2007-03

Analysis and classification of traffic in wireless sensor networks

Wang, Beng Wei

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ANALYSIS AND CLASSIFICATION OF TRAFFIC IN WIRELESS SENSOR NETWORKS

by

Wang Wei Beng

March 2007

Thesis Advisor: John McEachen
Second Reader: Murali Tummala

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Analysis and Classification of Traffic in Wireless Sensor Networks		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Wang Wei Beng			
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** A	

13. ABSTRACT (maximum 200 words)

Wireless sensor networks have been widely researched for use in both military and commercial applications. They are especially of interest to the military planners as they can be deployed in hostile environments to collect vital information safely and cheaply. In view of this interest, there is a need to capture and categorize the data effectively under different operational conditions. This thesis captured traffic and data from sensor motes to analyze and present characteristics of the traffic in a meaningful manner.

Specifically, this thesis studied the traffic generated by wireless sensor networks by setting up two different commonly used network topologies, namely a direct connection to the base and a daisy-chain connection to base. A total of six runs of experiments were conducted, three for each topology. The data traffic between the nodes was captured over an extended duration of time. Using the captured information, analysis was performed to categorize and identify the information through anomalies and variations of traffic patterns. Data were also analyzed to study self-similarity and statistical distribution.

The experimental results have shown that it is possible to differentiate the two different topologies by monitoring the traffic distribution or by analyzing the types of messages sent. The status of the nodes can also be determined with the traffic collected. Examples include new nodes joining the network and operational status of the nodes. Statistical analysis has also been done and found that wireless sensor network traffic is not self-similar except for the interarrival time of the direct connection mode.

14. SUBJECT TERMS Wireless sensor networks, traffic analysis, self-similarity.			**15. NUMBER OF PAGES** 87
			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** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

ANALYSIS AND CLASSIFICATION OF TRAFFIC IN WIRELESS SENSOR NETWORKS

Wei Beng Wang
Civilian, Defence Science & Technology Agency (DSTA), Singapore
Bachelor of Engineering (Hons), University of Leicester, United Kingdom, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2007

Author: Wei Beng Wang

Approved by: John C. McEachen
Thesis Advisor

Murali Tummala
Second Reader

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

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Wireless sensor networks have been widely researched for use in both military and commercial applications. They are especially of interest to the military planners as they can be deployed in hostile environments to collect vital information safely and cheaply. In view of this interest, there is a need to capture and categorize the data effectively under different operational conditions. This thesis captured traffic and data from sensor motes to analyze and present characteristics of the traffic in a meaningful manner.

Specifically, this thesis studied the traffic generated by wireless sensor networks by setting up two different commonly used network topologies, namely a direct connection to the base and a daisy-chain connection to base. A total of six experiments were conducted, three for each topology. The data traffic between the nodes was captured over an extended duration of time. Using the captured information, analysis was performed to categorize and identify the information through anomalies and variations of traffic patterns. Data were also analyzed to study self-similarity and statistical distribution.

The experimental results have shown that it is possible to differentiate the two different topologies by monitoring the traffic distribution or by analyzing the types of messages sent. The status of the nodes can also be determined with the traffic collected. Examples include new nodes joining the network and operational status of the nodes. Statistical analysis has also been done and found that wireless sensor network traffic is not self-similar except for the interarrival time of the direct connection mode.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

My heartfelt appreciation to my beloved wife, Celeste and son, Hengyue for the understanding, trust, love and moral support demonstrated.

I will also like to express my utmost gratitude to Professor John McEachen for his meticulous advice and encouragements during the process of this thesis. His enthusiasm and belief in my work has given me invaluable support in completing this thesis.

My appreciation to Professor Murali Tummala for his time and effort in reviewing the thesis and splendid feedback contributed.

Last but not least, my thanks to all those who have contributed to the completion of this thesis in one way or another.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The promise of solving challenging real world problems with wireless sensor networks has attracted a lot of attention and interest from the academic world and commercial industry. As a result, wireless sensor networks have been widely researched for applications in different fields.

A wireless sensor network (WSN) is made up of a large number of sensor nodes that are deployed densely over an area of interest. These nodes are usually inexpensive, small, and have an energy constraint limitation. Due to the ad hoc nature of deployment, they can be deployed in places where there is no fixed infrastructure. These sensor devices are designed to cooperatively monitor physical or environmental conditions over a wide area. Data are sampled real-time and collaboratively disseminated through multi-hop broadcasting through the air. The multi-hopping of packets in turn extends the effective range of operation for the networks. WSNs are especially of interest to military planners as they are able to provide accurate, reliable, real-time, and multi-type information and surveillance intelligence from field operations to command centers under different and extreme weather environments and terrain.

As the number of wireless sensor networks deployed increase exponentially, users are faced with many unexpected and peculiar problems which were not seen in the computer simulations or lab experiments. Although traffic analysis in wired and wireless (IEEE 802.11) areas has long been an active research topic, the unique characteristics of wireless sensor networks present a new set of challenges to researchers and developers. In addition, though there is plenty of research done using computer simulations [1] to analyze and predict wireless sensor network traffic behavior, studies have shown that the performance of these simulation results vary widely [3]. A physical, albeit scaled down simulation, is still desired by implementers. Troubleshooting in a wireless sensor network is also several times harder and more expensive than in a traditional network.

In view of this interest, this thesis aims to study the traffic generated by the wireless sensor network by setting up two different commonly used network topologies, namely direct connection to the base and daisy-chain connection to base setup. The data traffic between the nodes is captured over an extended duration of time.

The captured data are analyzed based on the type of packets delivered, behavior and transaction pattern of different types of packets, statistical estimation of self-similarity and statistical distribution based on the packet size and interarrival time of the packet. These results will help in the development of intrusion detection/prevention tools, troubleshooting of performance issues, debugging of applications, and designing components for the nodes.

Three runs of experiments were conducted for each topology. The data traffic among the six sensor nodes and the base node are captured over forty hours for the direct setup and twenty hours for the daisy-chain setup.

The sensor motes used are the Crossbow MICAz motes together with a MTS310 sensor board. The sensor data is transmitted back to the base station using XMesh, the Crossbow routing protocol. A XSniffer mote is placed near the base to capture all incoming and outgoing traffic from the base station mote. This XSniffer comprises of a Micaz connected to a MIB520 gateway to send the data back to a collection terminal. All motes are configured to be in the default group ID of 125 with RF channel of 26 (2480 MHz). Motes are set with the lowest RF power of -25 dBm to reduce the need to place the motes over a large area.

The total numbers of packets collected for the direct connection to base mode is 557,629 over a period of 42 hours and 48 minutes for the first simulation. Another two simulations of 40 hours each were conducted to validate the first data set. For the daisy-chain mode, the total numbers of packets collected was 272,410 over a period of 20 hours and 37 minutes for the first simulation. Another two simulations of 18 hours each were also conducted to substantiate the first data set.

The first result is based on the percentage of transmission for each message type in the whole transmission. It is observed that in both topologies, the percentage of Hlth and Datup type of packets remains the same. However, a lower percentage of AckDwn packets are shown in direct connection configuration when compared with the daisy-chain connection. Another difference is in the Rte packet, where direct mode shows a higher percentage of Rte packet when compared to daisy-chain mode. These statistics can help to identify a direct connection mode from a daisy-chain mode.

In the analysis of different Active Message type packets, it is observed that new mote(s) joining the network can be detected when the broadcast of DatUp packets are captured. Rte packets can also be used as an indication of the mote's operational status as it is sent in regular interval. The two topologies can be identified using either Rte packet length, originating field or Hlth packet length.

In terms of statistical analysis, variance time plots were constructed for both packet length and interarrival time to determine if WSN traffic is self-similar. The results of the variance time plots for packet length for both modes show no indication of self-similarity. In both modes, the DatUp packet has shown to be the dominant packet in the packet length distribution.

The interarrival time for direct connection to base configuration shows slight self-similarity characteristics but is not seen in the daisy-chain mode. Histograms were also plotted to see the distribution of interarrival time in both setups and it shows a distinctive exponential distribution in both direct and daisy-chain connection configurations.

These results are important as it allows identification of the topology of a network by capturing the traffic. The analysis also shows that new nodes joining the network can be identified and connectivity of the nodes can be monitored using Rte packets. The self-similarity study allows a developer to use the exponential model on the network with confidence. The correct modeling will result in better evaluation of network capacity and determination of battery power based on the forecasted traffic workload.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. GENERAL

The promise of solving challenging real world problems with wireless sensor networks has attracted a lot of attention and interest from the academic world and commercial industry. As a result, wireless sensor networks have been widely researched for applications, and this is especially so in the health, military and disaster recovery fields.

A wireless sensor network is made up of a large number of sensor nodes that are deployed densely either inside or near the area of interest. These nodes are usually inexpensive, tiny in size, and have an energy constraint limitation. Positioning of these sensor nodes can either be predetermined or random. Due to the ad hoc nature of deployment, no fixed infrastructure such as a base access point or switch exchange is normally available or required. These sensor devices are designed to cooperatively monitor physical or environmental conditions over a wide area of landscape. Data are sampled real-time and collaboratively disseminated through multi-hop broadcasting. Nodes with neighbors who are within the radio range can communicate directly by going through the wireless link. Communication with nodes that are far apart can be done through relaying of packets over other nodes, which in turn extends the effective range of operation for the wireless sensor networks.

These nodes have the characteristic of being autonomous, self-configuring, self-healing, easily deployed, robust to nodes failures, and adaptive to different kinds of environments. These characteristics make them an excellent candidate for many real life problems. They are especially of interest to the military environment as they are able to provide accurate, reliable, real-time, and multi-type information and surveillance intelligence from field operations to command centers under different and extreme weather environments and terrain.

B. THESIS MOTIVATION

As the number of wireless sensor networks deployed increases exponentially, users are faced with many unexpected and peculiar problems which were not seen in

computer simulations or lab experiments. Although traffic analysis in wired and wireless (IEEE 802.11) areas has long been an active research topic, the unique characteristics of wireless sensor networks present a new set of challenges and perspective to researchers and developers. Some examples of these challenges include ad hoc network architecture, shared wireless channel, scarce energy resources, and frequent changes of network topology.

There has also been considerable research done using computer simulations [1] to analyze and predict wireless sensor network traffic behavior. Some studies have also developed simulators with the sole purpose of running only wireless sensor network applications [2]. However, there are studies showing that the performance of these simulators' results varies widely [3].

