throughput will require data transmission at a higher data rate, which could lead to a compromise on the integrity of the data that has been received at the receiver end.

3. Integrity

Data communication within the wireless network requires a certain amount of data integrity. Failing to comply with the required data integrity level will cause information transmitted to be lost in transmission. As highlighted in point 2 above, there must be a balance between throughput and data integrity desired within the network.

4. Scalability

With the adhoc deployment nature of the wireless network, the number of users that will be connected to the network cannot be determined *a priori*. Therefore, a wireless network should be scalable to accommodate large numbers of nodes within the area of interest.

5. Battery Life for Mobile Platforms

Typically, network nodes operate on battery power since a constant power supply may not be readily available. Therefore, battery life for mobile platforms should be taken into consideration against the nature of deployment and duration of deployment for the wireless network.

6. Safety

With the basic form of communication within a wireless network being radio frequency, care has to be taken to ensure that the wireless network deployed does not cause adverse health affects to any living beings in the neighborhood of the network.

B. WIRELESS SENSOR NETWORKS

A wireless sensor network is essentially a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor the environment. The initial motivation for such capabilities was to be employed in the military environment to allow for maximum situational awareness while minimizing the requirements on humans to achieve this objective [7]. Success in experimentation in the military environment prompted researchers to look into employment of such networks in civilian application [3] areas.

Components within a wireless sensor network can be generally categorized into hardware and software. Hardware components refer to the physical motes that are being deployed in large numbers within the area of interest, much like a swarm of locusts in a paddy field. Software components are the operating system and applications that determine the operational capabilities of the motes.

1. Hardware

This section attempts to look at the hardware components of a sensor mote. In addition, the potential benefits and limitations of each sensor mote will be examined to develop a better understanding of the capability of the motes.

a. Processor

The processor, being the brain of the sensor mote, collects sensory inputs from the sensor boards, processes the information and decides on the next course of action. The processor is also responsible for processing data received from other motes and decides if the data packet should be processed, repeated or dropped.

b. Memory

Similar to all devices, the sensor motes have two types of memory embedded within, namely Random Access Memory (RAM) and flash memory. Flash memory is used to store information that should remain within the mote even after a power failure. Therefore, program code will be stored in this memory due to its stability. As for RAM, it is more of a transient memory, where the information within this memory area will be erased upon a power interruption or power failure. Therefore, only non critical information such as transient sensor data should be stored within this memory area.

c. Radio

With the deployment of motes following a swarm concept, it is inevitable that motes will have to communicate with each other through radio frequency. The radio on the sensor motes can be adjusted for a range of output power through software. Power levels range from -20 dBm to 10 dBm, with a resolution of 1dBm. RF output power level can be adjusted through software [8].

d. Power Supply

Sensor motes are designed to operate autonomously without a constant power supply from an external source. Therefore, the primary source of power would be obtained from the battery supply attached together with the sensor motes. Due to the low processing power of the sensor motes, power consumption is minimal and hence, a sensor mote could easily run for a long time with a new battery.

2. Software

This section will look into the processing portion of the sensor mote. Understanding the type of software that has been used as the operational platform of the sensor mote will enhance understanding of the processing capabilities of the individual mote. Through a good understanding of the hardware and software components of the individual mote, the range of applications of each individual mote will become more apparent.

a. TinyOS

TinyOS is an open source operating system designed for wireless sensor networks [9]. Its component-based architecture allows for rapid implementation and execution of instructions by the sensor motes, while minimizing code size, due to the limited memory size of the sensor motes.

Originally developed by the University of California, Berkeley, TinyOS is widely used as an operating system for sensor motes due to its object oriented and event driven mode of operation. TinyOS has a wide range of libraries such as data acquisition tools, communication packages and tools for data observations. This has made this operating system truly versatile for implementation in a wireless sensor network. The unpredictable nature of wireless communication requires precise control on the power management of communication devices. This requirement could be achieved through the event-driven execution model of the TinyOS.

Data transmission in TinyOS is achieved through transmission of packets. The structure of a TinyOS packet is as shown in Figure 1.

Header(5)	Payload(29)	CRC(2)

Figure 1. Packet structure of TinyOS (Bytes) (from [8])

For the header component, two bytes are reserved for the destination address of this packet. There are two general addresses that have been reserved by TinyOS, specifically: 0xFF, which is reserved for the broadcast channel, and 0x7E, which is reserved for the Universal Asynchronous Receiver/Transmitter (UART) channel. Similar to a conventional networking protocol, this address specifies the destination node that is supposed to receive this packet. In the event that a broadcast address is selected, all nodes will then receive this packet and re-transmit the packet.

One byte of the header is reserved for the Active Message (AM) type number. The structure and components of AM will be discussed further in the following section.

The remaining two bytes of the header are reserved for the Group ID of the terminal and the length of the payload, respectively. Upon receiving a packet, the mote will perform a check to verify that the packet is generated by its own group. Packets with different group IDs will be dropped and packets with the same group ID will be processed. Knowledge of the length of the payload within the packet is helpful in processing the received packet. The last byte of the header, which contains the length of the payload, fulfills this requirement, allowing for more efficient processing of the packets by the motes.

Finally, to ensure the data integrity of this packet, two bytes are reserved for Cyclic Redundancy Check (CRC) at the end of the packet.

b. Active Message (AM)

AM is a communication protocol which has a low overhead in its message packet structure [10]. This characteristic makes it especially efficient for implementation in networks such as wireless sensor networks, high speed networks or networks that have a limited amount of available bandwidth or low latency tolerance. The event driven nature of the AM protocol allows for efficient use of memory, processor and power requirements, what are generally the primary constraints presented when traditional communication protocols (e.g., TCP/IP) are being implemented in a network of mini-devices.

There are several forms of packet format that can be transmitted in TinyOS. Each packet format is identified through its unique AM type number. All packets with different group IDs will be dropped. The AM type number will be retrieved from the header of packets with the same group ID. Following which, the processor will process the payload based on the structure of the AM type number specified.

The purpose of AM indicator is to integrate communication between resource limited mini devices and the conventional computer terminals. The primary means of achieving this target is by ensuring that AM messages are being processed sequentially. In addition due to the nature of AM, all packets that are being received by the individual motes will be processed as fast as possible and buffering of messages will be kept to a minimum. If there is a need to keep a certain individual packet, the packet of interest has to be kept at a separate memory location. Through the above methods of communications, processing of AM packets is more efficient and resources for the mini devices are better conserved.

c. JAVA

Java is a programming language that allows for applications to run on a variety of computers [11]. This ability is achieved through the implementation of a virtual machine on any operation system (e.g., Mac OS, Linux or Windows). This language creates two main types of programs, namely applications and applets. Briefly, an application is a program that is executed under the operating system of the terminal, while an applet is an application that is designed to be transmitted over a network. Conventional software programs are governed by the circumstance that the processor will be in. Factors such as devices that will be affected or changes in circumstance will influence the programmer on how the program should be written, so as to ensure robustness of the program. Java's approach of object oriented programming is a paradigm shift away from the conventional programming concept. With a set of well-defined interfaces to the data, Java organizes its program around its data, hence, reducing programming complexity [11].

Another paradigm shift in software programming made possible with Java programming is running of applications without an underlying operating system [11]. With the emergence of Java Virtual Machine, battery operated sensor devices will be able to execute functions such as over-the-air programming or even performing device computations within the device itself.

In addition, the platform-independent ability of Java makes it a suitable programming language to be used in wireless sensor networks. With this ability, Java is able to act as a common language for all devices with different operating systems to interact with each other.

3. Proposed Function for Wireless Sensor Network

The common operating fundamentals of a wireless communication network and wireless sensor network suggest that it is possible to form a wireless communication network that is a hybrid between that of a conventional wireless network and a wireless sensor network.

Due to the cost of deployment and maintenance of conventional repeater stations, it may not be a viable option to setup an adhoc wireless communication network to provide network coverage in an area where communication between terminals is minimal. In such instances, if there is an existing wireless sensor network, it would be more cost effective to re-program the sensor motes to operate as a distributed set of repeaters which establish a communication link between the terminals of interest and perform the role of a repeater station. Wireless communication between terminals will no longer be limited to the deployment of bulky or power consuming repeater stations for minor communication requirements.

The only consideration for communication through such means is the limited resources in each sensor mote. This will limit the type of information that is able to be transmitted.

C. SUMMARY

This chapter highlighted the similarities and differences between a conventional wireless network and a wireless sensor network. Factors that determine the operating boundaries of a conventional wireless network were discussed to better understand the guiding factors for this network. In addition, the respective working modules of a wireless sensor network were discussed, to gain a better understanding of the capabilities and limitations of a wireless sensor network.

The next chapter will study the guiding principles and the configuration required for a sensor mote to allow it to operate as a repeater station in a wireless sensor network.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SENSOR MOTES AS REPEATER STATIONS FOR WIRELESS NETWORKS

It has been observed that fundamentally, the mode of operation for a wireless sensor network is similar to that of a wireless network. Although a wireless communication architecture is established for the employment of both networks, the use of sensor motes as primary means of communication between terminals as part of a wireless communication network architecture will not be desirable due to the power constraints of the individual motes.

Programming sensor motes as repeater stations for wireless data communication is not a feasible option, but it is interesting to look into the possibility of using sensor motes as repeater stations on an adhoc basis. In a situation where two terminal stations within a sensor network face a requirement to transfer data between each other, it would definitely be useful to have the sensor motes be able to achieve this task rather than to setup another independent wireless network to fulfill this requirement.

Essentially, sensory data transmitted by sensor motes is encapsulated within a data packet before being transmitted to a data collection station either through a direct link or through neighboring motes within the network. To allow for non-sensor data transmission to be possible within a sensor network, there must exist an interface layer between the terminal station and the sensor network, to encapsulate the data of interest into a data packet, such that data from the terminal is being encapsulated, transmitted and repeated by the sensor motes instead of sensory data.

In addition, data encapsulated packets should be transparent to the sensor motes, such that to the sensor motes, these packets will be processed as a normal packet.

A. FACTORS TO BE CONSIDERED

To achieve this capability, there is a need to look into several factors that would otherwise limit the ability of the sensor network to multi-task as a data transmission network. Some factors that are considered are listed as follows:

1. Variable Length Data Segments

For a normal sensor network environment, there are a fixed number of sensors that are generating information and therefore the packet length and the length of payload within the packet can be predetermined. The knowledge of this information in advance also allows the packet structure to be pre-determined. Consistency in packet structure and length of payload within the packet reduces additional processor resources for both the transmitter and the receiver to allow for transmission of information to occur.

However, in the scenario of data originating from a terminal station being transmitted, there is no guarantee that the length of the data to be transmitted will be fixed. This flexibility in the length of the data string adds to the complexity of the encapsulation protocol, to ensure that the information has been encapsulated correctly and efficiently.

2. Error Control Protocols

In a conventional sensor network environment, for a set of sensor data to be sent back to the data collection terminal, this information will typically be broadcast by the individual sensor motes, with a specific destination address. Similar to that of the User Datagram Protocol, delivery of this information to the destination address is on a best effort basis. There is no significant effort to ensure that all packets broadcast are being received correctly at the receiver end.

Due to the nature of transmission within a wireless sensor network, types of error include but are not limited to packets being dropped at intermediate motes due to congestion, or packets arriving out of sequence, causing reconstruction of data stream to fail. Therefore, a certain level of error control has to be implemented in this interface layer to improve on the level of integrity of the data that is being transmitted and received through this architecture.

There are several ways that error control could be achieved. Some of the methods could include transmission of redundant packets such that in the event of packets being dropped, there is still another packet being transmitted. Constraints with this method would be that monitoring of the sequence number of each individual packet has to be done at the receiver to ensure that no duplicate packets are being processed.

Another form of error control protocol could be implementation of a request for re-transmission by the receiver (as implemented in Transmission Control Protocol). Upon reception of all packets, the receiver will process the packets and arrange them in the order of their sequence number. Following which, a check for lost packets would be performed. In the event of a lost packet, a request for specific packets or for the re-transmission of the entire data stream is made to ensure accuracy in the data stream that is being reconstructed.

The latter method will require a certain amount of handshaking between the transmitter and the receiver. This requirement is undesirable in the wireless sensor network due to the inherent constraints of the sensor network.

As mentioned, there are several methods that can be implemented within this interface layer to mitigate the probability of erroneous packets being received at the receiver end. The general concept is to ensure that the transmitter will be transmitting packets in a manner that any possible errors or a compromise in data integrity is being handled at the onset of transmission.

3. Motes and Base Station Area of Coverage

In a normal situation for deployment, sensor motes will be randomly placed within the area of interest. However, where possible, efforts should be taken to ensure that there is considerable spacing between each individual mote and the base station as well. The effects of distance between motes and base stations are clearly highlighted in Figure 2.

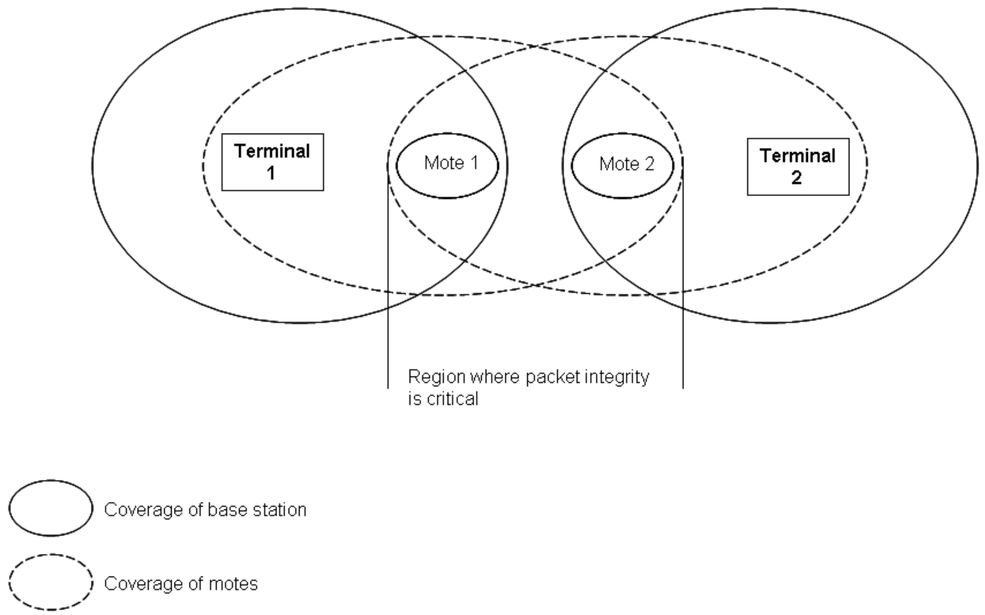

Figure 2. Factors concerning area of coverage

As observed above, with two motes being within range of each other, Mote 2 will broadcast the packet it has received from Mote 1 after verifying that it is a new packet. However, as both motes are within range of each other, the re-broadcast packet will be received by Mote 1 again.

At Mote 1, this packet will be checked once again, and will be dropped since it is a repeated packet. New packets from Terminal 1 base station will be dropped if these packets arrive when Mote 1 is in the process of verifying the status of the packet it has received from Mote 2.