Troubleshooting a wireless sensor network is several times harder and more expensive than a traditional network. This is due to the fact that wireless sensor network deployment is likely to be in a hard to reach environment or at a location where it is hazardous to humans. As minimizing energy usage during operation is one of the most critical design criteria, there is a possibility that it is difficult or impossible to extract additional information for analysis or troubleshooting.

The data are later categorized and analyzed to provide a better understanding of the traffic profile of a wireless sensor network. These results will help in the development of the following research.

1. Intrusion Detection/Prevention

By doing a traffic profiling of a wireless sensor network under different topology settings, data can be captured, compared, and modeled as a signature or baseline for normal operational behavior. This baseline can be used in an intrusion detection or prevention mechanism to identify a misbehaved network.

2. Troubleshooting of Performance Issues

By comparing traffic against the captured data in the experiment, it will allow the developer and user to quickly pinpoint the source(s) of any performance problems or faulty nodes. With this knowledge, users will be able to spend more time focusing on the application instead of troubleshooting and debugging. Time and money can be saved as

immediate action can be taken to resolve issues quickly. This study can also be used as an input to proactively monitor and identify issues quickly and accurately during the operation phase, saving situations where long downtime is prohibited or undesirable.

3. Debugging of Applications

With the experiment data sets, an application developer will be able to compare the results against his or her mock up to understand and troubleshoot any problem before the application is deployed.

4. Design of Components

With the collected statistical traffic of the wireless sensor network, analysis of the data will help to improve the design of memory size and processor speed of the nodes. By using the data to customize the component for different specific applications, longer operational life, cheaper nodes, and larger traffic throughput is achievable.

C. THESIS OBJECTIVE

As the design of robust and reliable wireless sensor networks becomes an increasingly challenging task, there is the need to study and understand the traffic profile of wireless sensor networks in more detail. In view of this interest, this thesis aims to study the traffic generated by a wireless sensor network by setting up two different commonly used network topologies. A total of six experiments were conducted, three for each topology. The data traffic between the nodes is captured over an extended duration of time. Using the captured information, analysis will be done to categorize and identify the information through anomalies and variation of traffic patterns. Data will also be analyzed with statistical tools for self-similarity and statistical distribution.

The captured data will be analyzed based on

- Type of packets delivered
- Behavior and transaction patterns of different kinds of packets
- Statistical estimation of self-similarity based on the packet size and interarrival time between packets
- Statistical behavior of packet size and interarrival time between packets

These results will help to create a traffic profile for the studied wireless sensor network environment for an in-depth understanding of the research issues in this emerging technology. The analysis of the study will assist developers and users in understanding

the traffic behavior and statistical characteristics of the wireless sensor network and help in developing the next generation of wireless sensor networks. This data can also be used to identify and preempt potential problems or misbehavior in a wireless sensor network.

D. PREVIOUS WORK

Teo [4] had studied the characteristics and performance of XMesh, the routing protocol created by Crossbow Technology. The studies included the connectivity range of the motes in different transmission power settings. He has also simulated node failure in a wireless sensor network and recorded the time needed to recover from the breakage in a network. The results from his study are based on the time of recovery and signal strength of the motes.

Kirykos [5] and Ling [6] have studied the self-similarity characteristic of a wireless sensor network, where the analyzed data were collected over a short period of time. Kirykos results were based on a two node setup with a captured time of no more than three hours. Though Ling's traffic was based on a network of more than four motes, the captured time only lasted for 12.5 minutes. This thesis extends the data collection to a larger number of nodes and longer duration of time to better approximate a realistic deployment.

E. THESIS ORGANIZATION

Chapter I presents the motivation and objective of the thesis. It also presents prior work done. Chapter II provides an overview on the theory of wireless sensor networks. Both hardware and software components of wireless sensor networks are presented. The components used in the experiment are also described with detailed specifications. Different applications used by the wireless sensor network are illustrated to provide a better understanding of the capability of wireless sensor networks. Detailed discussions on the different topologies are introduced followed by a brief presentation of the underlying ZigBee standard.

Chapter III specifically focuses on the TinyOS and XMesh packet structures. The different kind of Active Message (AM) types are explained next. It also briefly reviews the concept of self–similarity and the methods used to determine self-similarity of traffic.

Chapter IV describes the setup of the experiment with detailed specifications and configuration. Chapter V shows the collected data and presents the analysis of the experimental data based on the objective of the thesis. It also discusses the findings and results of the experiment. Chapter VI provides the conclusion to the thesis. Recommendations are made and future work is proposed.

THIS PAGE INTENTIONALLY LEFT BLANK

II. WIRELESS SENSOR NETWORKS

This chapter begins by introducing the characteristics of a wireless sensor network and states the performance metrics for evaluating a wireless sensor network. It also points out the differences between a wireless sensor network and a traditional network. Various applications are cited to give an overview of the capability of this emerging technology. It discusses the various components of a wireless sensor network, and the hardware used in the experiment. Different topology setups are discussed in detail and the final section describes the underlying ZigBee standard commonly used in wireless sensor networks.

A. INTRODUCTION TO WIRELESS SENSOR NETWORKS

Wireless sensor networks have created a new paradigm for inexpensive, accurate and reliable monitoring and sensing. They are starting to replace traditional sensor systems, which are often large, expensive to operate, difficult to deploy, and require complex maintenance. Though wireless sensor network applications show a great deal of diversity, there are still a number of shared characteristics among them, regardless of their specific sensors and application objectives. Some of these characteristics include having self-organizing capabilities, short-range broadcasting, multi-hop routing, cooperation between sensor nodes, frequent changes of topology due to node failures, and limitations in power, memory and computing power. A list of metrics is recommended by [7] to determine the performance of a sensor network.

1. Performance Metrics for Wireless Sensor Networks

a. Energy Efficiency

As sensors are designed to be deployed unattended, they are normally powered with batteries. With efficient energy usage, the energy resource can be maximized and the lifetime of the network will be extended.

b. Latency

Though most sensor networks are not deployed for real-time applications, there is still a need for the results to be sent back within a certain time frame. One of the examples is a wireless sensor network deployed as a defense or intrusion detection perimeter.

c. Accuracy

Having high data accuracy will enable a user to deploy fewer nodes or cover a wider area of monitoring.

d. Fault tolerance

As sensor nodes are prone to failure due to the limited energy source and radio range, the network needs to be highly resilient to node failure. This can be achieved by redundancy of nodes or collaborative processing and communication, or a combination of both.

e. Scalability

As a wireless sensor network is normally made up of hundreds to thousands of nodes, it has to be able to ensure that the performance of the network is not affected with increasing network size.

2. Differences Between Traditional Networks and Wireless Sensor Networks

Although sensor networks have similar components as traditional networks, they have to be designed and implemented differently. This is because wireless sensor networks have various constraints in their computation power, storage, memory, energy, and bandwidth. The major issue in wireless sensor networks is often the energy resources as they are normally deployed unattended or in a hazardous environment or a physically non-accessible location. Parameters like latency, bandwidth, and accuracy are often trade offs with this major design consideration to extend the operational lifetime of the networks.

Wireless sensor networks can be further differentiated [8] from traditional ad hoc networks by the following:

- Numbers of nodes is on the order of hundreds to thousands to enable finer granularity and increased robustness of the network
- Sensor nodes are more densely deployed than in an ad hoc network
- A sensor node is prone to failures
- Frequent changes of topology are expected in a wireless sensor network

- Transmission by broadcasting instead of point-to-point communication
- Constraints in power, processing power, bandwidth, and memory
- Sensor node may not be uniquely identifiable due to a large number of sensors

3. Challenges

As mentioned in the previous section, wireless sensor networks are different in many ways compared to traditional networks. Therefore there are different sets of challenges for WSN. A number of studies have been done in the area of sensor networks and challenges are stated to focus on the problems to be resolved [9] [10] [11]. A more systematic way of looking at the challenges of wireless sensor networks has been described by Rommer and Matten [11]. They view the challenges from the perspective of the "design space." The twelve design space components are as follows:

- Deployment (random or one time use)
- Mobility (active or passive or partial or all)
- Cost, size, resources and energy
- Heterogeneity (homogeneous or heterogeneous)
- Communication modality (radio or light or sound)
- Infrastructure (infrastructure or ad hoc)
- Network topology (star or hybrid)
- Coverage (sparse or dense or redundant)
- Connectivity (connected or intermittent)
- Network size (tens, hundreds or thousands)
- Lifetime (days or years)
- Quality of service (QoS) requirements (real-time or robust to failures)

As can be seen from the above, a wireless sensor network has a very different set of challenges from the traditional wired or wireless network; therefore relevance of previous knowledge will need to be assessed again.

4. Application

One of the reasons why wireless sensor networks are becoming more popular is due to the wide range of applications where they can be deployed. Applications vary from detection of an event, periodic measurement, and edge detection, to tracking of targets [12]. The following are some examples of applications.

a. Environmental Monitoring

Sensors can be deployed to monitor an environment where it is labor intensive or hazardous to mankind. Some examples include using sensors to do precision measurements of fertilizer concentration and seismic activity, monitoring of water quality, tracking of wild animals, detection of fire or floods, and checking contamination levels of air or soil. One such example is using a wireless sensor network to observe the breeding behavior of a small bird called Leach's Storm Petrel on Great Duck Island, Maine, USA [13]. Previously it was almost impossible to collect data like their usage patterns of the nesting burrows, yet now variations among breeding sites to changes in environmental conditions inside and outside of the burrow can be done with wireless sensor networks. Nodes with humidity, pressure, temperature and light sensors are installed inside the burrows and surface to collect these statistics. Data are then relayed back to the back-end station through the multi-hop network.

b. Seismic/Glacier Detection

Sensors can be placed to detect seismic activity along a fault line or coastline. This will give early detection to a natural disaster like a tsunami, a volcanic eruption or an earthquake. They can also be deployed to glaciers to monitor the displacements and dynamics inside the glacier [14]. Such monitoring normally requires a few months to years of monitoring in order to have any meaningful results.

c. Disaster Operations

A wireless sensor network can be a helpful tool in times of disaster operations. These disaster scenarios can range from chemical or biological contamination, a forest fire, and flooding to an earthquake. A network of communication can be deployed without existing infrastructure to help the authorities have better command and control over the operations. It can also be setup to monitor the situation without risking any life. Sensors have been deployed to assist rescue teams in saving people buried in avalanches [15]. By using this solution, the rescue team is able to locate the victim and have access to the health state of the victim for prioritization.

d. Medical Monitoring

A wireless sensor network can be setup to monitor vital signs of patients either in the hospital or out of the hospital. It can also be used as a tool for doctors to retrieve past data collected since the last visit. Another use of a wireless sensor network is to track suspected infectious patients so that further spread of a contagious virus is minimized or prevented. Such implementation has been used and proved to improve data accuracy and provide more convenience for the patient [16].

e. Military Surveillance

Wireless sensor networks are increasingly being used by the military in recent years and form a critical part of military command, control and communication. There are even routing protocols specifically designed for military applications [17]. The small form factor of a node makes it attractive to use it as an area monitoring tool. In addition to that, WSNs can be easily deployed, and they are coupled with self-organization capability and high fault tolerance, which is an essential requirement in a military environment. By fixing sensors on to equipment and personnel, weapons and troop status information can be gathered and sent back to a command center to provide information and better battlefield intelligence. An ad hoc network can be setup easily where wiring is not feasible or possible. Sensors can be deployed as a smart minefield with intelligence that can differentiate between a friendly or enemy force. In biological or chemical warfare, sensors can be deployed to determine the presence of a toxic substance in the monitored area without unnecessarily risking human life. It can also be used to track movement of troops or vehicles over an area [18]. Another application is a situation where magnetometer sensors are dropped randomly over an area by unmanned aerial vehicles (UAV) and nodes can collect the data from the field. With the sensor data, it can estimate the path and velocity of the tracked vehicle and send it back to the UAV.

B. HARDWARE

In this section, the various components of a node in a WSN will be discussed. This will be followed with the details on the actual hardware used in the experiment.

1. Components of a Sensor Node

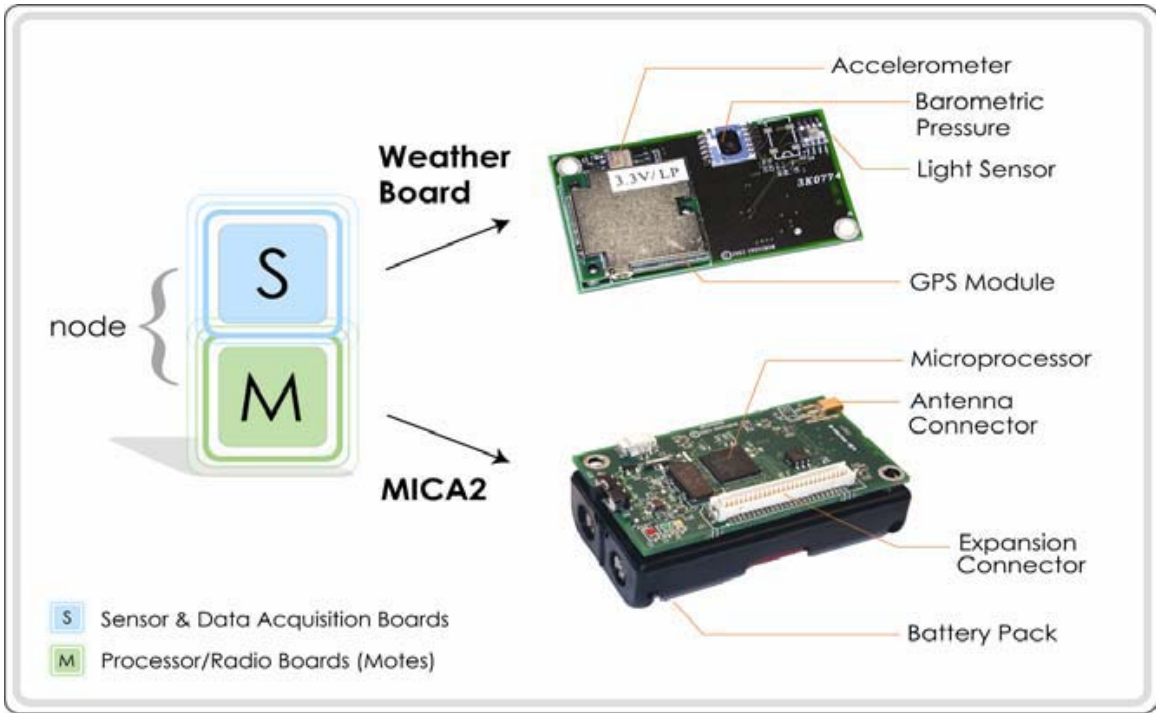

Figure 1. Basic Component of a Sensor Mote. From Ref [19].

Figure 1 shows the pictorial diagram of a typical sensor node. They are basically made up of a sensor board and a mote. A mote is made up of a processor, memory, radio transceiver, and power supply. These components will be briefly discussed in the following paragraphs. A block diagram of the mote is shown in Figure 2.

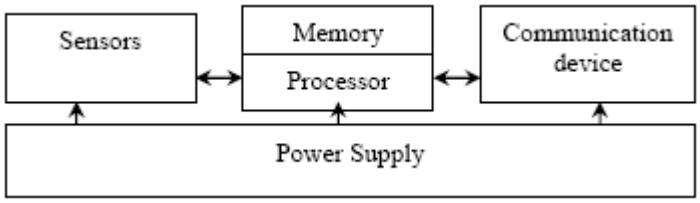

Figure 2. Block Diagram of Sensor Mote Components. From Ref [12].

a. Processor

The main function of a processor is to collect data from sensors and process the data. It will then decide when and where to send it, to receive data from other sensors nodes, and decide on the actuator's behavior. Various programs have to be executed locally, ranging from communications protocols and time critical signal processing to application programs.

b. Memory

There are two types of memory commonly found in a mote, namely Random Access Memory (RAM) and flash memory. RAM is used to store intermediate data like sensor readings and packets from other motes. Though RAM is fast, the disadvantage of using it is that it loses data once the power is interrupted. Therefore the program code is stored in flash memory where data can be retained after a power interrupt. The flash memory also serves as a secondary storage for data in case the RAM is not sufficient. The high energy usage and slow access time however prevent it from replacing RAM completely. A trade off between the two is needed in order to suit the requirements of the application.

c. Power Supply

Wireless sensor networks are normally deployed in an unattended or hazardous environment where replacement of energy is impractical or impossible. Therefore energy for the motes is normally the most important component in order for the network to perform to its intended design. One way to maximize the energy is to provide maximum stored energy with minimal energy usage. This results in a trade off of sacrificing accuracy and responsiveness. The other way to extend the lifetime of the motes is by scavenging energy from some external power source. Some of the external sources include using temperature gradient, vibrations, pressure variations, air/liquid movement, and solar power.

d. Radio

Motes need to communicate with each other to exchange data and extend their operational range. Radio frequency (RF) based communication is chosen for wireless sensor networks due to its relatively long range, high data rates, and acceptable error rate, as well as it does not require line of sight to communicate. The commonly used frequency range is between 433 MHz to 2.4 GHz for wireless sensor networks. The radios are built to be bidirectional but in half duplex mode. Multiple channels are available for each band, where software can be used to control the band.

e. Sensors

There are two main categories of sensors, namely passive and active sensors. Passive sensors are defined as sensors that measure a physical quantity by probing without manipulating the environment. Examples of these kinds of sensors are temperature, light, vibration, and humidity sensors. Passive sensors can be further broken down to be omni-directional or narrow-beam sensors, as defined by their direction of measurement. An example of the latter is a camera, which takes measurement in a predefined direction. A seismic or radar sensor is an example of an active sensor, which probe the environment actively.

2. Types of Hardware Used in the Experiment

The different types of components used in the experiment are discussed in the following sections.

a. Mote-Micaz

Figure 3. Actual MICAz Hardware. From Ref [19].

Micaz, as shown in Figure 3, is one of latest generation of motes developed by Crossbow Technology. It is a popular choice for research as it is one of the few models of motes that is compliant with the IEEE 802.15.4 standard. In addition, it also provides hardware security using the AES-128 standard. It has the same microprocessor as MICA2, an ATmega128L chip, capable of a maximum throughput of 8 million instructions per second (MIPS) when operating at 8 MHz. It uses a Chipcon CC2420 for its radio communication. The Chipcon CC2420 implements the physical layer as prescribed by the IEEE 802.15.4 standard, transmitting in the standard specified 2.4 - 2.4835 GHz range, a globally compatible industrial, scientific and medical (ISM) band. The radio transceiver can transmit up to a 250 kbps data rate. It has 128 kB of flash for

program memory and 4 kB of SRAM for data and variables. It uses direct sequence spread spectrum (DSSS) encoding with Offset Quadrature Phase-Shift Keying (OQPSK) modulation. It theoretically can transmit up to 135 meters, line of sight with a half-wave dipole antenna [19]. There is also an external serial flash memory of 512 kB.

b. Gateway-MIB 520

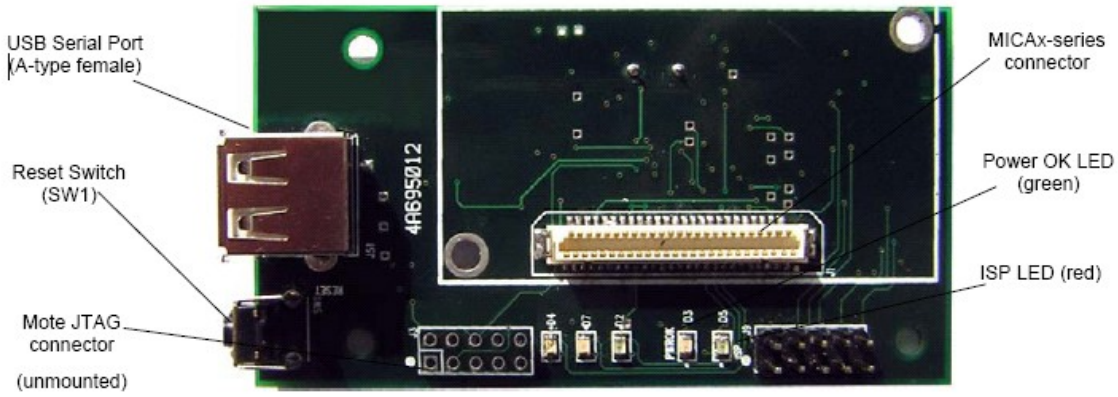

Figure 4. MIB520 USB Gateway. From Ref [19].