Due to the possibility that packets will be transmitted back to its originating mote, there will be a region where the probability of packets being dropped is highly probable. With more motes being deployed in close proximity, this probability will definitely increase, further compromising the data integrity of the network. Therefore, a desirable situation is where the density of motes within an area of interest is sufficient to establish a stable network but not high enough to create multiple regions where packet collision and packet drops are high, as highlighted in Figure 2.

To achieve an optimal mote density per unit area, dispersal techniques for the motes would have to be taken into consideration together with the range of the individual motes and the base station.

3. RF Transmission Power for Motes and Base Stations

The transmitted power is taken into consideration in conjunction with the previous point about minimum deployment distance between motes. In addition to antenna sensitivity, transmission power will also affect the maximum distance that the transmitted packet can reach. Higher power transmission will also exhaust the battery power in a shorter time as compared to a mote with a lower transmission power.

On the contrary, with the minimum distance being taken into consideration, a mote that has a higher transmission power will require fewer motes to be deployed within an area of interest to achieve the same level of coverage. Therefore, there must be a fine balance between the numbers of motes required for deployment and the expected deployment time for each individual mote against the RF transmission level of each individual mote.

B. PROPOSED SOLUTION

In this section, several solutions to the previously discussed factors will be studied and the most feasible solution will be selected for experimentation.

1. Segmentation and Padding of Variable Length Data

In order for data transmission to be a viable option within the sensor network, the interface layer must be able to determine the length of the incoming data stream and segment this data stream into multiples of the packet payload.

At the receiver end, the interface layer must be able to recognize that the packet received is actually part of a longer string of data information and be able to reconstruct the entire length of data before processing the information.

Drawing similarities from the concept of ATM-AAL5 [12], a proposed solution is to ensure that the interface is able to read the entire data length and calculate the length of the incoming data stream. With knowledge of the length of the data stream, this interface layer will then be able to break the data stream into multiples of the payload size to prepare for transmission over the network. Remaining bits of data that are not able to fill up the entire packet payload will be padded with redundant bits to ensure that all data bits have been correctly encapsulated.

Prior to transmission, a signaling packet with its key components, as shown in Figure 3, must be sent to apprise the receiver on the information that it requires for the reconstruction of the data packets that it is going to receive.

Dest Addr	AM Type	Grp ID	Payload Length	Seq No	Terminal ID	Total no. of packet	Padding Bits	Option Number	Remaining 24 bytes reserved for expansion

Figure 3. Structure of a signaling packet

It can be observed from the format in Figure 3 that the first five parameters for the packet actually contains header information for TinyOS transmission. Updating and utilization of parameters in this field will be determined and processed autonomously by TinyOS. However, information on the AM type number will have to be dictated by the user. Parameters in this field can be set in AM's header file, *AM.h*.

The function of this signaling packet includes allowing the receiver to know the total number of packets that it should be expecting, and the number of padded bits that have been included in this transmission. Such information will allow the receiver to correctly strip off the padded bits and reconstruct back the data stream for higher layer processing.

The structure of a data packet is as shown in Figure 4. A typical data transmission sequence will consist of one single signaling packet followed by multiple data packets. There is a need to ensure consistency in the length of the payload of all data packets that are being transmitted across the network. This consistency will assist in reducing the complexity in error detection at the receiver.

Dest Addr	AM Type	Grp ID	Payload Length	Seq No	Terminal ID	Remaining 27 bytes reserved for data. Padding bits will be added if required.

Figure 4. Structure of a data packet

Consistency in packet payload can be achieved through the conceptual implementation of ATM AAL-5 within this network. Although ATM AAL-5's payload is a total of 48 bytes [12], the proposed network has only 27 bytes of payload. Therefore, taking this constraint into consideration, a data stream that is required to be transmitted will be dissected into multiples of 27 bytes. Since the last data packet will collect the remaining data bytes, padding bits will be added to this packet to ensure that the payload is fixed at 27 bytes.

2. Redundant Packet Transmission

With the probability that packets may be dropped due to the status of the sensor mote at the time of packet's arrival, packets will be transmitted twice to generate redundant packet transmission. Redundant packets are being transmitted so that receiver will be able to receive all packets (either the first packet or redundant packets transmitted) in the event that some of the packets have been dropped.

However, with this implementation, there needs to be a second level check to ensure that the receiver will be able to recognize a duplicate packet and drop the packet accordingly.

3. Fixed Transmission Interval

Implementation of a fixed delay interval between transmissions of packets will allow time for current packets in the sensor motes to be completely processed and re-transmitted, mitigating the probability of congestion at the sensor motes due to a sudden influx of data packets.

It has to be noted that having a fixed transmission interval for a network of a certain size or topology does not necessarily mean that this timing interval will work for a network of another size or topology. A network with a daisy chain topology may require a 100-ms interval between transmissions for a reliable link, while a network with a star topology may require 300 ms to achieve the same level of reliability.

As the main mode of operation for sensor motes is through broadcasting of data packets, a network with a more complex topology will require a longer time for the current packet to be registered by its surrounding motes before another packet can be sent. With this as the limitation, the main consideration to take into account when determining the transmission interval would be the complexity of the topology that network is configured for.

The proposed methods highlighted above require the majority of the work to be done at the transmitter end and minimal processing to be done at the receiver end. This methodology will allow resources at the receiver end to be freed up as soon as possible to prepare for the next transmission, if required.

At the receiver end, one of the tasks for the receiver will be to ensure that no duplicate packets are being processed. To achieve this, the receiver will have to check the sequence number of the packets that it has received and drop all packets that are duplicates of previous packets.

The other task of the receiver will be to reconstruct the data sequence that has been dissected by the transmitter. To correctly reconstruct the sequence, the receiver will first have to remove all the header information from the individual packets and store the payload in a buffer. Once all the transmitted packets have been received, the receiver will then concatenate the entire received payload previously stored in the buffer to achieve the transmitted data sequence.

The methods highlighted above do not require much handshaking between the transmitter and the receiver. The receiver only checks on the sequence number of the packets that it has received and drops all repeated packets. As such, it can be concluded that the protocol employed in this concept can be very similar to that of a User Datagram Protocol.

C. DEVELOPMENT OF WIRELESS SENSOR NETWORK ADAPTATION INTERFACE LAYER (SNAIL)

The main motivation for this interface layer is to allow terminal stations to communicate with each other within a sensor network, without the requirement of a conventional wireless networking architecture to be present.

There are a total of three components that have been designed for the trial concept of this interface layer. First, to allow for future expansion of this research, there is a simple server module that has been written to demonstrate the role of a server station. The role of the server station is also to prepare the data stream to be ready for transmission and to implement all network protocol rules that have been set.

Second, there is a client module which replicates a data generating device. By grabbing data for transmission from the input devices of the terminal station, users performing trials will be able to pre-determine the length of the packet and the number of packets that is required for transmission.

Finally, there is a listen module whose primary function is to collect packets that have been received by the individual base stations and attempt to reconstruct the packet with the information that has been obtained. This section will look into the three portions mentioned above in detail. Shown in Figure 5 illustrates the interface layer denoted in the specific working modules.

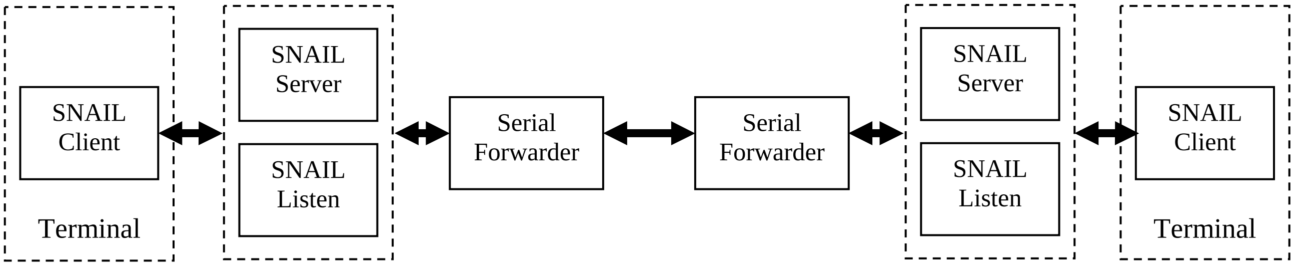

Figure 5. Working modules in SNAIL

1. SNAIL Server Module

This module serves as the main data processing portion of the interface layer whose primary function is to read in data that are being queued for transmission, prepare the data stream and send them to the base station for broadcasting into the wireless sensor network.

Once the data stream has been completely received, the server converts the stream into ASCII characters and proceeds with segmenting the entire data stream into packets of 27 bytes each. The reason for converting the data stream into ASCII characters is to facilitate the storing of the information into TinyOS packets for transmission in the subsequent stages of operation.

Next, the server will continue to perform a simple calculation on the number of padding bits required and stuff the last packet with the required number of padding bits accordingly. Upon completion of the above mentioned operation, the server module will have sufficient information on the data stream to construct the signaling packet for transmission.

As shown in Figure 4, there is other information that is required by the server to be included in the signaling packet so as to ensure that the receiver will be equipped with the necessary information for the reconstruction of the subsequent packets. As a basic form of error control, all packets will be transmitted twice, to increase the probability of a packet being successfully transmitted across the wireless sensor network to the receiver station.

The flow diagram in Figure 6 represents the execution sequence within the SNAIL server upon receiving a data stream.

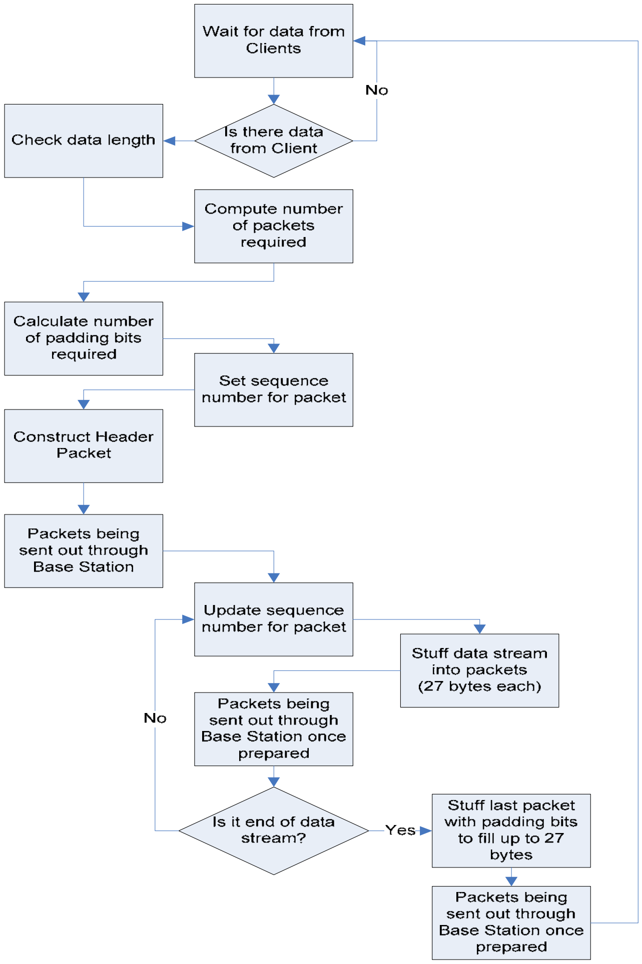
Wait for data from Clients
No
Is there data from Client
Check data length
Compute number of packets required
Calculate number of padding bits required
Set sequence number for packet
Construct Header Packet
Packets being sent out through Base Station
Update sequence number for packet
Stuff data stream into packets (27 bytes each)
No
Packets being sent out through Base Station once prepared
Is it end of data stream?
Yes
Stuff last packet with padding bits to fill up to 27 bytes
Packets being sent out through Base Station once prepared

Figure 6. Flow Diagram for sequence of SNAIL server

In addition to the sequence flow of the server module, it would also be beneficial to have a look at the separate interfaces that this module utilizes and understand the role of the respective interfaces to allow the server module to achieve its required functions.

Figure 7 shows the separate interfaces that the server module requires for collecting the required information.

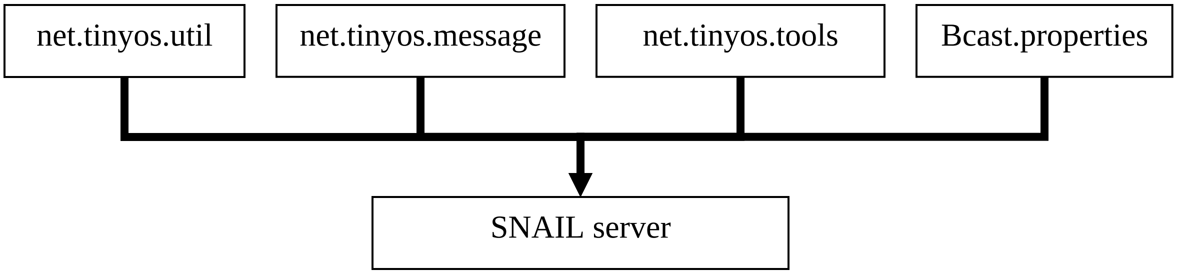

Figure 7. Interface for Server module

Net.tinyos.util contains common utility classes used by several TinyOS applications. The application used here is a SerialForwarderStud, whose main purpose is to allow communication to occur with another Java application, SerialForwarder (as shown in Figure 5). The primary purpose of SerialForwarder is to act as a parallel-to-serial converter, allowing data bits generated by the terminal station to be serially pushed to the base station for broadcasting into the wireless sensor network.

Net.tinyos.message is a message format layer whose main application in the server module is to translate the packet that is being prepared by the server into TinyOS message format for transmission over the air through the wireless sensor network's base station.

Net.tinyos.tools is where the server module is stored. This package is more of a repository of applications that can be used to communicate with the motes. Along with the server module, there are several other applications that are useful in debugging the transmission link and the integrity of the data packets when they are being received by the receiver.

Bcast.properties is where the current sequence number of the packets is stored. When a packet is ready for transmission, the server module will request a sequence number for the packet from this file. When a sequence number has been issued, the sequence number will be incremented by 1. One constraint is that the range of the sequence number is currently limited to 127.

2. SNAIL Client Module

At this stage, transmission is currently limited to text messages generated from computer terminals. As such, the function of the client module is fairly straight forward. There are several proposed capabilities that could be included in this project. Proposed capabilities could include file transfer protocols, to allow for transfer of a small file through this network. However, due to time constraints, only text messaging is addressed in this thesis.

Using a blocking reader module, upon invoking the client module, the client will wait for a message to be generated from the keyboard of the terminal station. The nature of the blocking reader will be to ensure that no action will be triggered by the client module unless there is some data input from the terminal station.

Upon receiving the message to be transmitted, the client will convert the string of information into a character array. Casting the information into a character array allows for modifications to be made to the data streams if required. It has been acknowledged that there are possible future expansions to this thesis and hence, to allow for further expansion in this project, data streams should be stored in the buffer of the transmitter in such a manner that modifications by subsequent developers would be easily performed when necessary.