A gateway is needed for configuration and programming applications into MICA sensor motes. This gateway also serves as an interface for the base station to transmit the data back to the data collection terminal.

Figure 4 shows the top view of a MIB520 programming board. The MIB520 provides USB connectivity from the PC to the MICA family of motes. It offers two separate ports, one dedicated to in-system Mote programming and a second for data communication over USB. Any motes can operate as a base station once connected to the gateway. The MIB520 has an on-board processor that programs MICA Processor Radio Boards (PRB). USB bus power eliminates the need for an external power source for the gateway and motes, making it more attractive for field deployment.

c. Sensor-MTS310

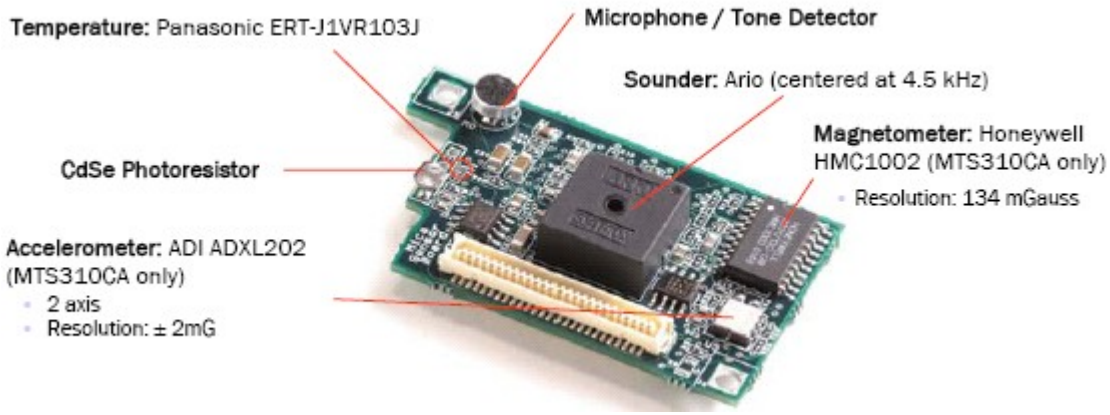

Figure 5. Crossbow MTS310 Sensor Board. From Ref [19].

The Multipurpose Sensor Board (MTS310), as shown in Figure 5, is one of the sensor boards which have a variety of sensing modalities. These sensing capabilities can facilitate developing a variety of applications like vibration and magnetic anomaly detection, acoustic ranging, vehicle detection, target movement, acoustic ranging and many more. Five sensors are embedded into the board as shown in Figure 5. They are described in the following paragraph. [20]

(1) Microphone and Tone Detector. This component can be used for acoustic recording and measurement, where audio files can be recorded into the logger flash memory for later download or analysis. Another more interesting application of this component is to use it for acoustic ranging. A pulse can be sent using the sounder together with an RF packet via radio. Another mote will listen for the RF packet and note the time of arrival by resetting a counter on its processor. It then increments the counter until it detects the pulse. The time of flight of the sound wave between the two motes can be determined from the counter value. It can then be converted into an approximate distance between the two motes. A crude but cheap localization and positioning tool can be built using this method.

(2) Light Sensor. The light sensor is made up of a simple Cadmium Selenium (CdSe) photocell. It has a maximum sensitivity at the light wavelength of 690 nm. The resistance varies from 2 kΩ, when exposed to light, to 520 kΩ when in total darkness. One of the common applications is to use it for motion detection.

(3) Temperature Sensor. The thermistor is a surface mount component with a mid-scale reading at 25 °C.

(4) Two Axis Accelerometer. This accelerometer features a very low current draw (<1mA) and 10-bit resolution. It can be used for tilt detection, movement, vibration or seismic measurement. It has a G-range of $\pm$2 G, resolution of 2 mG, and sensitivity of 167 mV/G $\pm$ 17%.

(5) Two Axis Magnetometer. This is a silicon sensor that has a unique bridge resistor coated with a highly sensitive nickel-iron (NiFe) coating. This highly sensitivity bridge enables it to measure the earth's field and other small magnetic fields. Vehicle movement from a radius of 4.5 m (15 feet) has been successfully detected using this sensor in a test.

C. SOFTWARE

Traditional operating system will not be suitable for wireless sensor node due to the node's constraint in energy, processing power and memory capacity. This section will discuss TinyOS, the commonly used operating system in wireless sensor nodes.

1. TinyOS

TinyOS is an open-source operating system designed for wireless embedded sensor networks. It was originally developed by the University of California, Berkeley, featuring component-based architecture and enabling rapid implementation. It is widely used as the operating system for wireless sensor network applications due to the small memory footprint requirement. It is also an object-oriented and event-driven operating system that makes it attractive for low power devices. TinyOS has a huge component library which includes network protocols, sensor drivers, distributed services, and data acquisition tools. These components allow developers to use or further refine a custom application in a short period of time. Fine-grained power management can be achieved with the event driven execution model in the TinyOS architecture.

TinyOS can support various microprocessors ranging from 8-bit architectures with as little as 2KB of RAM to 32-bit with 32 MB of RAM or more, making it very flexible for developers. It provides a well-defined set of application programming interfaces (APIs) that allows an application designer to select from a variety of system components in order to meet application-specific goals. This API provides access to computing capabilities of the sensor node and allows more intelligence to the network. With these capabilities, nodes can preprocess sensor data and remove unnecessary messages before transmission. This in turn helps to optimize both network throughput and battery life, thus extending the operational lifetime of the network.

D. TOPOLOGY

Topology is normally described as the physical or logical layout of the nodes in a network. There are a number of network topology configurations that present their own set of advantages and disadvantages. Studies have to be done to choose an efficient network topology to meet the power requirements of a WSN deployment. The three most common topologies deployed for wireless sensor networks are the star, mesh and star-mesh hybrid, where each supports different power profiles. Each of them is appropriate under some circumstances but may be unsuitable in others. In wireless sensor networks, there are three different types of nodes in a topology.

- Endpoints - Endpoints are integrated with sensors and actuators to capture sensor data. They do not have the capability of forwarding packets upstream or downstream. Endpoints are commonly referred to as reduced function devices (RFDs) in a ZigBee network. Each RFD can only associate with a single full function device (FFD) at an instance.
- Routers - Routers can be data collectors like the end points, but they have the capability to route messages, thus extending the network area coverage. They can also dynamically reroute packets through a second link in the case of congestion or device failure. Routers are commonly referred to as full function devices (FFDs) in a ZigBee network.
- Gateway - A gateway aggregates data from all nodes and acts as an interface to the data collection terminal, LAN, or Internet. It is also often used for the monitoring and configuration of a network. ZigBee terms a gateway as a personal area network (PAN) coordinator.

In the paragraphs that follow, we will discuss the various topologies used in WSN deployment.

1. Star Topology

A star topology is the simplest topology of the three as all the nodes are a single hop away from the gateway. Figure 6 shows the diagram for a star topology. As no routing is needed, all nodes can be configured as endpoints. The endpoints do not pass data or commands to each other, as the gateway is used as the coordination point instead. This topology consumes the least power for the endpoints as they are allowed to go into sleep mode, wake up and take a measurement, and send the measurement back to the gateway without the need to synchronize or route other node packets.

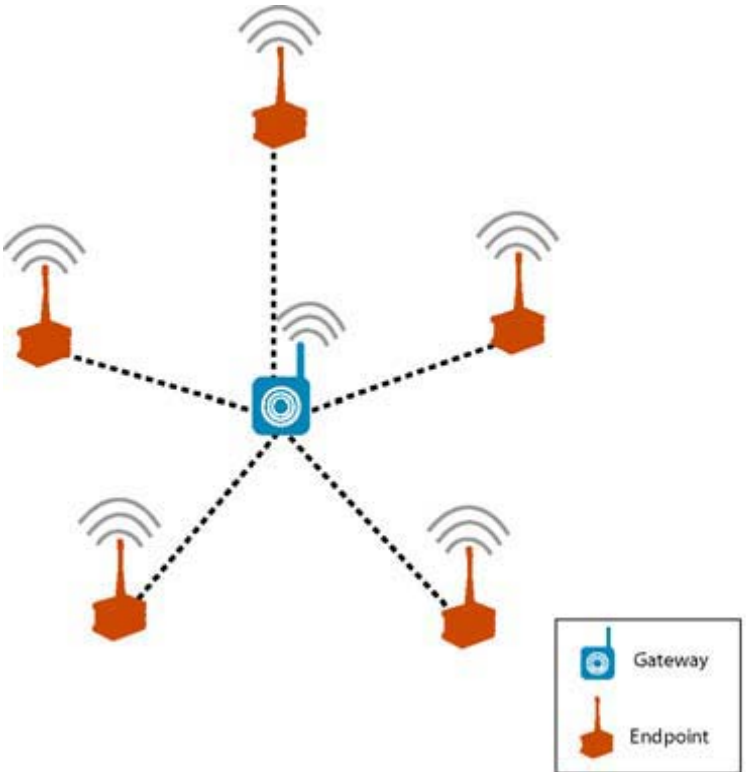

Figure 6. Star Topology. From Ref [21].

However, this topology severely limits the area of sensor monitoring as it is constrained by the transmission distance of the radio between the endpoints and gateway. This transmission distance varies as it is defined by the quality of the radio link. This link can be affected by physical objects crossing the path of the endpoint and gateway, interference from radio wave reflections off nearby surfaces, or interference from other

wireless transmissions like microwave ovens or 802.11 networks. With no routing capability, endpoints will not be able to communicate with the gateway once the communication path is blocked.

Star topologies are a good fit in environments where the area of monitoring is small and extended duration of monitoring is needed. Endpoints with two AA batteries are known to be able to last up to five years in this topology [21].

2. Mesh Topology

Figure 7 shows the diagram for a mesh topology. Mesh or peer-to-peer (ZigBee equivalent term) topologies are characterized by nodes with connectivity to each other through multi-hopping transmissions. They are configured as a router and theoretically have an unlimited monitoring range. A mesh topology dynamically optimizes routes based on the best RF link quality between nodes. Functioning as a router, each node can route packets through another link if the primary link fails, making the network highly tolerant to failures.

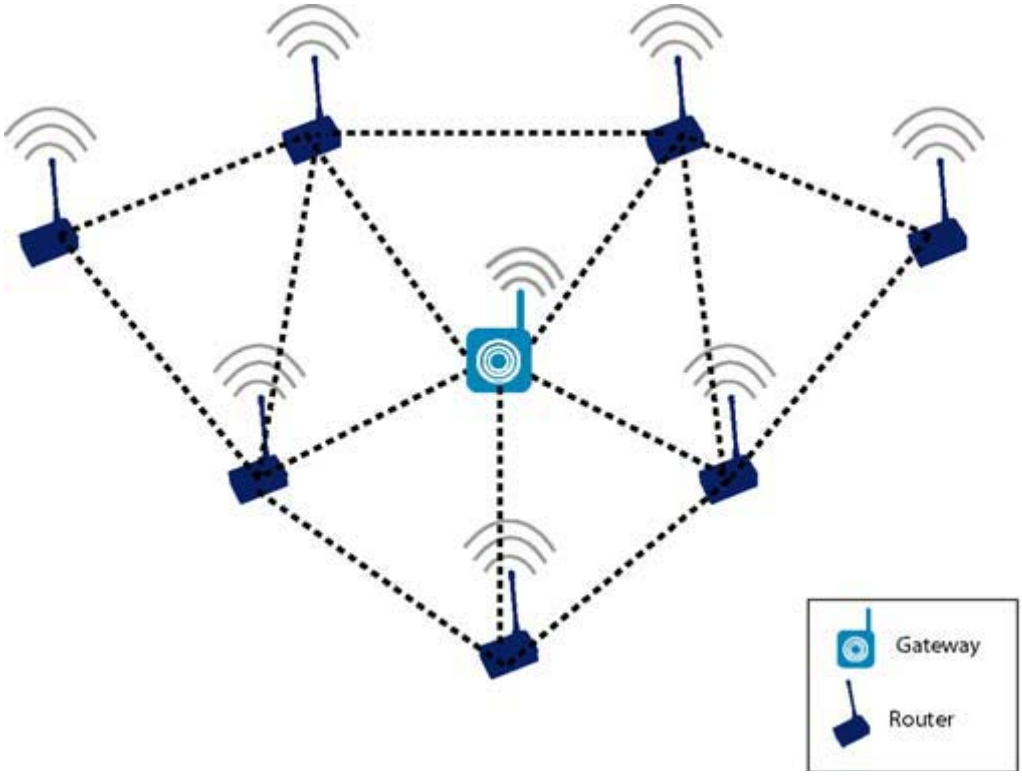

Figure 7. Mesh Topology. From Ref [21].

Due to multi-hopping through nodes to deliver the packets, the network may also have increased latency. The added complexity means that the nodes will need more memory and computing power, and have less operating lifetime. Mesh routing protocols can help to optimize the battery life by coordinating the sleep mode across the network or have frequent short wake up times to listen for messages to be routed.

Mesh topologies are suitable for networks that need an extended operating range or where frequent changes of RF conditions are expected which require dynamic re-routing. Battery operational life for the nodes in this topology is expected to last between one to three years [21].

3. Star-Mesh Topology

A star-mesh or cluster tree (ZigBee equivalent term) network is a combination of both star and mesh topologies. It combines both the simplicity and low power features of a star topology together with the extended operational range and dynamic routing of a mesh network. Figure 8 shows the diagram for a star-mesh topology.

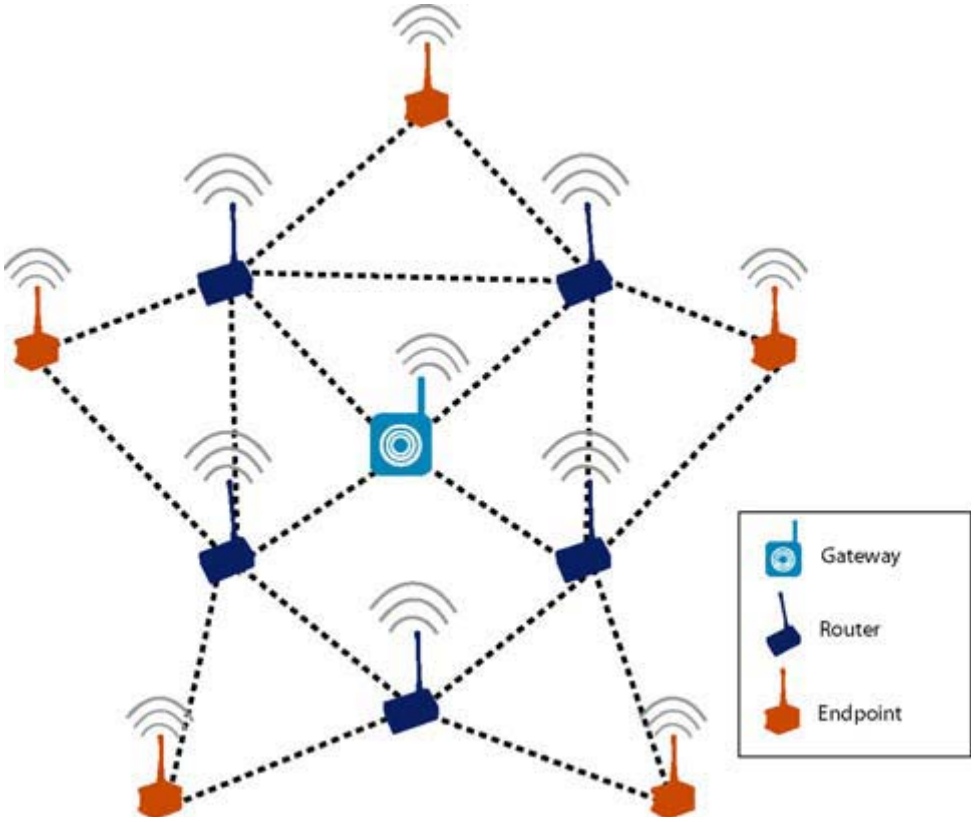

Figure 8. Star-Mesh Topology. From Ref [21].

This hybrid topology organizes sensor nodes in a star formation around routers, which in turn organize themselves into a mesh topology. The router with the direct link back to the gateway allows the range of the network to be extended and provides robustness through redundant links. If one of the routers fails or there is interference to the radio link, nodes will be able to reroute the traffic around the other routers. Routers in the star-mesh network are normally line-powered devices as they are constantly awake to listen and forward incoming and outgoing messages.

This topology will be suitable for networks that need a backbone with high bandwidth and minimal latency or where lined power is readily available for the routing nodes. A star-mesh topology is used for home lighting control as stated in ZigBee specifications [21].

E. ZIGBEE AND 802.15.4

Like most technologies, communication protocols in WSNs are looking towards standardization so that different vendors' products can interoperate with each other. It is much more cost-effective to design applications based on a common standardized ZigBee platform rather than to create a new proprietary solution from scratch each time. Though Bluetooth and wireless local area networks (WLAN) have been around for many years, they are found to be unsuitable for low power applications due to their high cost as well as complex and power hungry radio frequency integrated circuits (IC) and protocols. Figure 9 shows the intended data rate and range of operation for some of the popular wireless standards.

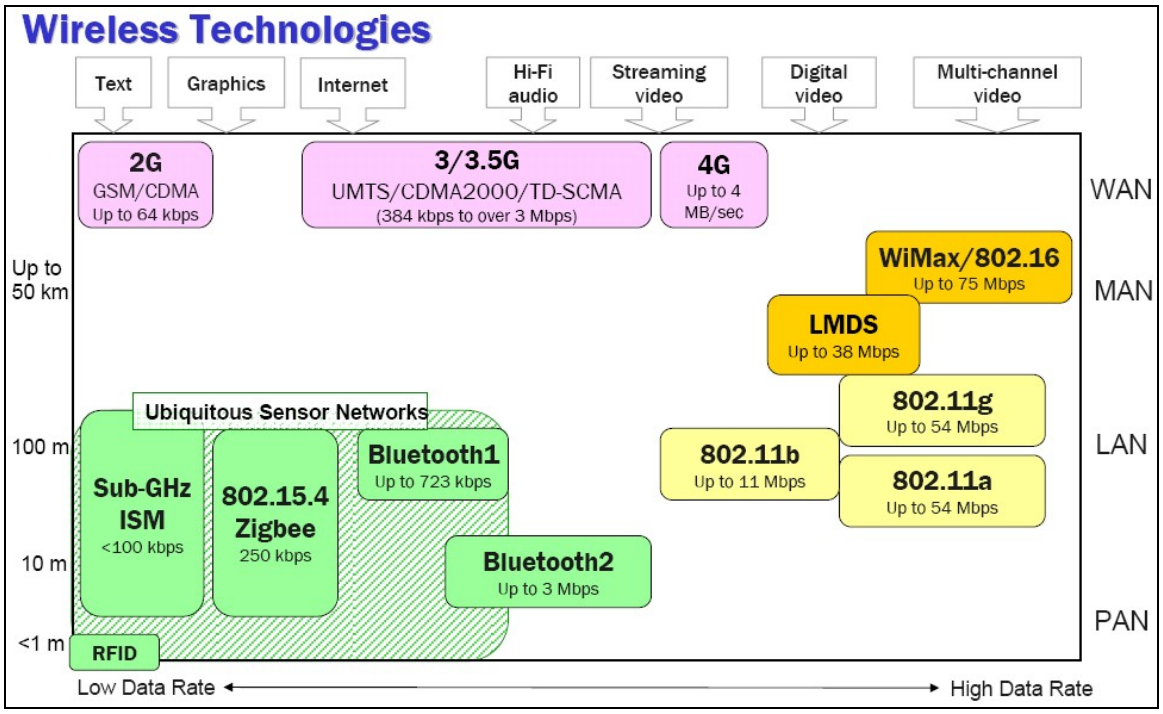

Figure 9. Wireless Technologies. From Ref [19].

IEEE 802.15.4 is a standard that has been built with low cost, low data rate and low power consumption in mind. It defines the physical layer (PHY) and medium access control (MAC) layer for low rate – wireless personal area networks (LR-WPANs). It can operate in one of the three ISM frequency bands as shown in Figure 10. The frequency band 2.4 GHz has the most potential for wide usage as the high data rate reduces the transmission time and thus creates energy-per-bit savings, an important criterion for wireless sensor nodes. Given the higher number of available channels and no restriction in most countries, it is likely to be the standard for many wireless sensor networks in the near future.

Figure 10. 802.15.4/ZigBee frequency band. From Ref [22].