Once the data array has been modified and is ready for transmission, the client module will convert the data array back to a string and have this string of data sent to the server module for broadcasting into the wireless sensor network. Once the string of data has been sent to the server, the client module will continue to wait for new data from the terminal station for transmission.

The flow of operation for the SNAIL Client module is as illustrated in Figure 8.

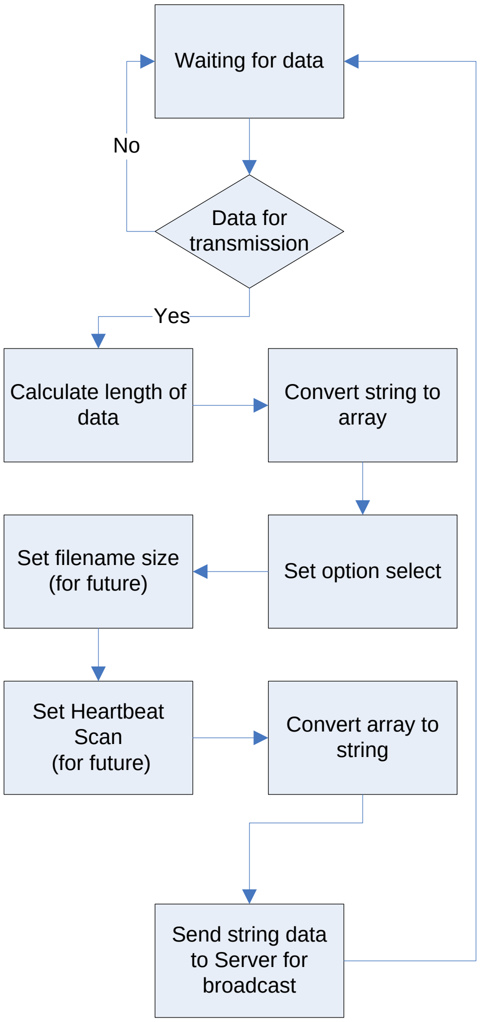

Figure 8. Flow Diagram for sequence of SNAIL Client

3. SNAIL Listen Module

The third module in the interface layer is the Listen module. Using a blocking reader to receive all the packets that have been transmitted, the primary function of this module is to perform the task of reconstructing the received packets.

Due to the transmission of repeated packets as a form of error prevention through redundancy by the transmitter, the listen module will be required to perform a sequence number check on all packets that have been received. This check is to ensure that duplicate packets are dropped and not processed when they have been received. Processing duplicate packets will cause an offset in the pointer of the buffer and also corrupt the total packet count in the receiver.

Corruption of this information causes an unrecoverable error in the reconstruction of the current data stream and in extreme cases, data integrity of the subsequent data transmission could also be compromised.

Primarily, all information that the Listen module requires for the proper reconstruction of the packets will be stored in the signaling packet. Due to the sequence of packets being transmitted, the receiver understands by default that the first packet received will be the signaling packet. As the packet structure of the signaling packet is different from the packet structure of the subsequent packets, the packet processing sequence for the first packet that the receiver receives will be different from the packet processing sequence for the subsequent packets received by the receiver.

Information required for the reconstruction of the data stream will be extracted by the receiver from the various fields within the signaling packet accordingly. The subsequent packets that the Listen module receives after it has completed processing the signaling packet will be the packets containing the data of interest. As such, information stored within the payload field of the subsequent packets will be extracted and stored in a buffer within the end user devices. With the “total packet count” indicator provided by the signaling packet, the Listen module will be able to know in advance how many packets it should be expecting for the entire data stream to be reconstructed.

Once the total number of packets have been received and stored into the buffer, this module will then retrieve the data from the buffer and convert it into a string of data, before handing it to a higher layer for processing.

For this thesis, the capability of the Listen module will be limited to only converting the payload of the received data packets to string data and displaying the received information on the screen of the processing terminal, allowing the user to verify that the correct data stream has been received and processed.

The flow of operation for the SNAIL Listen module is shown in Figure 9.

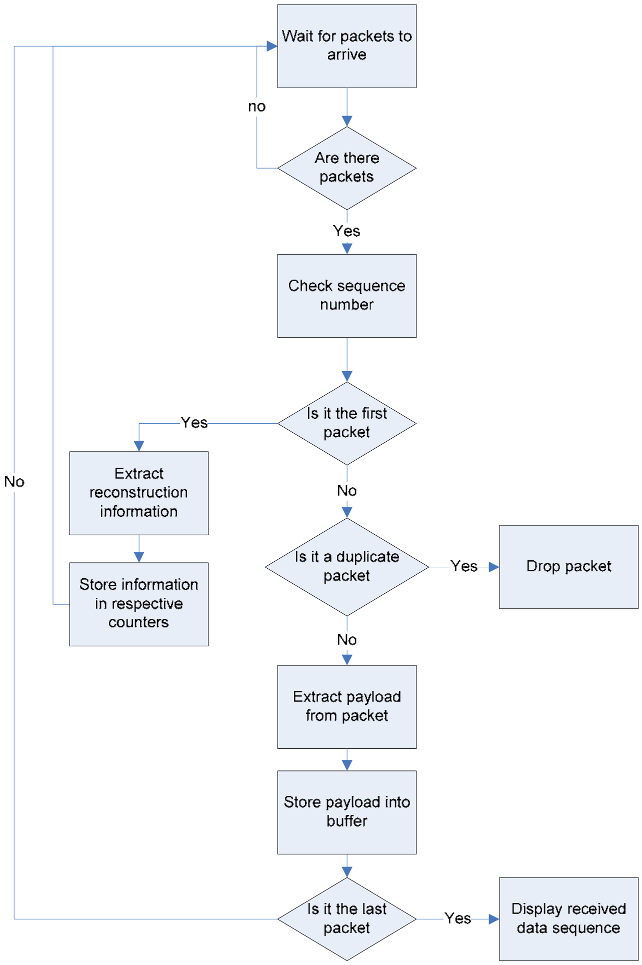
Wait for packets to arrive
no
Are there packets
Yes
Check sequence number
Is it the first packet
Yes
Extract reconstruction information
Store information in respective counters
No
No
Is it a duplicate packet
Yes
Drop packet
No
Extract payload from packet
Store payload into buffer
Is it the last packet
Yes
Display received data sequence

Figure 9. Flow Diagram for sequence of SNAIL Listen

4. Sensor Broadcast Module

Figure 10 depicts the placement of the broadcast module in the transmission link.

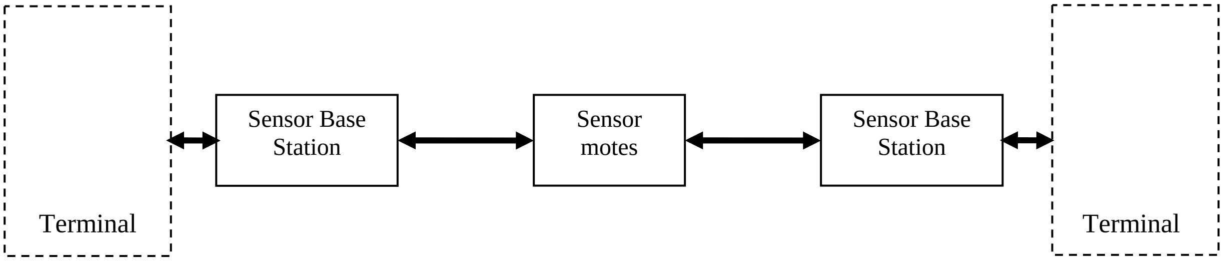

Figure 10. Placement of sensor as a repeater

Fundamentally, this module programs the sensors that have been deployed in the area of interest to check on the sequence number of the packets they receive and broadcast these packets if they are not a duplicate packet. As compared to the previous software that the sensor motes have been programmed with, this module does not require the sensor motes to extract and process the information that is being embedded within the payload of the packets. The only information that is required by the motes is the sequence number in the header field of each packet.

Acknowledging that processing speed of the motes and the overall network delay will be one of the key performance indicators, processing only the sequence number within the packet greatly reduces the processing time required by the packet at each individual mote.

D. SUMMARY

This chapter provided an overview of how the operating modules are linked functionally and the overall operational concept of SNAIL. It includes examination of several networking issues that are common in all network environments that would affect the reliability of SNAIL. The considerations taken for the development of the packet structure and the different modules were discussed to allow for a better understanding on the concept of operation of SNAIL.

The next chapter will look into the different experiments that have been conducted on the proposed concept of SNAIL. Results obtained from these experiments will include the transmission interval requirement for each packet and interval requirement for each data sequence. Proposed amendments to the protocol will also be investigated and discussed. The results achieved by both protocols will be compared and discussed.

IV. EXPERIMENT SETUP AND RESULTS

There is a need to understand the capabilities and limitations of SNAIL. This objective could be achieved by conducting experiments under several scenarios to evaluate the performance of the transmission link established using SNAIL.

This chapter aims to provide a description of the physical setup of the experiments and the results that have been obtained from the runs conducted under different scenarios. It also includes a short discussion on the results that have been obtained and capabilities or constraints that have been validated.

There are several scenarios where an experiment will be conducted and results are obtained. Firstly, communication between a base-station to another base-station will be conducted to verify that terminal to terminal communication can indeed be executed within a wireless sensor network. Subsequently, complexity will be included in the link by including a single hop and, later, two hops within the transmission. Variations on the transmitter end will include altering the timing interval between the transmission of packets and varying the packet length.

A. EXPERIMENT SETUP

The experiments are conducted in a laboratory environment where the motes (if required) and base stations are placed within line of sight of each other. There are three variations of networks that will be tested: a direct base-station to base-station transmission, a link with a single sensor mote in the middle and a link with two sensor motes in the middle.

1. Hardware

The hardware that was used in the experiments included two base stations, two sensor motes and two laptops for communication. Details of the individual hardware components that were used in this experiment are as follows:

a. MICAz Sensor Motes

MICAz motes are small wireless hardware platforms which function as both a wireless sensor mote when a sensor board is attached or as a wireless node when the sensor board is disconnected [8]. As the ability of the sensor mote to act as a wireless node is the capability of interest in this thesis, sensor boards are not connected during these experiments.

b. MIB510 Parallel Programming Board

The MIB510 contains an RS-232 serial port for programming and for external communications with the PC [8]. Acting as a gateway between the PC and the sensor motes, this programming board has an onboard ATmega 16L to translate instructions from the PC into logical processes for the sensor motes.

2. Software

There are two programming languages that were used to program the equipment for their respective functions. For the PC portion of the project, Java was used to extract the data of interest from the terminal station and organize the data in a manner that was suitable for transmission within the wireless sensor network. For the sensor motes, nesC was used to program the motes to function primarily as repeater stations.

B. EXPERIMENT SETUP FOR BASE STATTION TO BASE STATION COMMUNICATION

For this case, the setup of the link is as shown in Figure 11.

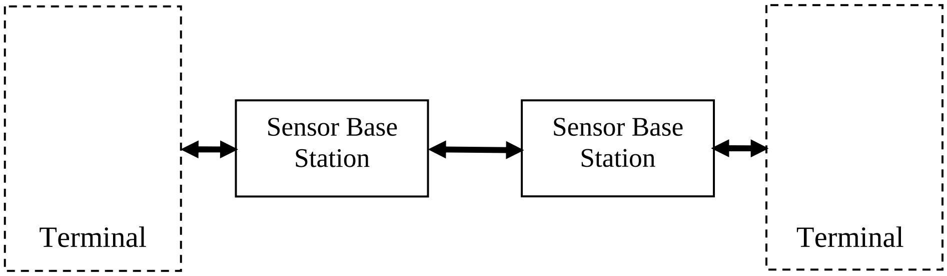

Figure 11. Experiment 1 – Base station to Base station communication

The aim of this setup was to establish the possibility that data transmission between terminals could be implemented within a wireless sensor network, instead of a conventional wireless local area network card.

1. Examination of the Effects of Timing between Data Sequences

This experiment consists of two terminals that are placed in close vicinity and within line of sight with each other. The conduct of this experiment is straightforward, with one terminal station sending the data stream and the opposite station in receiving mode, translating the information that it has received.

The data displayed on the screen of the receiver were checked and the results tabulated. The transmission sequence of the test data can be visualized as shown in Figure 12.

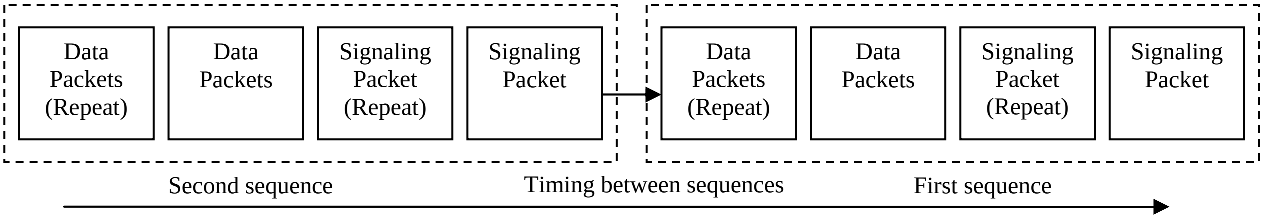

Figure 12. Sequence of transmission of test packets

To better understand the capabilities and limitations of this data transmission link, the interval between sequence transmissions were varied and other parameters under which the packets are transmitted, such as interval between packets and length of data transmitted, were fixed at 1 ms and 22 bytes, respectively.

The test sequence and the number of repetitions are as shown in Table 1.

Data Length	Test message	No of packets transmitted for each test message	No of times transmitted
1 packet	This is a test message	4	20

Table 1. Test Sequence for experiment on interval between sequence

The results obtained from running the experiment are as shown in Figure 13. It should be noted that although all 80 packets that were transmitted were fully received by the base station, not all 20 messages were correctly reconstructed for cases where the timing interval was lower than 80 ms. This phenomenon is because packets received by the base station are not necessarily processed by the receiver.

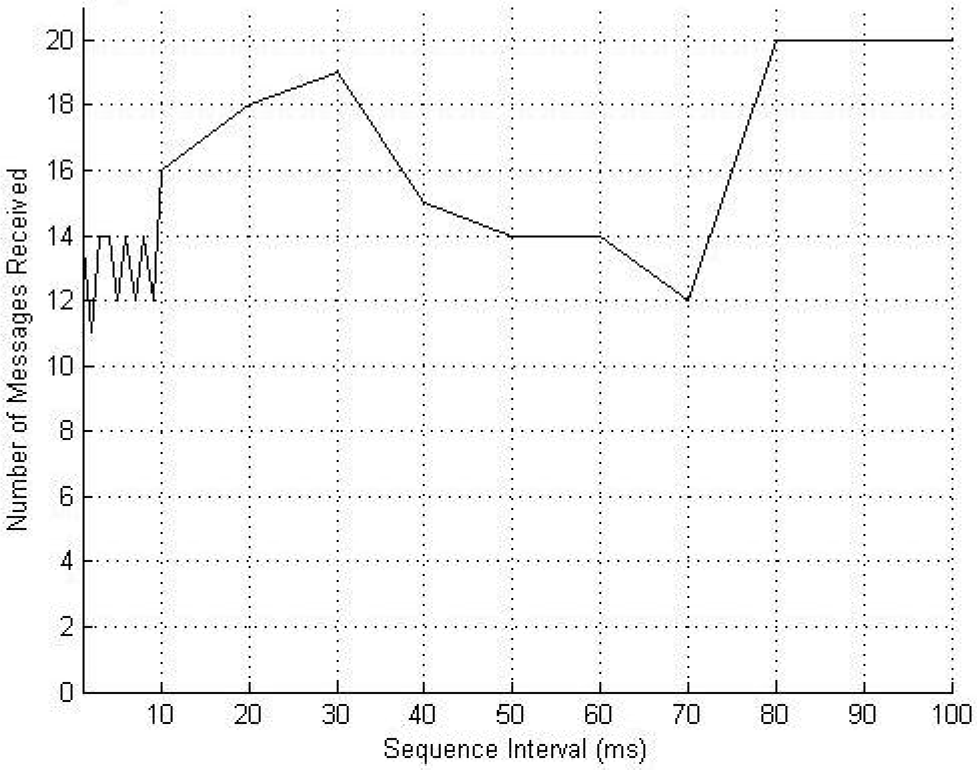

Figure 13. Results obtained by varying the interval between sequences

To reduce the processing complexity in the receiver, the processor is programmed such that a packet received when it is in the midst of processing a previously received packet is dropped. Packets will only be processed if the processor is available upon its arrival. Loss of packets will result in the message not being reconstructed correctly.

However, it can be observed that with sufficient delay inserted into the transmission, reliability of the link increases. The reason for the increased reliability is due to sufficient amount of time given to the receiver to process all received packets before the next transmission sequence arrives. By understanding this limitation, the next part of the experiment will be to investigate what impact varying the timing interval between packet transmissions will have on the message recovery performance of the link.

2. Examination of the Effects of Varying the Timing Interval between Packets

For this part of the experiment, a message that had a constant length of 22 bytes was transmitted through the link. In addition, to ensure a reliable link for the experiment, a sequence interval of 100 ms was used. A total of three runs were conducted for this portion and the results obtained are tabulated in Table 2.

Run Number	Data Length	Test message	No of packets required	No of times transmitted	Number of packets transmitted	Number of packets received	Number of messages correctly reconstructed
1	1 packet	This is a test message	4	20	80	40	20
2	1 packet	This is a test message	4	20	80	40	20
3	1 packet	This is a test message	4	20	80	40	20

Table 2. Results for experiment run with 1 ms as time interval

It was observed that there was a 100% recovery of the data that was transmitted across the link even though there was only a 1 ms delay between transmissions of data packets across the network. The reason for the 100% recovery was due to the synchronized processing speed of both base stations.

For a base station to base station transmission, the rate at which packets were leaving the transmitter was similar to the rate of packets reaching the receiver base station. As such, the entire chain of data packets was received and processed by the receiver in such a synchronized and timely fashion that there were no packets that were dropped or data streams corrupted.

Based on the results obtained from the run above, it is clear that there was no requirement to conduct further tests to examine the effects of interval timing between packet transmissions. With 100% recovery of all data transmitted at the rate of 1 ms per packet being transmitted, this transmission link will be able to handle transmission with intervals between packets at a slower rate.

To further understand the capabilities and limitations, the following experimental run will be conducted with the length of the data being varied, generating a scenario where multiple packet transmissions will be required for the entire data stream to be transmitted.

3. Examination of the Effects of Variable Length Data

The intent of this experiment was to examine the effects on message recovery of the link when data sequences of various lengths were transmitted. As such, we set the timing interval between packets at 1 ms and the timing interval between sequences at 1 sec.

In this part of the experiment, several data streams, each of length several packets long were transmitted across the transmission link. Results from this experimental run were compiled and provided in Table 3.

Data Length	Test message	No of packets transmitted for each test message	No of times transmitted	Total number of packets transmitted	Total number of packets received	Number of messages correctly reconstructed
1 packet	This is a test message	4	20	80	80	20
2 packets	This is a test message 1 This is a test message 2	6	20	120	120	20
3 packets	This is a test message 1 This is a test message 2 This is a test message 3	8	20	160	160	20
4 packets	This is a test message 1 This is a test message 2 This is a test message 3 This is a test message 4	10	20	200	200	20

Table 3. Results for experiment run with variation in message length

The results obtained above further validate the conclusion that was drawn in the previous experiment about the capability of this transmission link when it was employed in a base station to base station communication mode.

By varying the length of data that was required to be transmitted, coupled with the interval timing between packets transmission set to 1 ms, a total of 100% data recovery was still achieved. This result establishes the understanding that variation in length of data to be transmitted has no effect on the performance of the transmission link established directly between two base stations.

C. TRANSMISSION WITH ONE HOP

The success of the experiments conducted with the base station to base station link motivated further experiments to be conducted with a single mote as the repeater station and subsequently, two motes as repeater stations.

This section of the thesis will look into the experiments that were conducted with a single mote in between two terminal stations to allow the capabilities and limitations of a single hop through a sensor mote to be studied in greater detail.

Similar to the previous experiment, the parameters that were varied were the timing interval between packets being transmitted and the length of the data message that was transmitted.

1. Examination of the Effects of Varying Timing between Data Sequences

The configuration of this experimental run is as shown in Figure 14. This experiment was conducted with two terminals placed at a distance where both stations were out of range with the other station. An effort was also made to ensure that there was no line of sight between the two terminals.

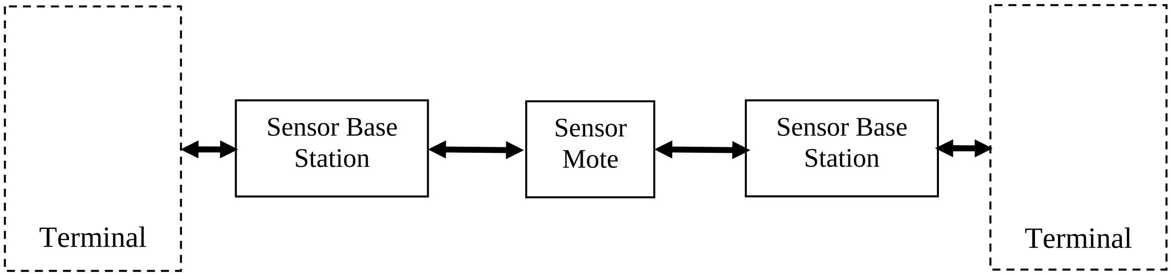

Figure 14. Setup with single mote to establish link

A single mote was placed at a position where it was able to bridge the link for both terminals that were out of range and without line of sight with each other as depicted in Figure 14. The conduct of this experiment was straightforward, with one terminal station sending the data stream and the opposite station in receiving mode, translating the information that it had received, with the transmitted sequence being read and broadcast by the repeater mote in the process.

For this experiment, the data that was transmitted was fixed to the length of one packet, timing intervals between transmission sequences were varied and the results from the different runs were consolidated and tabulated in Figure 15.

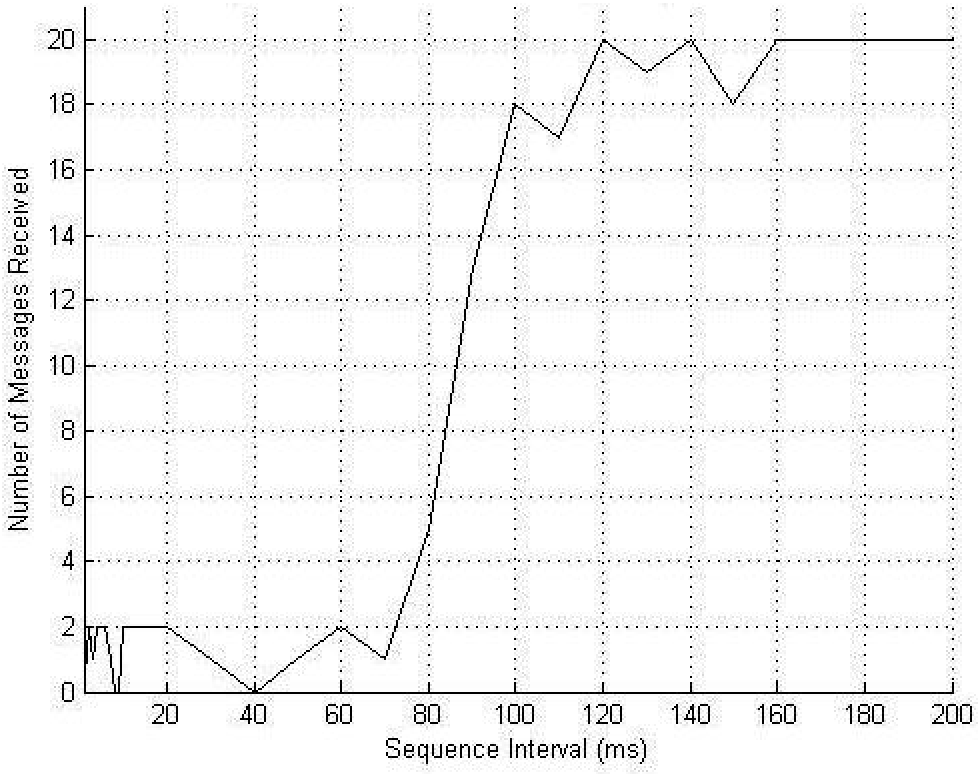

Figure 15. Results for transmission with 1 ms interval between packets

From the results obtained above, we can see that in order to achieve a reliable exchange for a transmission link with a single hop, there is a requirement for a delay between the transmissions of the data sequence. It can be ascertained from the results obtained that there should be at least 160 ms of delay between transmissions to achieve a relatively high level of reliability in the transmission link.

2. Examination of the Effects of Varying the Timing Interval between Packets

For this part of the experiment, the impact on the reliability of this transmission link due to the timing interval between packets was investigated. This objective was achieved by transmitting a predetermined data sequence over the established link with variations in the timing interval between packets for each run. For this run, to investigate the impact that the packet inter-arrival time has on the link, the interval between sequences was fixed at 1 sec, which was shown to be sufficient for a reliable link in the previous experiment. Data sequences of one packet in length were transmitted across the link, and the results obtained are tabulated in Table 4 and plotted in Figure 16.

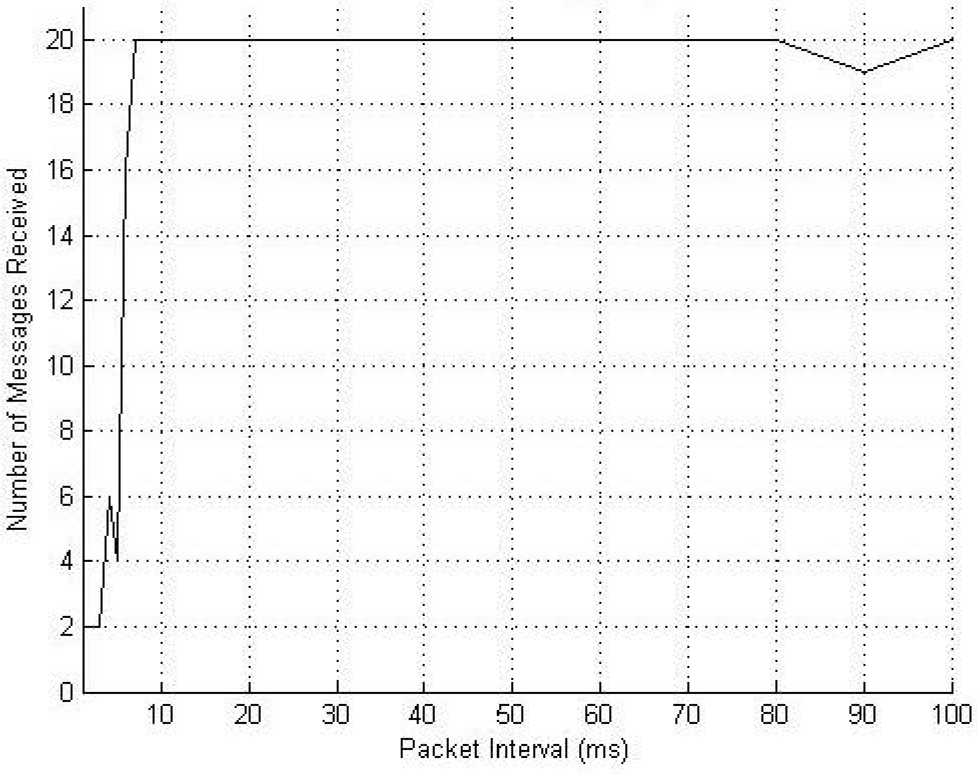

Figure 16. Experiment II – 1 sec delay between sequences

It can be observed from Figure 16 that reliability of this link is relatively high, with all messages correctly recovered in several runs, and the maximum number of messages not recovered correctly is 3 messages. This result infers that a high reliability of a single hop link is achievable if sufficient delay is incorporated into the transmission of the data sequence.

Run Number	Test message	Interval Timing (ms)	No of packets transmitted for each test message	No of times transmitted	Total number of packets transmitted	Total number of packets received	Number of messages correctly reconstructed
1	This is a test message	1	4	20	80	23	2
2	This is a test message	2	4	20	80	20	2
3	This is a test message	3	4	20	80	21	2
4	This is a test message	4	4	20	80	23	6
5	This is a test message	5	4	20	80	30	4
6	This is a test message	6	4	20	80	36	16
7	This is a test message	7	4	20	80	40	20
8	This is a test message	8	4	20	80	40	20
9	This is a test message	9	4	20	80	40	20
10	This is a test message	10	4	20	80	40	20
11	This is a test message	20	4	20	80	40	20
12	This is a test message	30	4	20	80	40	20
13	This is a test message	40	4	20	80	40	20
14	This is a test message	50	4	20	80	40	20
15	This is a test message	60	4	20	80	40	20
16	This is a test message	70	4	20	80	40	20
17	This is a test message	80	4	20	80	40	20
18	This is a test message	90	4	20	80	39	19
19	This is a test message	100	4	20	80	40	20

Table 4. Results from experiment for packet interval

From results obtained in Table 4, it can be observed that although the first three runs have the same number of messages recovered, each run has a different number of packets that are received by the base station. From this result, it can be inferred that the nature of the packets that have been dropped also plays a critical role in determining the performance of the transmission link. This observation is evident in the second run. Although 50% of the packets have been received, only a total of 2 messages have been correctly reconstructed. This is a case where majority of the signaling packets are being dropped, causing the reconstruction process at the receiver to be corrupted due to insufficient information. This results in 18 messages being unable to be reconstructed correctly.

As such, it is inferred that there is no strong correlation between the numbers of packets that were received compared to the number of messages that were correctly recovered.

With the results obtained from the above experiment, further experiments are again conducted to understand if the variation in the length of the data sequence has any impact on the link performance.

3. Examination of the Effects of Variable Length Data

This part of the experiment investigates the reliability of the link with a single mote being employed to extend the range of the transmitter, when data sequences of variable length are being transmitted. Test sequences comprising of different data length were transmitted across this link, and the results of the transmission recorded and plotted in Figure 17 of the following page.