Some of the features of the IEEE 802.15.4 standard are [19] [23]:

- Star or peer-to-peer operation
- Allocated 16-bit short or 64-bit extended addresses
- Allocation of guaranteed time slots (GTSs)
- 128-Bit AES encryption in hardware
- Carrier sense multiple access with collision avoidance (CSMA-CA)
- Full acknowledgement protocol for transfer reliability
- Low power consumption, (e.g. the CC2420 receives at 18 mA and transmits at 20 mA @ 1 mW)
- Energy detection (ED)
- Link quality indication (LQI)

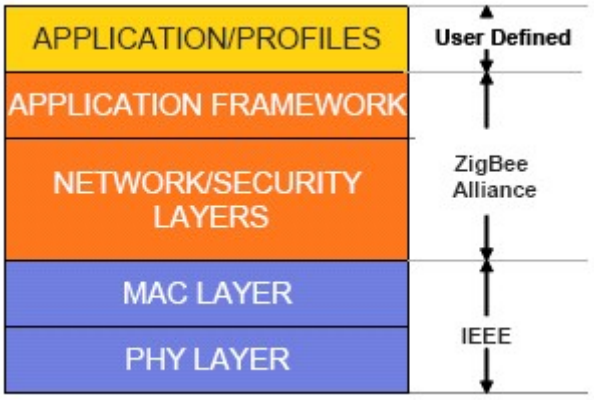

Figure 11. IEEE 802.15.4 and ZigBee Stack. From Ref [24].

ZigBee is a standard that is based on IEEE 802.15.4 and expands it by adding a network, security, and application services framework to the physical and MAC layers defined by IEEE 802.15.4. A pictorial block diagram to describe the relationship can be seen in Figure 11. ZigBee became the commercial name for the IEEE 802.15.4 standard after an alliance formed between the IEEE 802.15.4 task group and the ZigBee Alliance. ZigBee's self-forming and self-healing mesh network architecture allows data and control messages to be passed from one node to another node via different multiple paths. This feature improves data reliability and extends the range of the network.

F. SUMMARY

This chapter highlights the characteristics and differences between traditional networks and wireless sensor networks. Some of the applications of WSN were discussed to better appreciate the potential of the technology. Generic components of a WSN were outlined and specific components used in the thesis were discussed. Topology configuration and the ZigBee standard were introduced. The next chapter will concentrate on different fundamental parameters of WSN traffic.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PARAMETERS OF TRAFFIC ANALYSIS

In this chapter, the packet format of the TinyOS message will be examined. This is followed by a close up look into the MICAz, XMesh and XSensor headers. The packet transaction will be explained in detail to allow a better understanding of the results in Chapter V. Lastly, the basic concept of self-similarity will be discussed.

A. TINYOS PACKET FORMAT

To analyze the traffic profile, an understanding of the transmission setup and packet header information is important.

The TinyOS message packet transmission sequence is shown in Figure 12. Before packets are sent through the radio, there is a need to synchronize the receiver to receive the packet. There is a random MAC delay of up to 15 packet times prior to transmission. This is followed with a check for a clear channel using Carrier Sense Multiple Access Collision Avoidance (CSMA/CA). If the channel is free for transmission, a delay of 250 milliseconds is observed. The data are then added with a preamble message which helps the receiver to synchronize the timer to the incoming data. A two-byte synchronous (Sync) message is then added to mark the start of a TinyOS message. Transmission time of a packet is dependant on the link rate of the motes. With a link rate of 250kbps in IEEE 802.15.4 format, the time duration to transmit a byte will take about 32 microseconds.

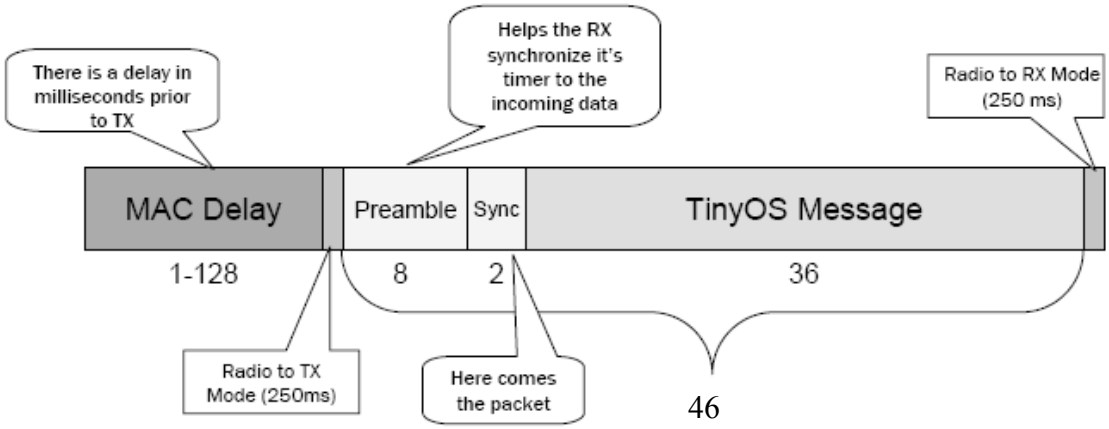

Figure 12. Transmission Sequence of TinyOS Packet. From Ref [25].

1. TinyOS Header

The TinyOS message consists of a header address with a size of five bytes, followed by a payload field that can go up to 29 bytes and 2 bytes of cyclic redundancy check (CRC) as shown in Figure 13.

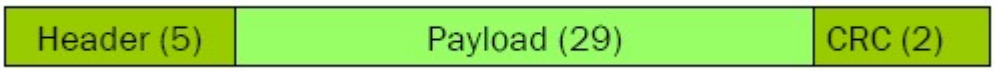

Figure 13. TinyOS header. From Ref [19].

The TinyOS header is further broken down as shown in Figure 14. The components are as follows:

- Destination addresses (2 bytes) – this field specifies the next single hop destination.
- Active message type (1 byte) – this field identifies the different kinds of message types sent.
- Group identification (1 byte) – nodes will be able to differentiate the traffic from other groups and traffic will be dropped if the group ID does not match the receiving mote group ID.
- Length (1 byte) – this field indicates the size of the payload.

Bytes: 2	1	1	1	Variable
Destination Address	AM Type	AM Group	Length	Data Payload
TinyOS Header				

Figure 14. TinyOS Header. From Ref [19].

2. MICAz Header

As MICAz uses the CC2420 IEEE 802.15.4 radio hardware, it requires 802.15.4 headers to be added. Therefore the MICAz packet has a total length of ten bytes for its header. It can also be extended up to 125 bytes as supported by 802.15.4 if the need arises. This makes the total message length of MICAz packets of 41 bytes as compared to a standard TinyOS message of 36 bytes.

Bytes							
1	2	1	2	2	1	1	variable
Length	FCF	SeqNo	Dest Pan ID	Destination Address	AM Type	AM Group	Data Payload

Figure 15. Micaz TinyOS Header. From Ref [19].

The MICAz TinyOS header is shown in Figure 15. The header is comprised of:

- Length (1 byte) - this field indicates the size of the payload.
- Frame control (2 bytes) – for frame control.
- Sequence number (1 byte) – this counter keeps track of the packets sent. This can be used for troubleshooting (to see if there is any lost packet), QoS (determine the service level when a predefined number of packets are not received within the stated time frame) or security (prevent replay attack) purposes.
- Destination PAN ID (2 bytes) – Identifies the 16-bit address of the mote (used in ZigBee specification).
- Destination addresses (2 bytes) – Identifies the 16-bit address of the destination mote.
- Active message type (1 byte) – identifies the application ID of the packet.
- Group identification (1 byte) – The group ID of the mote.

3. XMesh Header

Bytes: 5	2	2	2	1	Variable
TinyOS Header	Source Address	Origin Address	Sequence Number	Hop Count	Data Payload
	XMesh Header				

Figure 16. XMesh Header. From Ref [25].

Following right after the TinyOS header is the XMesh header. It has a payload length of 5 bytes as shown in Figure 16. This routing protocol header includes:

- Source address (2 bytes) – identifies the mote that transports this packet.
- Origin address (2 bytes) – identifies the mote that created the packet.

- Sequence number (2 bytes) – Indicates the numbers of packets sent.
- Hop count (1 byte) – The number of hops needed to reach the base. This is used in calculating the cost path in the routing algorithm.

4. XSensor Header

Bytes: 5	0/7	1	1	2	Variable
TinyOS Header	XMesh Header	Sensor Board ID	Sensor Packet ID	Parent	Data Payload
		XSensor Header			

Figure 17. XSensor Header. From Ref [26].

Figure 17 shows the XSensor header sent after the XMesh header. It has a payload length of 4 bytes and includes:

- Sensor board ID (1 bytes) – Identifies the sensor board attached to the mote.
- Sensor packet ID (1 bytes) – indicates how the sensor readings are organized in the subsequent payload.
- Parent (2 bytes) – Identifies the uplink of the mote.

5. MTS310 Payload

Bytes: 5	0/7	4	2	2	2	10	2
TinyOS Header	XMesh Header	XSensor Header	Voltage	Thermistor	Light	…	CRC
			MTS310 Payload				

Figure 18. XSensor MTS310 Header. From Ref [26].

The size of the payload length depends on the sensor board used in the mote. In this experiment, a MTS 310 sensor board is used and it has a payload length of 16 bytes. Figure 18 and Table 1 shows the sequence and details of the sensor data payload respectively.

Type	Field Name	Description
uint16_t	**voltage**	Battery reading (also known as vref or voltage)
uint16_t	**thermistor**	Thermistor sensor reading
uint16_t	**Light**	Light sensor reading.
uint16_t	**microphone**	Measurement of acoustic range and recording.
uint16_t	**Accel_x**	Acceleration reading on the x –axis.
uint16_t	**Accel_y**	Acceleration reading on the y –axis.
uint16_t	**mag_x**	Magnetic reading on the x-axis.
uint16_t	**mag_y**	Magnetic reading on the y-axis.

Table 1. MTS310 Data Payload Contents. From Ref [26].

6. Cyclic Redundancy Check (CRC)

The packet ends with a CRC field of 2 bytes which is calculated on the entire packet and excludes the packet header and the CRC field itself. The XOR operation is used with the current byte with a shifted CRC accumulator to obtain the CRC value.