To ensure that this test was conducted with a link that was relatively reliable to begin with, parameters for both the interval between the test sequence transmission and packet transmissions were chosen from the previous experimental runs. The parameters that allowed for the most reliable transmission to be achieved were used. Therefore, for this run, the interval between test sequences was set to 1 sec and the interval between packet transmissions was set to 10 ms.

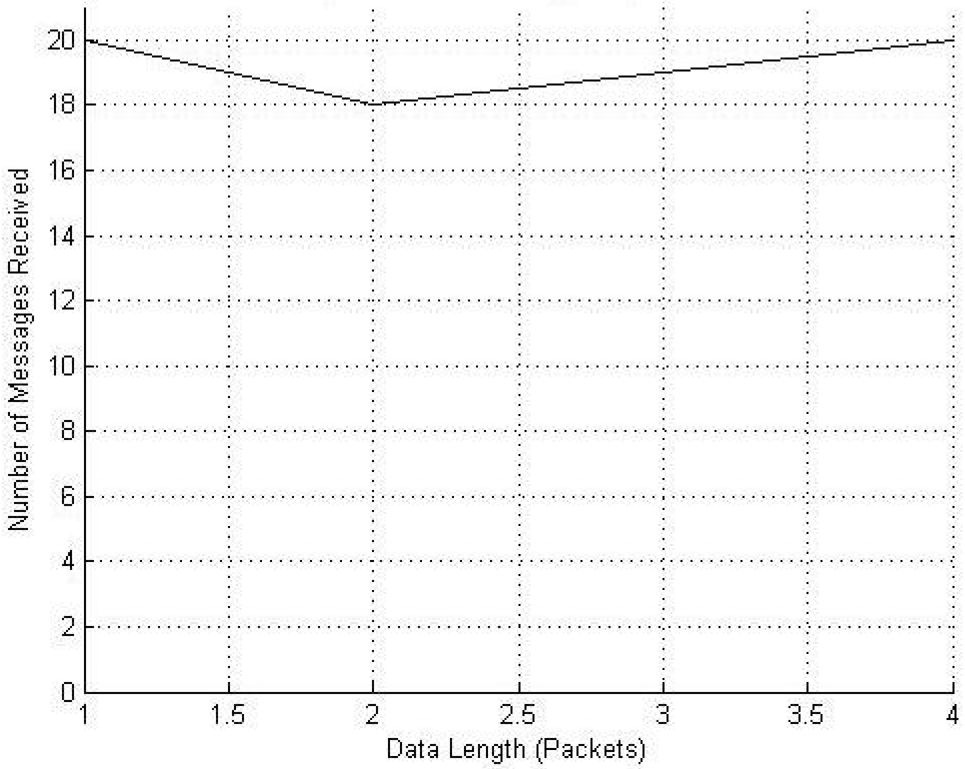

Figure 17. Experiment III – Variable data length

Results obtained from the experiment conducted showed that there was minimal loss in link reliability with the variation in the length of the data that is being transmitted. For all 4 separate runs, the number of messages that was correctly recovered is consistent with the number of messages reconstructed correctly during steady state in the experiment conducted previously.

From the above three experiments, it is clear that reliability of this transmission link lies in the interval timing between transmission for both the data sequence and also the packets and that there is no correlation between the reliability of this link and the length of the data that is being transmitted.

D. TRANSMISSION WITH TWO HOP

It was understood that in the wireless sensor network environment, it is unlikely that connectivity could be achieved through a single hop. More often than not, multiple hops have to be used before a link between terminal stations can be established. This consideration makes it a requirement to conduct an experiment to establish the capabilities and limitations for a link established based on two hops.

This section of the thesis will look into the experiments that have been conducted with two motes in between two terminal stations to allow the capability and limitation with such a link to be further investigated. The configuration of this experimental setup is as shown in Figure 18.

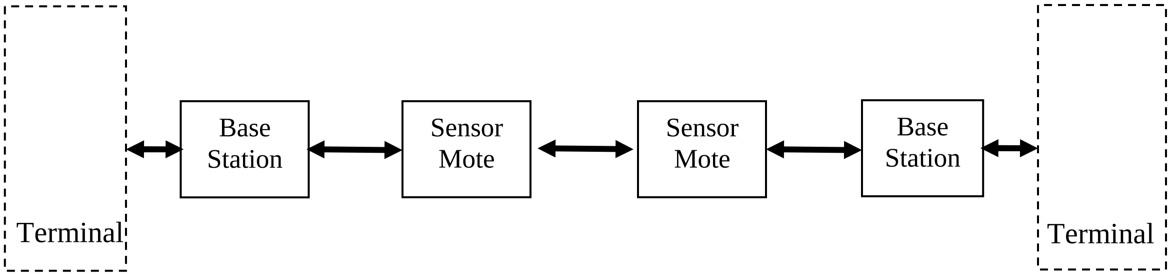

Figure 18. Setup with two motes to establish a link

1. Examination of the Effects of Varying Timing between Data Sequences

For this part of the experiment, the interval between packets was fixed at 1 ms and details of the test sequence that were transmitted are shown in Table 5.

Data Length	Test message	No of packets transmitted for each test message	No of times transmitted
1 packet	This is a test message	4	20

Table 5. Test Sequence transmitted in Double Hop experiment

The results obtained from this experiment are shown in Figure 19. In comparison to the previous experiments, it can be seen that increasing the number of hops reduces the reliability of the transmission link. In comparison to the previous two experiments, steady state for the number of messages correctly reconstructed is at around 15 messages per 20 messages transmitted. This is approximately 75% of the messages are correctly received.

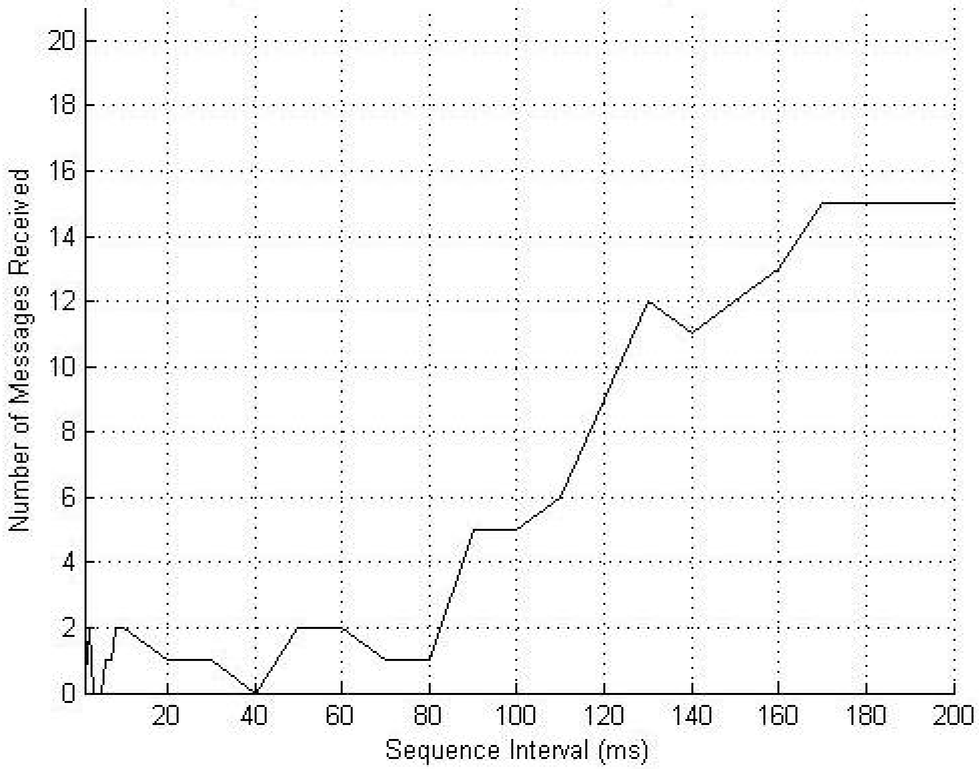

Figure 19. Results from experiment on sequence interval for Double hop

From this result, it can be inferred that the error control protocol that has been implemented is only robust for links established without any hops or up to a single hop. The protocol is not robust enough to develop a reliable link for two hops and above. To establish a more reliable link, a better error control protocol has to be established.

2. Examination of the Effects of Varying the Timing Interval between Packets

Using the same test sequence as shown in Table 5, the experiment was conducted with the interval between sequence transmission being set at 1 sec. Results from the experiment were consolidated and plotted in Figure 20.

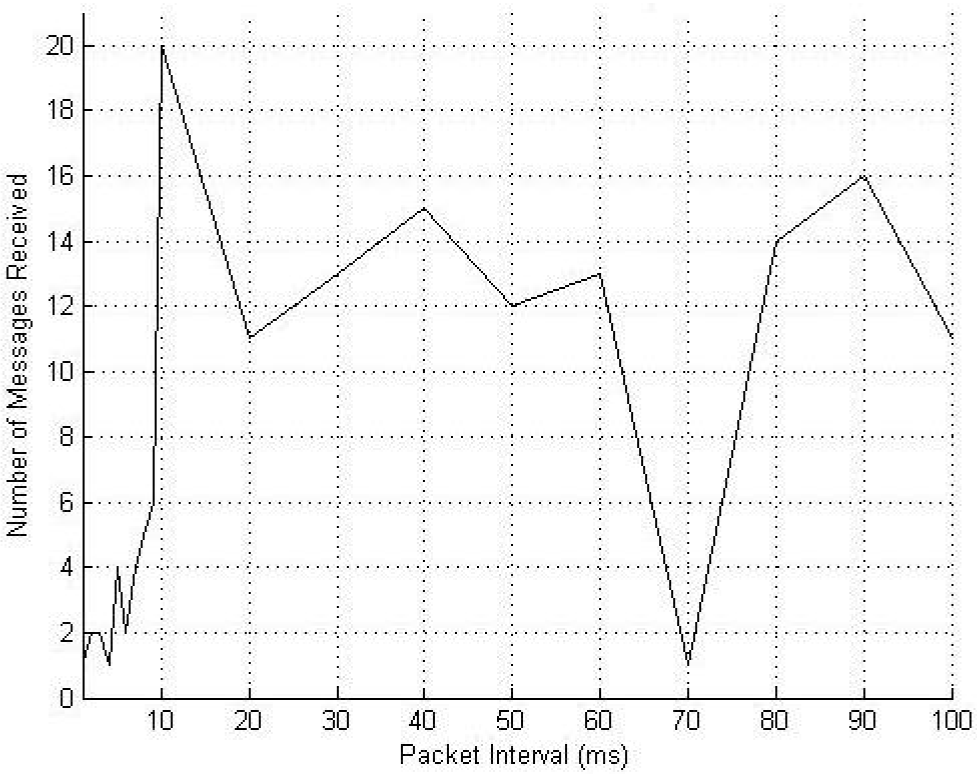

Figure 20. Results from experiment on packet interval for double hop link

Consistent with the conclusion obtained from the previous experiment, the reliability of the link degenerated with the increase of another mote as a repeater station. This can be seen clearly from the results obtained in Figure 20 where it can be observed that the inclusion of the extra mote causes the reliability of the link to fluctuate.

This result further validates the point that a more robust error control protocol has to be developed and implemented within this communications link in order for this link to be deployable in an actual operational environment.

3. Examination of the Effects of Variable Length Data

This part of the experiment investigates the reliability of the link with two motes being employed to extend the range of the transmitter, when data sequences of variable length are transmitted. Test sequences comprising different data lengths are transmitted across this link, and the results of the transmission recorded and plotted in Figure 17.

To ensure that this test was conducted with a link that was relatively reliable to begin with, parameters for both the interval between test sequence transmissions and packet transmissions were chosen from the previous experimental runs. It can be observed from previous runs that best results obtained for a double hop is around 75% message recovery. Therefore, for this run, the interval between test sequences was set to 1 sec and the interval between packet transmissions was set to 20 ms.

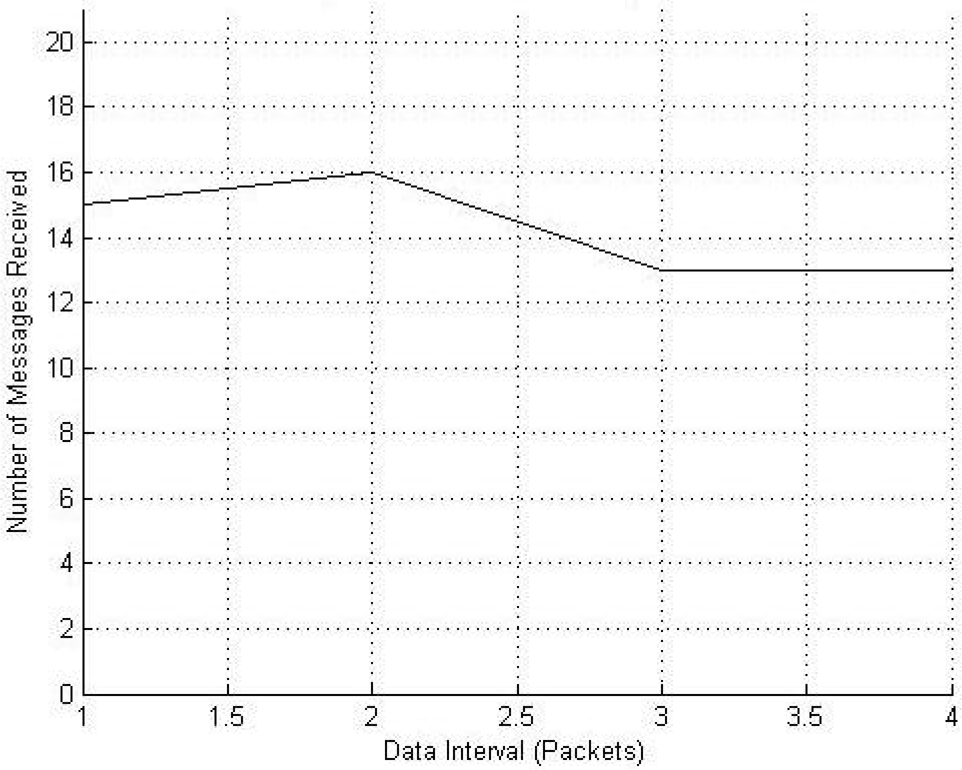

Figure 21. Results from experiment on data length variation on double hop

Consistent with the results that were obtained from the previous experiments; it was observed that the number of messages that were correctly recovered is similar to the number of messages that were correctly recovered in the steady state for previous experiments with variations in sequence interval and packet interval.

This result further validates the conclusion that the reliability of this link is determined by the sequence interval and the packet interval as opposed to variations in the length of the data that is being transmitted.

E. ANALYSIS AND DISCUSSIONS OF RESULTS OBTAINED

This section discusses in detail the results that were obtained from the previous experiments and what capabilities and limitations can be inferred from the results.

1. Variation of Sequence Interval

In this part, the effects on varying the interval between data sequences will be studied in detail. Results plotted in Figure 22 indicate that there is indeed a level of correlation between the numbers of messages that can be correctly reconstructed against the number of hops to establish the transmission link.

It is clearly depicted in the plot that, for the same amount of interval between sequences, the number of messages correctly reconstructed by a base station to base station link is higher than messages correctly reconstructed by a single hop and a double hop transmission. As the number of hops required to establish a link increases, the necessary interval between sequences for successful transmission increases.

The primary reason for the degradation in the number of messages recovered with the increase in the number of hops required is due to collision and dropping of packets. With a base station to base station transmission link configuration as shown in Figure 11, there is no additional processing required to be performed in the midst of transmission and effectively, all packets that have been transmitted will be able to reach the receiver. As a signaling packet requires more computational time as compared to a data packet at the receiver end, it can be concluded from here that for a base to base link, the dominating factor that affects the performance of the link is the sequence interval.

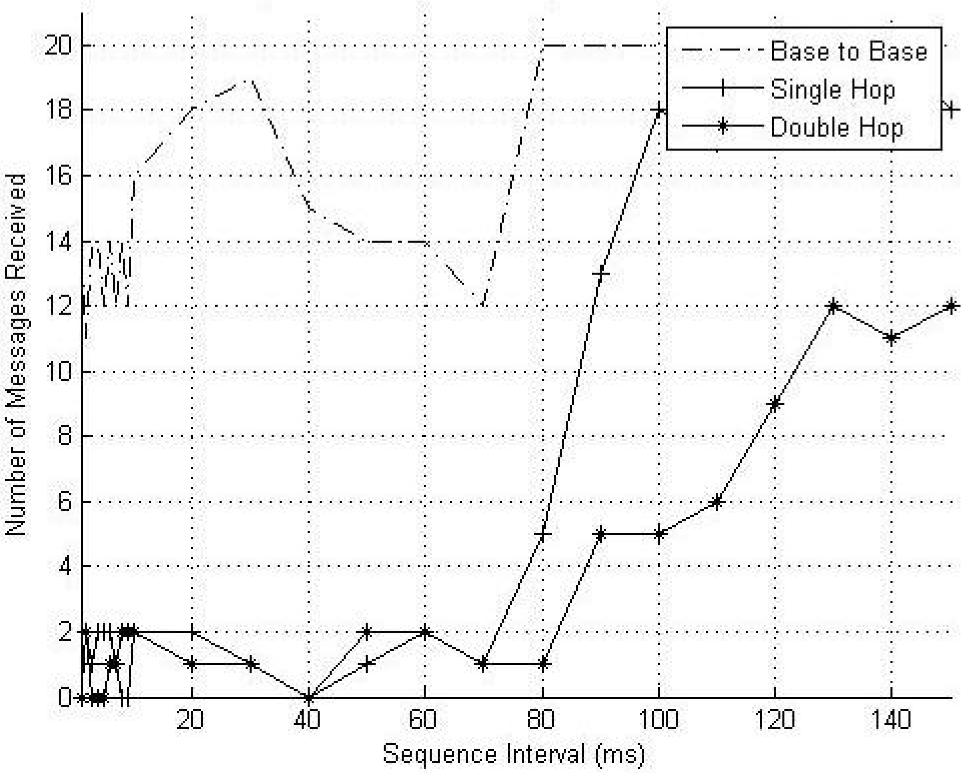

Figure 22. Comparison of Sequence Interval Variation for all three modes

The inclusion of a single mote as shown in Figure 14 implies that additional processing tasks will have to be performed by the mote.

It was established in earlier discussion that the task of a sensor mote in this link is to forward all newly received packets, and to drop all repeated packets. Performing these tasks requires processing time on the part of the motes and hence, a longer delay between sequences transmissions would allow for the required processes to be completed before the mote is ready for another packet. Consistent with the results that were obtained, the sequence interval is higher for a transmission link established on single mote as compared to a transmission link established through a base station to base station transmission.

Experimental runs with a transmission link established through a double hop further validate the conclusions above. With reference to Figure 22, it can be seen that the performance of the link is greatly degenerated with the double hop transmission link. The main reason is due to the existence of a region that crosses over both repeating motes, as shown in Figure 23.

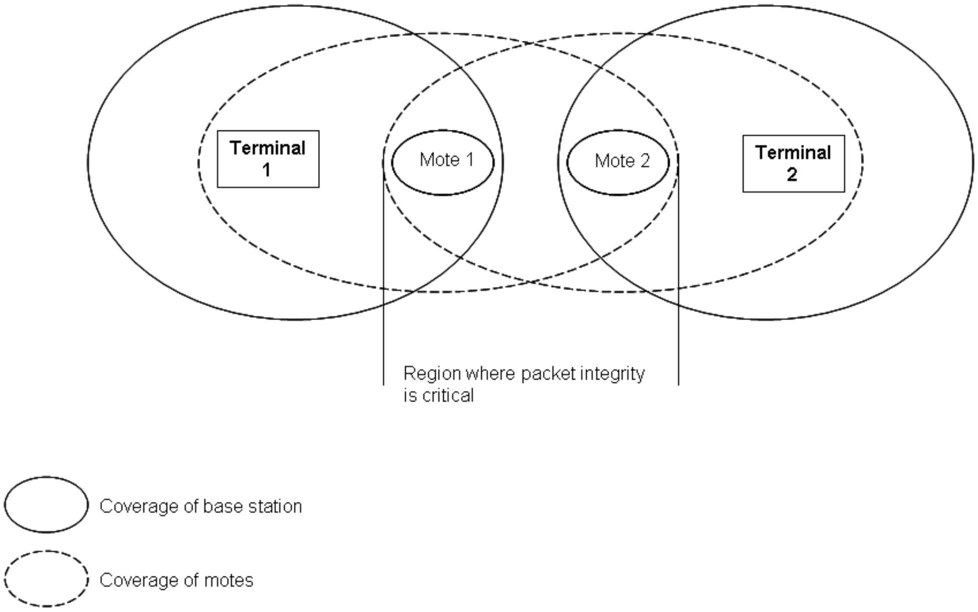

Figure 23. Area of coverage for link with double hops

It is shown above that with both motes clearly within range of each other, Mote 1 will be able to pick up messages that have been transmitted by Mote 2. Due to the basic operating nature of a repeater mote, it is anticipated that packet traffic will be extremely heavy in this region.

Coupled with the busy traffic in general, the number of packets returning to Mote 1 will be increased as well. The reason is that Mote 1 will receive and process redundant packets that have been broadcast by Mote 2 to Terminal 2 before dropping the packets after verifying that they are repeated packets.

This processing takes time and resources from Mote 1, leaving it with a tighter window for actual packets to be received and processed from Terminal 1. As such, it is inevitable that the sequence interval has to be increased for a double hop transmission so as to achieve a relatively reliable transmission link.

From this analysis, it can be inferred that the sequence interval has a direct correlation with the number of hops required to establish the transmission link. As the number of hops increases, the sequence interval will have to increase as well.

2. Variation of Packet Interval

This part of the experiment is conducted with a sequence interval timing that generates an unreliable link. The motivation for this run is to investigate if link reliability can be enhanced by increasing packet interval. Therefore, sequence intervals for both runs were set to 1 ms.

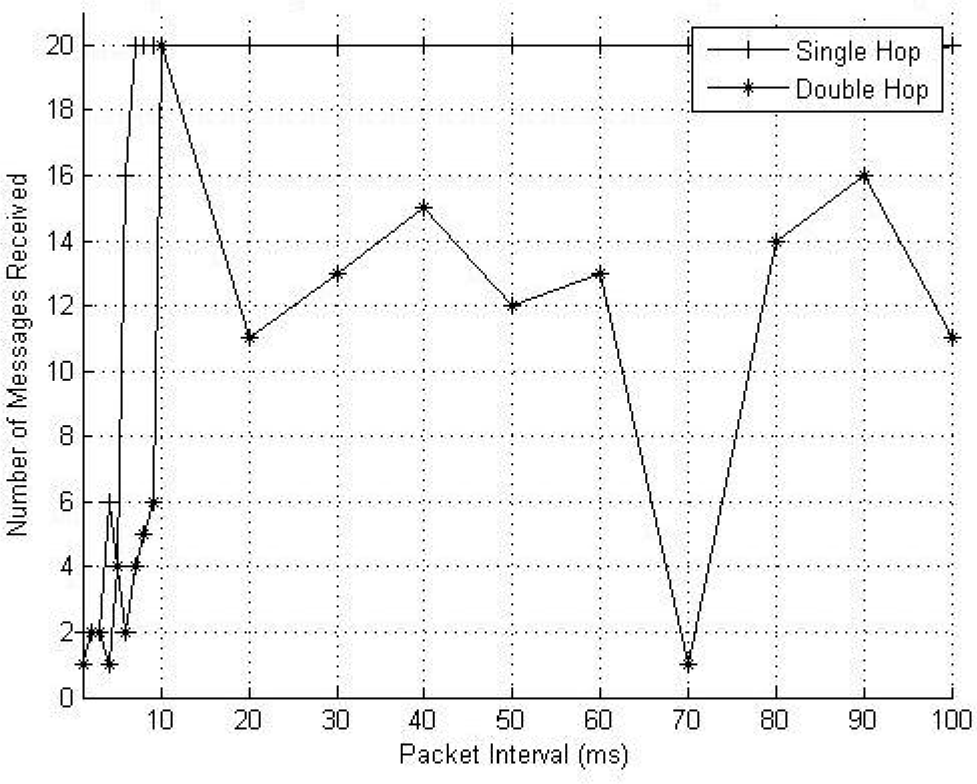

Figure 24. Comparison of Packet Interval Variation for single and double hop

Results were obtained from runs with the timing interval between packet transmissions being varied. Although link performance is fluctuating at lower packet interval, link reliability for a single hop link only begins to stabilize as packet interval increases to approximately 10 ms. However, link reliability for a double hop link is still fluctuating even when packet interval reaches 100 ms. The reason why a single link is able to achieve higher message recovery with a faster packet interval is due to the error control protocol that was implemented, which will be discussed further in the chapter.

In addition, observation on the number of messages being correctly reconstructed reflects that for the link with the single hop, the number of messages recovered was relatively higher as compared to the link with a double hop. The reason for this result being that as compared to a single hop is packets in a double hop link have a higher probability of being dropped due to collision or congestion.

It has been identified previously that with the establishment of another repeater mote, the processing requirement for both motes increases substantially and to handle the high traffic demand reliably, a longer delay has to be provided for the motes. Due to this circumstance, as the number of motes employed to establish the link increases, the interval between packets will also have to increase accordingly.

In addition, it has been observed that total number of messages correctly reconstructed also decreases as the number of hops required increases. The probability of packets being dropped at the motes contributed to this decrease in messages recovered. As the packets are being transmitted in a sequential manner, all packets are vital for the reconstruction of the message.

One observation made from Figure 22 is, as compared to a base station to base station transmission, that the single hop link requires a longer sequence interval to achieve a steady state performance of almost 100% message recovery. As such, an increase in the number of hops in a transmission link leads to an increase in the sequence interval required and a reduction in the number of messages reconstructed at the receiver.

However, one interesting observation made from results obtained in Figure 22 and Figure 24 is that for single hop link, a higher level of reliability can be achieved by varying the packet interval instead of the sequence interval. However, this improvement is not reflected in the results obtained in a double hop link. This behavior is due to the error control protocol that has been implemented in this version of SNAIL.

As discussed previously in Chapter III, a basic form of error control is achieved through transmission of redundant packets from the transmitter with the intent that there is a second packet to backup the first in the event that the first packet was dropped.

To understand the limitation of the error control protocol that has been implemented, there is a need to look into the rate of packets that are propagated at different stages of the transmission for a base station to base station link can be illustrated by Figure 25.

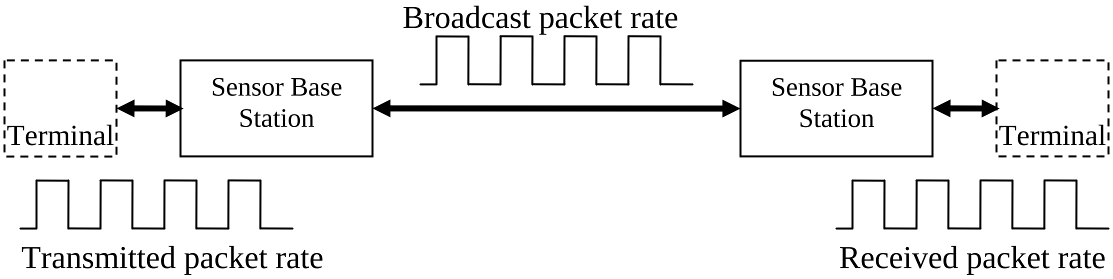

Figure 25. Packet transmission rate at different stages for Base station to Base station link

From the diagram above, it can be concluded that rate of packet transmission will be consistent for the entire link. Therefore, variation of the sequence interval will have a direct impact on the performance of the link.

The rate of packets that is being propagated at different stages of the transmission for a transmission link established with a single hop can be represented by Figure 26.

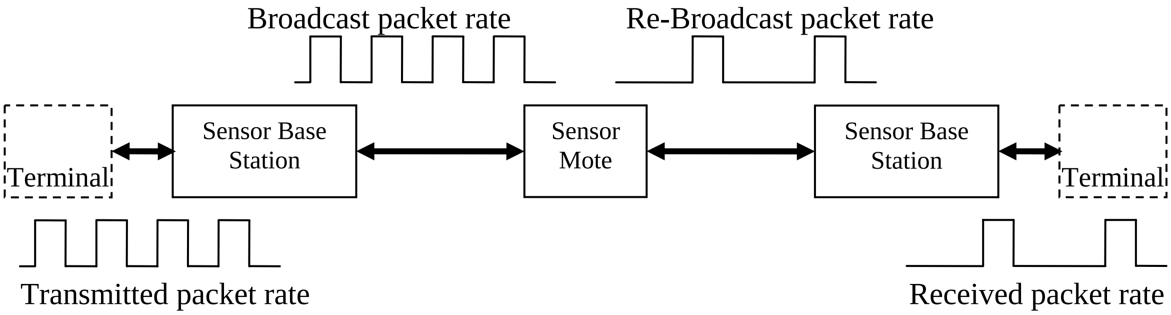

Figure 26. Packet transmission rate at different stages for single hop link

The error control protocol implemented in this link requires all packets to be transmitted twice before the next packet can be transmitted. At the repeater mote, the sequence number of all packets is verified such that only new packets will be repeated. Due to this check, half of the packets that have been broadcast by the transmitter will be dropped as it will be considered a redundant packet.

Consequently, this explains why the rate of packets being broadcast from the sensor mote is half the rate of packets arriving at the sensor mote. Due to this increase in the interval that each packet is being repeated by the sensor mote, an inherent delay is generated, allowing more time for the receiver to receive and process each packet. This explains why a shorter packet interval at the transmitter is required for the link with a single hop to reach its peak performance as compared to the link established with a base station to base station.

From Figure 27, it can be established that there will not be any change in the rate of transmission for the packets after Mote 1 as all redundant packets have been dropped by Mote 1 and all packets that reach Mote 2 will be processed as new packets.