B. ACTIVE MESSAGE (AM)

A node will listen for transmissions during its wake up period. Upon detection of a preamble, it will start to receive data. It will first start to identify the sequence number of the packet to ensure there is no duplication of packets. It will next check if the group ID of the packet matches with the node. It will not participate in others' group ID transmissions and drop the packet if the group ID is different. Next, the active message (AM) field is decoded and data of the packet is processed according to the AM type specified.

AM has been historically used as a service and application identifier in TinyOS. The idea of AM has a very similar concept as application port numbers in the Internet world. It allows users to route and differentiate messages to different services in their application. Table 2 shows some of the AM types used in XMesh.

Name of AM Type (#Define)	Value of AM_Type	Description
AM_HEALTH	3	Reserved for Health packets from the mote
AM_DATA2BASE	11	Upstream data msg from node to base, no end-end ack
AM_DATA2NODE	12	Downstream data to node
AM_DATAACK2BASE	13	Upstream guaranteed delivery to base
AM_DATAACK2NODE	14	Downstream guaranteed delivery to node
AM_ANY2ANY	15	Any to any
AM_MGMT	90	OTAP status message
AM_BULKXFER	91	OTAP transfer fragment
AM_MGMTRESP	92	OTAP status acknowledgement
AM_TIMESYNC	239	Time Sync packet
AM_PREAMBLE	240	Low power preamble packet
AM_FASTJOIN	241	Fast join
AM_FASTJOINRESP	242	Fast join
AM_DOWNSTREAM_ACK	246	Ack message from base to node
AM_UPSTREAM_ACK	247	Ack message from node to base
AM_PATH_LIGHT_DOWN	248	Light full power path down to node
AM_PATH_LIGHT_UP	249	Light full power path up to base
AM_MULTIHOPMSG	250	Route update message
MODE_ONE_HOP_BROADCAST	251	Single-hop service via XMesh
AM_HEARTBEAT	253	Base station heart beat message

Table 2. Active Message Type Used in XMesh. From Ref [21].

When running the experiment with MICAz fixed with a MTS310 sensor board in a network, four AM types were captured. These four types of AM are type 3 (Health), type 11 (DataUp), type 246 (Acknowledgement) and type 250 (Router). These AM packets however might be implemented in a different way depending on the vendor and developer of the application.

When motes join a network, they will start by sending data packets of AM type 11 through a broadcast address. Once the base receives the broadcast message, it will transmit a route update message of AM type 250 to a broadcast address too. This router update message consists of the following [21]:

- Parent ID: Shows the uplink of the mote. If the mote has not joined the network, 0xFFFF is shown.
- Cost: Tells other motes the path cost to send a packet upstream to the base station. Packets will be sent to the motes with the lowest path cost.
- Hop count: Indicates the number of hops to send a message to the base station.
- A list of qualified neighbors and their quality estimate. A maximum of five neighbors will be included due to the size of the TinyOS packet size. Motes with a received estimate of 100 for high power mode and 10 for low power mode are considered qualified neighbors.

Once the sensor nodes receive the router update message, the sensor nodes will send another router update message to a broadcast address. This will complete the router table update and the sensor node will be able to start sending data to the base after this round of "handshaking."

All nodes will be able to make use of the route message sent by the base station and build their routing table. In XMesh, sensor nodes will be able to have up to sixteen neighbors, whereas a base station can have up to forty neighbors in their routing table [21]. A mote can have a descendant or downlink of fifty for a sensor mote and hundred for a base station. The route update message is set to broadcast an address at every route update interval (RUI). The default value for XMesh is 36 seconds for high power and 360 seconds for low power. This is further randomized by multiplying a random number between 0.9 and 1.1.

Health statistic packets corresponding to AM type 3 are sent after the links are established. Health statistic packets allow users to monitor the health of the network. They are sent from the sensor motes back to the base for monitoring of network health. There are two types of health packets, namely statistics health packet and neighbor health packet. They have the same header and are transmitted in an alternate fashion. The statistic health packet includes statistics like number of packets sent since boot time, voltage reading, and accumulated power usage in low power, parent ID, link quality and Received Signal Strength Indication (RSSI) value. The neighbor health packet contains information about the mote's neighbors and radio link to these neighbors. Although motes can have up to sixteen neighbors in their routing table, health packets can only send information on five neighbors back one at a time. A different packet has to be sent if

there is information from more than five neighbors in the motes. If there is only a parent with no neighbors, this packet is not sent. Figure 19 shows the sequence of the health packets for a mesh with ten neighbors in the routing table.

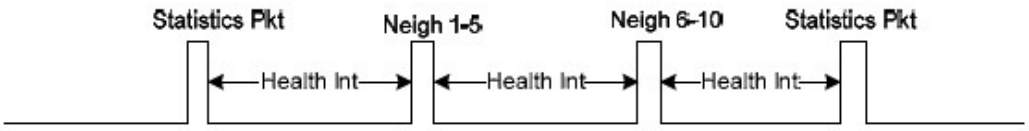

Figure 19. Sequence of Health Packets Sent in a Mesh with ten Neighbors. From Ref [26].

XMesh provides two levels of quality of service (QoS). Best effort mode uses a link level acknowledgement packet of AM type 246 to ensure that packets are delivered. This is done by retransmitting a packet until an acknowledgement is received from the receiver. However this does not guarantee that a multi-hop message was transmitted successfully from the source to the base station (or vice versa). This is often used in cases where low energy operation is required and some loss of data is acceptable. End-to end acknowledgement is used together with link level acknowledgement to achieve end-to-end QoS. This naturally consumes more energy and data bandwidth for transmission. This is normally used for health packet transmission.

C. SELF-SIMILARITY

To ensure quality of service in a data networking environment, many aspects depend on coping with peaks that might cause rare events [27]. These events can include packet loss, queue overflow, and violation of delay bounds in video traffic. Poisson and exponential distributions are commonly used for queuing studies as they are well behaved. As these distributions are stateless and peaks are not sustained, queues for the traffic do not fill up. The Poisson process models traffic as having clustering effects during short term periods and smoothing out over long periods. However, using Poisson modeling will not be accurate to represent most data traffic. Studies have shown that data traffic is more self-similar than Poisson or exponentially distributed traffic [28], [29], [30].

Self-similarity is a phenomenon that describes a particular process which appears the same when viewed at different magnifications of time scales [29]. These processes tend to exhibit persistence in clustering which implies that clustering occurs at different time scales. It also has the characteristic of traffic with bursty throughput over long periods of time. Self-similarity modeling will give a better and more accurate traffic representation as it takes into account the burstiness of data traffic. This study will help to determine if wireless sensor network traffic is self-similar and if network capacity needs to be increased. If the traffic is self-similar, buffer size can be better designed from the forecasted traffic workload.

1. Hurst Parameter

To determine the extent of self-similarity in WSN traffic, the Hurst parameter, *H*, is used. *H* also calculates the length of the long range dependence of a stochastic process. If *H* is between the values of 0.5 to 1.0, the traffic is self-similar. If *H* is equal to 0.5, it indicates an absence of self-similarity. Larger values of *H* will suggest higher degrees of persistent variability. When *H* is equal to 1, it indicates that the traffic has infinite long range dependence. To determine if a distribution has self-similar properties, four methods are commonly used [31]. The first method is using a variance-time plot, which relies on the slowly decaying variance of a self-similar series. The second approach uses an *R/S* plot, where the *R/S* statistic grows according to a power law with exponent *H* as a function of the number of points included. Next, the periodogram method uses the slope of the power spectrum of the series as frequency approaches zero to obtain *H*. Lastly, a more complicated method called the Whittle estimator can provide a more accurate result but an underlying stochastic process needs to be supplied.

The variance-time plot will be used for the analysis in this study due to its simplicity in understanding the results through graphical representation. A discrete random process x is said to be exactly self-similar with parameter β ($0 < \beta < 1$) if for all $m = 1,2,3 \ldots$ we have

$$\mathrm{var}(x^{(m)}) = \frac{\mathrm{var}(x)}{m^{\beta}} \tag{1}$$

Taking the log of both sides, the result will be

$$\log[\mathrm{var}(x^{(m)})] = \log[\mathrm{var}(x)] - \beta \log(m) \tag{2}$$

By plotting log(var($x^{(m)}$) against log(m) (where m is the aggregated time series), the approximate value of β, the gradient, will be obtained. The Hurst parameter (H) can then be calculated using the equation:

$$H = 1 - \frac{\beta}{2} \tag{3}$$

If the Hurst parameter lies between 0.5 and 1, the traffic distribution is considered to be self similar.

2. Self-Similarity Analysis

Two very important parameters to understand and model network traffic are the interarrival time and the packet size parameters. The interarrival time is defined as the time difference between two adjacent packets. This can be obtained from the elapsed time column of the XSniffer output. Further processing needs to be done to calculate the interarrival time between packets. This will be discussed in the following paragraph. The packet size length can be directly extracted from the Xsniffer output under the column "Len."

In the self-similarity discussion in Chapter V, two Mathcad scripts written by Lt. James Young from the Naval Postgraduate School are used. The first script computes the mean, variance and covariance of both the interarrival time and packet size. It also calculates the variance for five different time aggregates, namely 10, 32, 100, 320 and 1000 seconds, in order to extract the Hurst values. The second script calculates the histogram distribution of the interarrival time and packet lengths and exports it to a Microsoft Excel file. This file is used to plot the traffic distribution in Chapter V.

D. SUMMARY

This chapter discussed the various parameters used in this thesis. With the knowledge of the message flow, header field and active message used in each packet, a better understanding of the traffic can be achieved. This is followed with a basic description of the self-similarity concept, which will be used in Chapter V. The next chapter will illustrate in detail the configuration of hardware and software used in the experiment.

IV. EXPERIMENT SETUP

The hardware and software used in the experiment are described in this chapter. Physical placing of the nodes and overview statistics of the data are documented in this chapter. Detailed configurations and explanations are given as to why certain options are used.

A. EXPERIMENT SETUP

The experiment is conducted in a computer lab where the motes are placed on the floor with direct line of sight to the next higher hierarchical mote. The network consists of a base station mote, six sensor motes, and a sniffer mote.