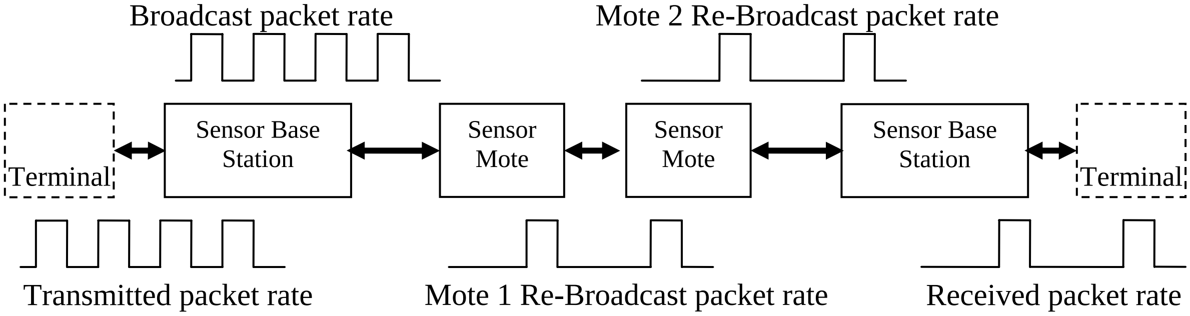

Figure 27. Packet transmission rate at different stages for double hop link

However, it was concluded from all experiments conducted that the performance of a link established with two hops was far inferior to the performance of a single hop. The reason for the inferior performance is due to the fact that redundant packets which act as a form of error control have all been dropped at Mote 1. As such, the error control protocol that has been implemented to enhance message recovery at the receiver end was effectively removed by the function of the motes of the first hop, resulting in the link after the first hop with no error control protocol.

As such, performance of the link between the first hop to the receiver station is purely dependent on the robustness of the link itself and without a contingency plan for dropped packets. Due to this weakness, as the number of motes increases, the probability of messages being unrecoverable due to packets being dropped increases.

Finally, from the results obtained, it can be established that by varying the transmission interval, performance of the link is able to be increased to a certain level, at approximately 95% reliability for a link with single hop and approximately 75% for a link with a double hop. To improve the performance beyond this, a more robust error control protocol has to be implemented.

3. Variation on Length of Data Transmitted

The motivation to conduct this experiment was to investigate the possibility of a correlation between the performance of the transmission link and the length of data that was being transmitted through this link. To ensure that the results obtained were not biased with unreliability induced from an insufficient sequence interval or packet interval, interval timing was selected to allow for a relatively consistent reliability from the link.

Therefore, sequence intervals for both runs were set to 1 sec. As noted from previous runs, these parameters would provide sufficient delay for the repeater motes to process the previous transmitted sequence before a new sequence arrives.

It is arguable whether increasing the length of the data transmission would lead to an increase in the number of packets required for the transmission of the data length, leading to a decrease in the performance of the link. However, it is noted that an increase in number of packets required to send a data sequence across would mean that there are more packets that belong to a single data sequence being transmitted across this link. As such, this would lead to an increase in the probability of packets belonging to the same data sequence to be corrupted or dropped.

Therefore, although the number of packets that could be corrupted or dropped would increase with the increase in the number of packets required to be transmitted, corrupting packets belonging to the same data sequence will still cause a single data sequence to be unrecoverable. Corrupted packets belonging to the same data sequence will not affect the message recovery of the subsequent data sequences.

This conclusion is supported from the results plotted in Figure 28. It can be observed from this plot, that the number of messages that have been correctly reconstructed is consistent with the results that were obtained from the experiments conducted previously, regardless of the length of data sequence that were transmitted.

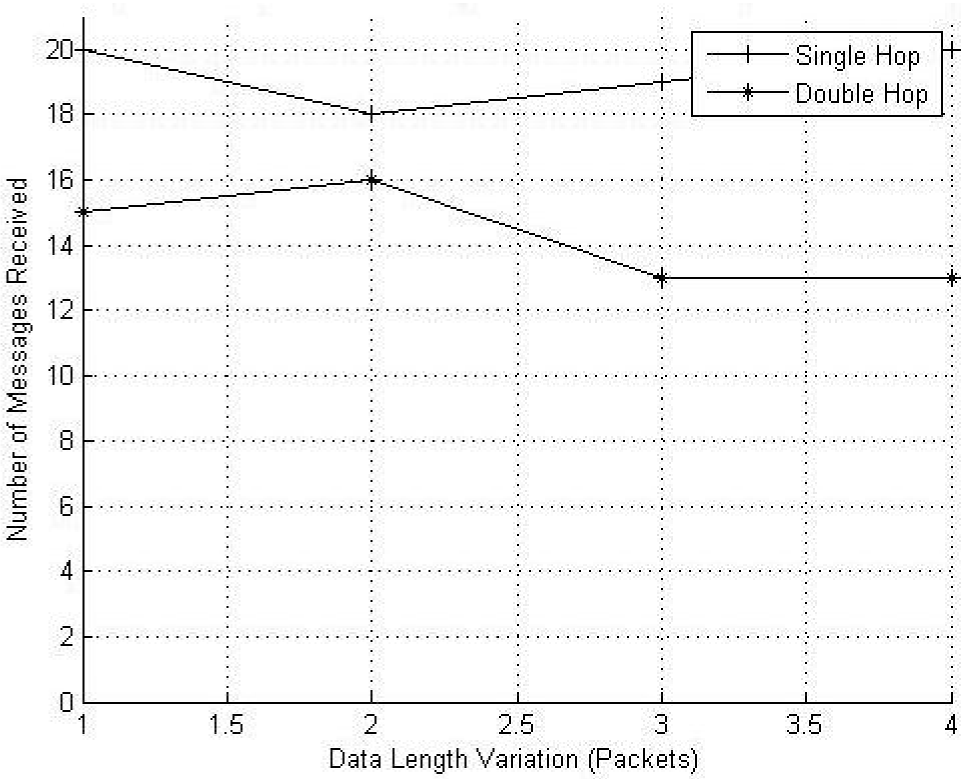

Figure 28. Comparison for Packet Interval Variation for single and double hop

This shows that there is little correlation between the lengths of the data to be transmitted and the messages that are being recovered. As such, it further reinforces the conclusion that one of the determining factor for the performance of the link lies with the interval between sequence and packet transmission.

In summary, it was established, from all results obtained, that the main determining factor of the link performance lies with the integrity of the signaling packet, which is the first packet transmitted across the link. Since it contains vital information required for the reconstruction of the packet, a loss of the signaling packet would mean that the message will be unrecoverable, even if all subsequent packets have been correctly received.

F. PROPOSED VARIATION TO ERROR CONTROL PROTOCOL

It was established from the discussions above that there is a need to reconsider the error control protocol that has been implemented. As each mote will only repeat a new packet once, there will be an absence of error control protocol once the packet leaves the first mote as represented in Figure 27.

Understanding of this limitation allows a variation of techniques that can be employed to overcome it. One such technique is to program the repeater motes to repeat each new packet twice instead of only once as implemented currently, while maintaining the ability to drop redundant packets received. This will allow the protocol to be maintained throughout the link. A representation the new error control protocol is depicted in Figure 29.

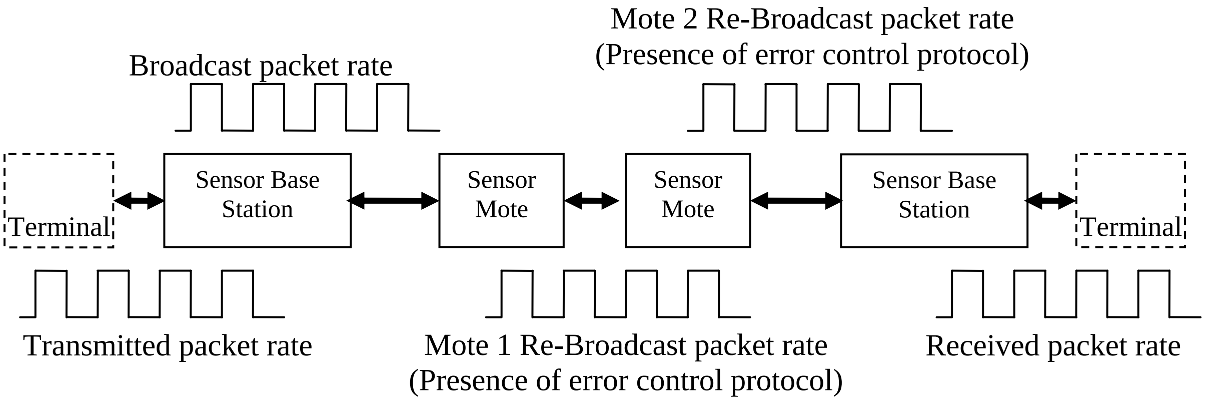

Figure 29. Packet flow representation of the new error control protocol

With the new protocol implemented in the motes, experiments were conducted once again with results obtained from the new protocol compared with the results that were obtained from the original error control protocol.

1. Comparison of Results for Variations in Sequence Interval for Single Hop

The performance of a single hop link was compared in this section of the thesis. Similar to the previous experiments, the packet interval was set to 1 ms with the sequence interval being varied over a range of timing, up to 200 ms. The number of messages that was correctly recovered by implementing the enhanced error control protocol was obtained and recorded.

Performance of the link under the different protocol was plotted and compared in Figure 30.

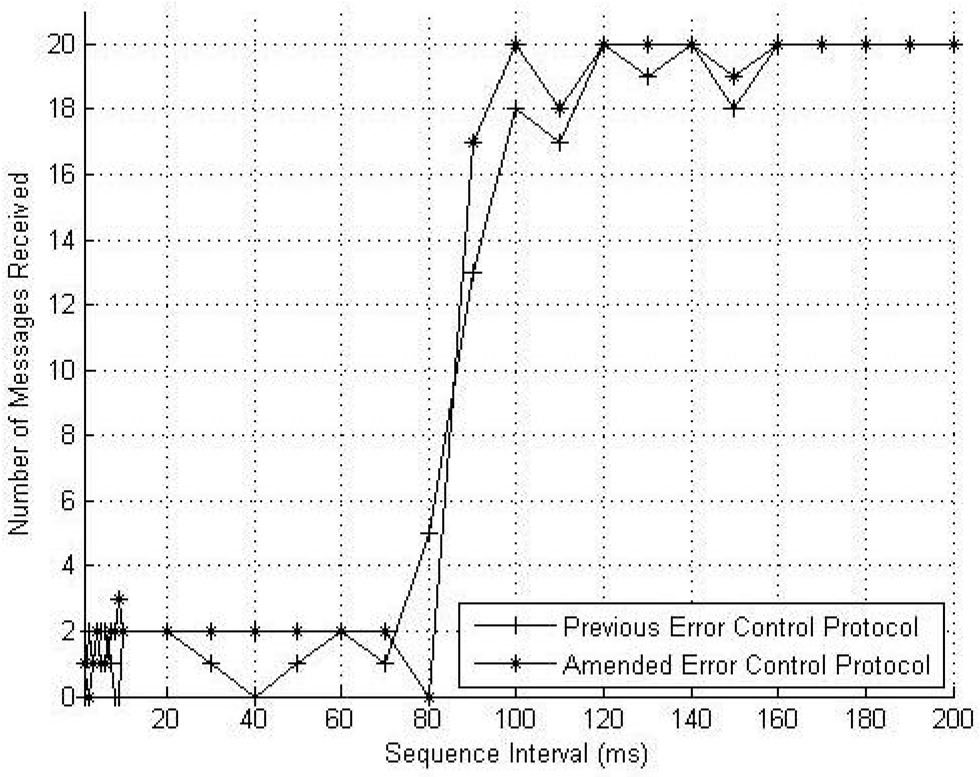

Figure 30. Impact from different error control protocol by varying sequence interval for single hop

It was observed from Figure 30 that link performance is almost similar in terms of sequence interval required to achieve a reliable link for transmission. Therefore, it can be concluded from this result that the amended error control protocol does not provide significant improvement to the link reliability as compared to the previous error control protocol.

2. Comparison of Results for Variation in Sequence Interval for a Double Hop

For a link established through a double hop, the same experiment was conducted and the results plotted in Figure 31.

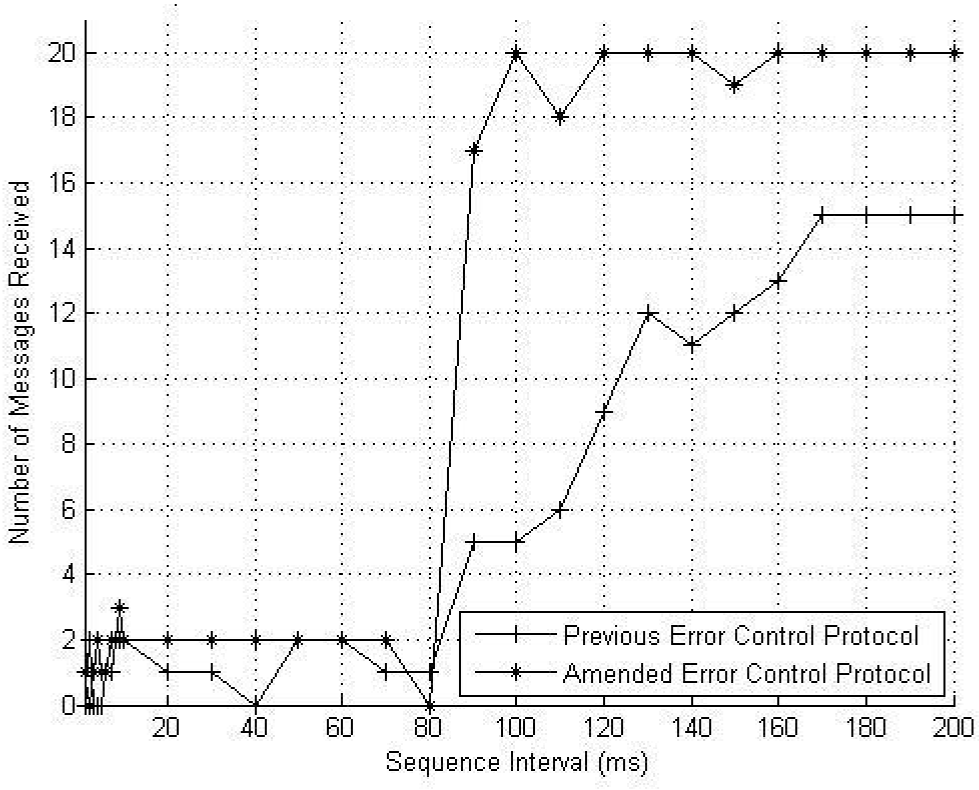

Figure 31. Impact from different error control protocol by varying sequence interval for double hop

Contrary to the results obtained for a single hop link, it was observed that a shorter sequence interval is required for the link to achieve its peak performance. In addition, implementation of the enhanced error control protocol allows a double hop link to achieve a peak performance of 100% message recovery after 120 ms of sequence interval, as compared to the peak performance of approximately 75% message recovery achieved with the previous protocol. It is clear that the enhanced protocol performs better in terms of error control as compared to the previous protocol.

3. Comparison of Results for Variation in Packet Interval for Single Hop

This section compares the performance of a single hop link using both error control protocols. Both protocols were implemented in the same link in two separate experimental runs, and the results obtained were plotted in Figure 32.

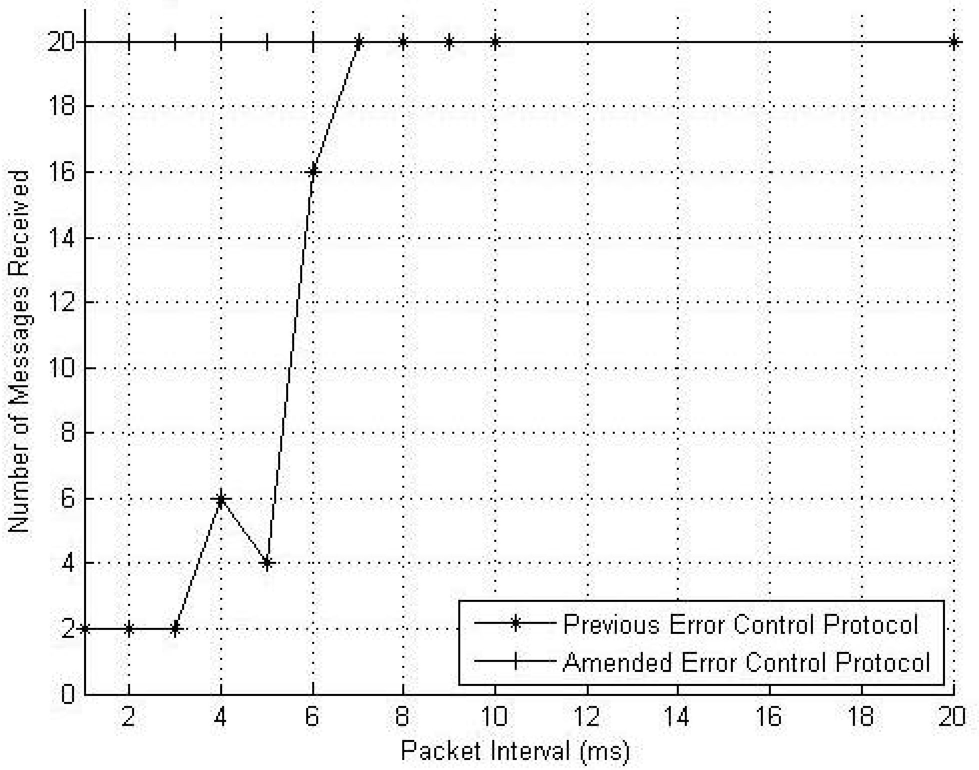

Figure 32. Impact from different error control protocols by varying packet interval for a single hop

From the graph above, it can be inferred that amended error control protocol is able to achieve a stable link with a lower packet interval. However, with slightly longer packet interval, both error control protocols are able to provide a relatively high message recovery rate for a single hop link.

4. Comparison of Results for Variation in Packet Interval for Double Hop

This section compares the performance of a double hop link that implements the previous error control protocol against the performance of a double hop link with the new error control protocol implemented. To be consistent with the parameters set for the previous run, the sequence interval for this run was also set to 1 sec, while the packet interval was varied. Results from both runs are represented in Figure 33.

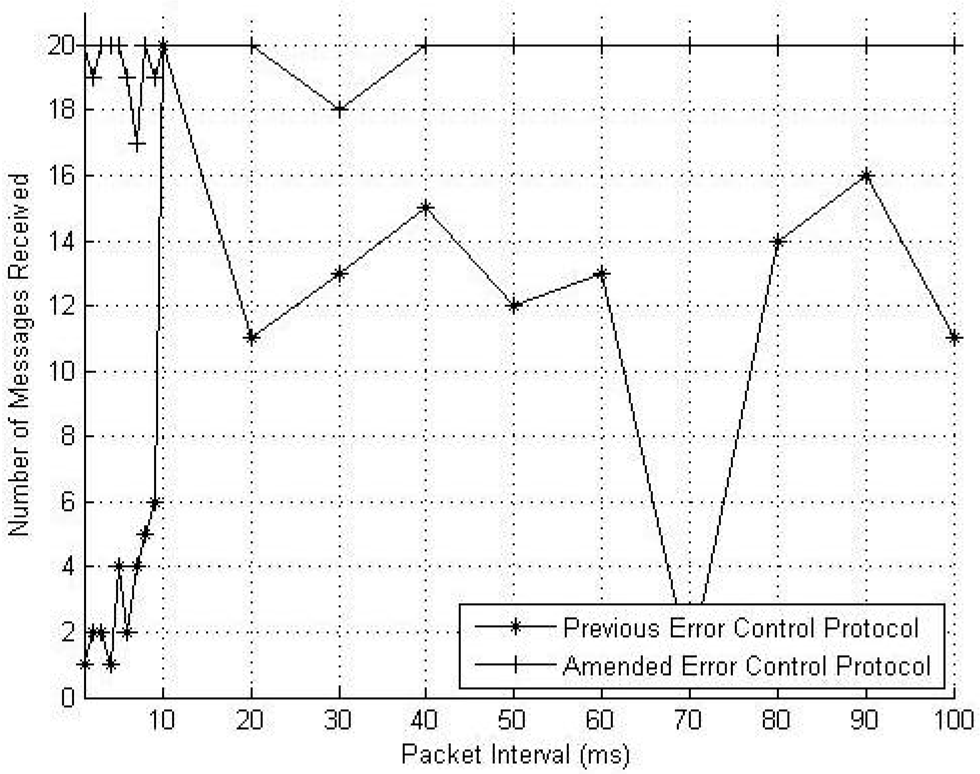

Figure 33. Impact from different error control protocol by varying packet interval for double hop

It can be observed that there is a significant increase in the performance of the link when the new protocol was implemented to provide error control. Although there is still a low level of instability at the early stages of the experiment, performance of the link reaches a steady state fairly quickly in comparison to the link implementing the previous protocol.

In addition, it can also be observed that the new protocol is able to achieve almost 100% message recovery in its steady state as compared to an inconsistent link performance with the previous protocol.

G. ANALYSIS OF ENHANCED ERROR CONTROL PROTOCOL

From all results obtained in the previous section, it is clear that better link performance can be achieved through the implementation of the enhanced error control protocol.

Although there are no significant improvements in performance for the single hop link, a significant improvement in the performance of a double hop link was observed from all comparisons made. There is a jump from an inconsistent link performance under the previous error control protocol to almost 100% message recovery under the enhanced error control protocol.

In addition, under the previous protocol, it was observed that link performance degenerates significantly when the number of hops required increases. However, as shown in Figure 30 and Figure 31, performance was consistent for both single hop and double hop links when the enhanced protocol was implemented. Although there is no significant advantage for the amended protocol in the single hop link, significant improvement in the performance for links with multiple hops was being able to be achieved with the amended protocol. As such, it was concluded that the enhanced error control protocol is a better protocol to be implemented.

From the results obtained above, it was concluded that the performance of the link is highly dependent on the recovery of the signaling packet. Sufficient time has to be provided for the signaling packet to be recovered fully for reliable transmission. This observation was made considering the results obtained from the experiment conducted for the double hop link where the sequence interval was being varied. It can be seen that at a lower sequence interval, where the signaling packet has a higher probability of being corrupted, the message recovery rate was at its minimal. However, upon reaching a certain level of delay, the probability of a signaling packet getting corrupted is reduced significantly and the performance of the link increases dramatically.

This observation is further substantiated with results obtained from the experiments that vary the packet interval. With a sufficient delay in the packet interval, there would be longer time for the processor to complete the last packet of the current sequence before the signaling packet of the next sequence arrives. From this observation, it can be concluded that reliable recovery of the signaling packet is essential for the transmission link to be reliable for data transmission.

H. SUMMARY

In this chapter, parameters such as sequence interval, packet interval and variable packet length were being analyzed on how it will affect the link performance. Several experiments were conducted, with each of the parameters being varied over a range of values until a relatively reliable link performance is achieved.

The results obtained from the various experiments were presented and analyzed. By analyzing the results obtained, it is shown that the reliability of the link is highly dependent on the recovery of the signaling packet, which contains the necessary information for the receiver to correctly reconstruct the data sequence.

In addition, analysis of the results obtained from both single hop and double hop reflected that there is a need to enhance the error control protocol implemented as the initial error control protocol is insufficient for transmission link with multiple hops. The amended protocol does not have a significant improvement for links with a single hop, but significant improvement in transmission link performance can be observed in links with two hops.

In the next chapter, the results will be summarized, and recommendations for future work will be provided.

V. CONCLUSION

A. CONCLUSION

Wireless communication has been at the forefront of technological advancement in recent years. With the introduction of WiFi and WIMAX, reliance on wireless access for either military or commercial applications has never been greater. However, all wireless communication links are only as strong as the weakest link; communication will not be possible once an individual receiver station is physically out of range of the transmitting station. Although there has been considerable research done on the techniques that can be employed within the wireless communication architecture to overcome this limitation of all wireless communication, the technique most common to all is the deployment of repeater stations within the area of interest to bridge any break in communication link due to coverage problems.

The problem with deployment of a repeater station lies with the size and power consumption of the station to allow for the transmission link to be established. This thesis studies the possibility of using wireless sensor motes to function as repeater stations by reconfiguring the capability of the motes.

Drawing similarities from ATM-AAL 5 [12], a data sequence of length longer than a standard payload will be segmented into smaller sized packets of 27 bytes long to facilitate transmission, with the last segmented packet being padded with ‘0’s to make up to 27 bytes. Segmenting a data sequence into smaller packets will require reconstruction at the receiver end and information such as the total number of packets and the total number of padding bits employed have to be made known to the receiver for reconstruction to take place.

As such, it was concluded that there is a requirement for a signaling packet to be transmitted to the receiver to provide these necessary information. This signaling packet will be transmitted to the receiver as the first packet of the entire sequence so to allow the receiver to be prepared for the incoming transmission and to update the status of the transmission as the packets arrive.

There are several topologies that can be formed by a wireless sensor network. However, only a daisy chain configuration was implemented in this thesis to study the feasibility of the concept. Several experiments were conducted to quantify the capabilities of the transmission link created through sensor motes. Link performance was analyzed based on the number of messages that were transmitted against the total number of messages that could be correctly recovered at the receiver. Parameters such as the interval between message sequences and the interval between packets transmitted were varied to further understand the influence delay has on the performance of the link.

This thesis introduced two error control protocols that were designed and implemented. Experiments were conducted and the results achieved from these two protocols were collected and compared. It was observed that the initial error control protocol requires a shorter message sequence interval in a single hop link to achieve its peak performance. However, the limitation of the initial protocol was a significant drop in performance with the increase in the number of hops required in the transmission link. This limitation contradicts the principle of scalability within a wireless sensor network.

The implementation of the enhanced error control protocol required a longer message sequence interval to achieve its peak performance. However, the result achieved from a single-hop link was consistent with the results that have been achieved with a double-hop link. This concludes that the enhanced error control protocol is able to provide a reliable, yet scalable tunneled transmission link in a wireless sensor network.

Observations made from experiments conducted for both protocols concluded that recovery of the signaling packet is critical to ensure the reliability and performance of the transmission link. To establish a reliable link for transmission, the parameters set for the link must ensure that the signaling packet can be transmitted and correctly received by the receiver. Failure to recover this packet will lead to a failure to recover the message transmitted.

B. FUTURE WORK

This study only covers the concept of using a wireless sensor network to perform tunneled data transmission. More work could be done in the future to explore establishing a tunneled path within a meshed network of motes. Some examples of future work are proposed in the following paragraphs.

1. Error Control Protocol

Currently, error control is achieved through redundancy. A single packet is repeated several times over the repeater motes, so as to ensure that the packets will be able to reach the designated station. The downside of this protocol is the requirement of a large amount of packets to be transmitted across a link. In addition, due to the high number of redundant packets that were being transmitted, the overall throughput of the link is also affected.

To implement a real-time application using this concept, there is a need to consider a variety of error control protocols. Specifically, a more efficient way to provide error control in the link should be established so that there can be a reduction in the amount of error control packets required without compromising the performance of the link.

2. Establishing a Tunnel within a Meshed Network

Given more time to study this topic, it would definitely be useful to look into the ways to establish a tunneled path within a wireless sensor network. Having established that the concept of tunneled data transmission is possible using sensor motes as a repeater station, the next step would be to look into ways of how a path can be established.

Having a protocol that is able to establish a tunnel within a wireless sensor network will definitely project wireless communication into the next era where connectivity of wireless communication will no longer be limited by deployment of bulky repeater stations, but enhanced by using wireless sensor motes.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Netorks," IEEE Communications Magazine, Volume 40, Issue 8, pp. 102 – 114, August 2002.

[2] Wang Wei Beng, "Analysis and Classification of Traffic in Wireless Sensor Networks," Master's Thesis, Naval Postgraduate School, Monterey, California, March 2007.

[3] University of Wisconsin-Madison, "Wireless Sensor Networks," Seapahn Megerian and Miodrag Potkonjak, http://www.ece.wisc.edu/~megerian/papers/sensor_nets.pdf , Last accessed 01 November 2007.

[4] Jumpot Phuritatkul and Tapio Erke, "An Investigation into Performance of Congestion Control Mechanisms in ATM-UBR Service for TCP Sources," p. 463-468, Ninth IEEE International Conference on Networks (ICON'01), Bangkok, Thailand, 10-12 October 2001.

[5] Tony Larsson and Nicklas Hedman, "Routing Protocols in Wireless Ad-hoc Networks – A Simulation Study," Master's Thesis, Lulea University of Technology, Stockholm, 1998.

[6] Arizona State Public Information Network, "Introduction to Wireless Networking," Greg Von Beck, http://aspin.asu.edu/projects/wireless/introduction.html, Last accessed 17 November 2007.

[7] Gaurav Jolly, Mustafa C. Kusçu, Pallavi Kokate and Mohamed Younis, "A Low-Energy Key Management Protocol for Wireless Sensor Networks," p. 335-340, Eighth IEEE Symposium on Computers and Communications, Antalya, Turkey, 30 June – 3 July 2003.

[8] Crossbow Technology Inc, "Smart Dust Training Seminar," San Jose, April 19-20, 2005.

[9] TinyOS: "TinyOS Mission Statement," University of California Berkeley, http://www.tinyos.net/special/mission, Last accessed 05 November 2007.

[10] B. Phillip, H. Jason and C. David, "Active Message Communication for Tiny Networked Sensors," submitted to IEEE INFOCOM, April 2001.

[11] Herbert Schildt *The Complete Reference Java* ^{TM} *2*, Fourth Edition, McGraw-Hill, Berkeley, California, 2001.

[12] William Stallings, “ Asynchronous Transfer Mode,” in *High-Speed Networks and Internets: Performance and Quality of Service*, Second Edition, pp 92-107, Prentice Hall, Upper Saddle River, New Jersey, 2002.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Department of Electrical and Computer Engineering, Code EC
   Naval Postgraduate School
   Monterey, California

4. Professor John C. McEachen, Code EC/Mj
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Murali Tummala, Code EC/Tu
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Yow, Thiam Poh
   Singapore Army
   Singapore