1. Hardware

The sensor motes used are the MICAz motes together with a MTS310 sensor board. This is to simulate a real world collection of sensor data and sending back to the base station either through direct link or multi-hopped transmission. All sensor motes are downloaded with XMTS310CB_2420_hp.exe, a Crossbow application which collects the sensors data from the MTS310 board and transmits back to the base station using XMesh, the Crossbow routing protocol. The sensor motes are numbered mote one to six for easy identification. The base station mote is loaded with XMeshBase_2420_hp.exe, an application for the mote to be a sink and collect the sensor data from the sensor motes. The base station needs to be setup with a node ID of zero. Sensor data can be retrieved from the base station or sink with either wired or wireless transmission to a collection terminal. As the main purpose of this study is not to collect sensor data, the base station is not connected back to a data collection terminal.

Lastly, a XSniffer mote is setup and placed near the base to capture all incoming and outgoing traffic from the base station mote. This XSniifer is comprised of a Micaz connected to a MIB520 gateway to send the data back to a collection terminal. The collection terminal in this study is a laptop with a sniffer program, XSniffer, by Crossbow Technology. The sniffer motes need to be downloaded with the application Xsniffer.exe. All motes are powered with two AA batteries except the sniffer mote which is powered from the laptop USB connection.

2. Topology

Two commonly used topologies are setup in this experiment to simulate as close as possible to real world deployment.

a. Direct Connection to Base Mode

The first setup is a direct connection to base station mode. In this mode, six motes with sensors are placed 15 cm (0.5 feet) away from the base station to have a direct connection as shown in Figure 20, and sensor motes are placed 15 cm (0.5 feet) away from each other to reduce RF interference. Motes are stationary for the duration of the experiment. They are set close to the base to ensure that there are direct communications and to prevent lost packets during transmission. Without dealing with retransmission, more data can be collected over a longer period due to the longer operational life. The total numbers of packets collected is 557,629 over a period of 42 hours and 48 minutes for the first simulation. Another two simulations of 40 hours each were conducted to validate the first data set.

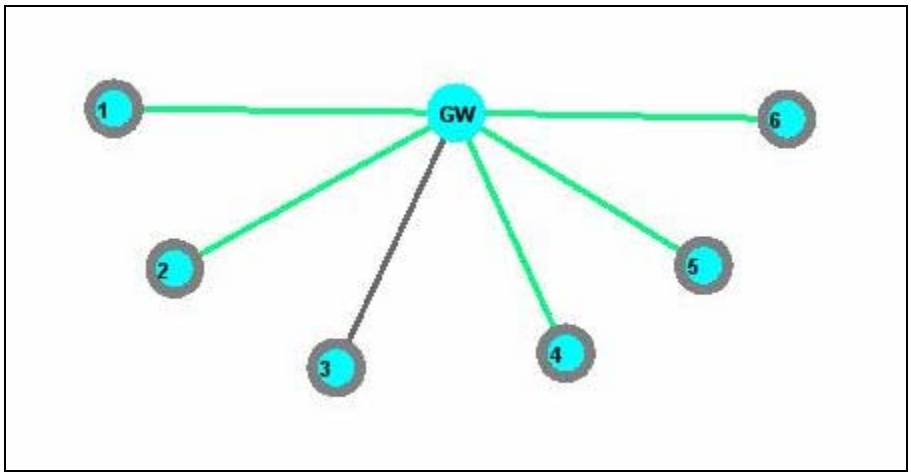

Figure 20. Direct Connection to Base Mode (Taken From MoteView Screenshot).

b. Daisy-chain Connection to Base Mode

The second setup is a daisy-chain connection to the base station as shown in Figure 21. In this setup, six motes with sensors are placed 45 cm (1.5 feet) away from each other in a straight line to ensure that a daisy-chain connection is formed. Motes ID *N* are only allowed to transmit data through mote ID (*N*-1) to force motes to do multi-hop routing. Motes are stationary for the duration of the experiment as in the first setup. This layout will ensure that each mote's traffic will only be routed through the mote in front of

it back to the base. Motes are placed in the order of ID number, meaning mote ID one is the one nearest to the base station followed by mote ID two, and mote six is the furthest away from the base station. The total numbers of packets collected is 272,410 over a period of 20 hours and 37 minutes for the first simulation. Another two simulations of 18 hours each were also conducted to substantiate the first data set.

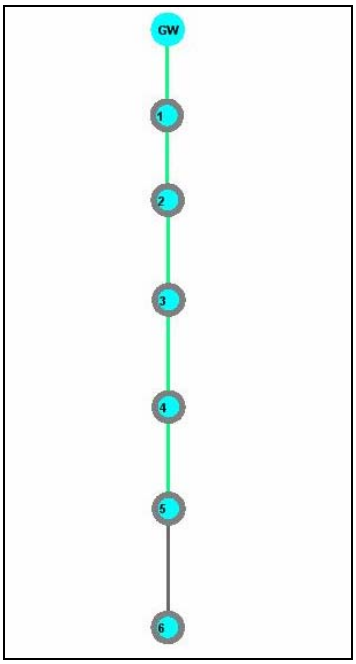

Figure 21. Daisy-chain Connection to Base Mode (Taken From MoteView Screenshot).

3. Parameters Setting

All motes are configured to be in the default group ID of 125 with RF channel of 26 (2480 MHz). The selection of group ID number is inconsequential as long as all motes are in the same group. The RF channel of 26 is chosen to eliminate any interference with other RF transmission, especially 802.11 standards. Figure 22 shows that channel 25 and channel 26 are non-overlapping channels with the 802.11b standard.

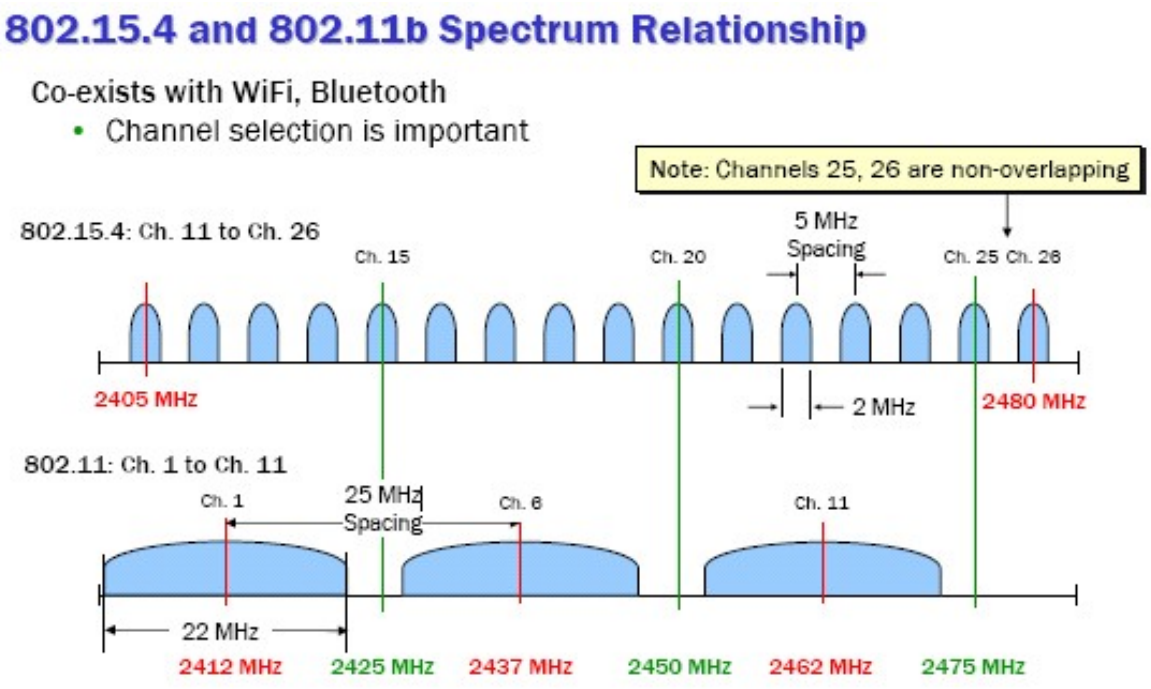

Figure 22. 802.15.4 and 802.11b Spectrum. From Ref [19].

Motes are set with the least RF power of -25 dBm to reduce the need to spread the motes over a large area. This is especially important when setting the daisy-chain topology. With a higher RF power, motes need to be placed further apart to prevent it from sending data through the mote other than the one in the next hierarchy. This setting also helps to increase the operational life of the motes with the reduced power usage. All motes are programmed to be in high power mode to stress test the motes and generate higher data traffic. In high power mode, route updates are set to send every 36 seconds. If set at low power mode, the route updates are sent every 360 seconds. All MICAz motes need to run on TinyOS 1.1.7 or a higher version.

B. SUMMARY

The configuration of hardware and software used in the experiment were explained in this chapter. These include details on the motes' placement to achieve the required topology and RF channel selection to prevent interference. These settings will allow easy replication of the experiment for further study. The next chapter will present the traffic collected and follows with discussion and analysis of the captured data.

V. TRAFFIC PROFILES AND DISCUSSION

In this chapter, the measured data are tabulated and presented in a graphical form to visualize the results. Traffic analyses are done based on the two topology setups. The results are broken down based on a percentage basis of different packets in transmission, behavior of packet transmission, statistical estimation of self-similarity, and packet distribution based on packet size and interarrival time.

A. PACKET TRAFFIC ANALYSIS

Data traffic collected is broken down to different types of packets and their respective percentage in the total transmission for analysis. A comparison is done to analyze the differences in different topology setups.

1. Direct Connection to Base Setup

The setup is described in Chapter IV. A total of three runs were conducted, with each run taking about forty hours. The number of packets collected averaged about 541,000 per run over the three simulations. Table 3 below shows the actual number of packets for all three simulations.

	Total	Hlth	Datup	AckDwn	Rte
First run	557629	15075	497236	15104	30214
Second run	532733	14312	475857	14359	28205
Third run	532889	14300	475923	14411	27989
Average	541083.7	14562.33	483005.3	14624.67	28802.67
Percentage(average)		0.026913	0.892663	0.027028	0.053231

Table 3. Number of Packets Captured for Six Motes Connecting Directly to Base Station.

Figure 23 below shows the average percentage of the types of packets captured during the experiment. It is shown that the health packet (corresponding to active message type 3) takes up 2.69% where AckDwn packets (corresponding to active message type 246) occupy about the same percentage of traffic with 2.70%. Data packets (corresponding to active message type 11) take up the majority of the transmission bandwidth with 89.26%. Rte packets (corresponding to active message type 250) take up 5.32% of the total transmission.

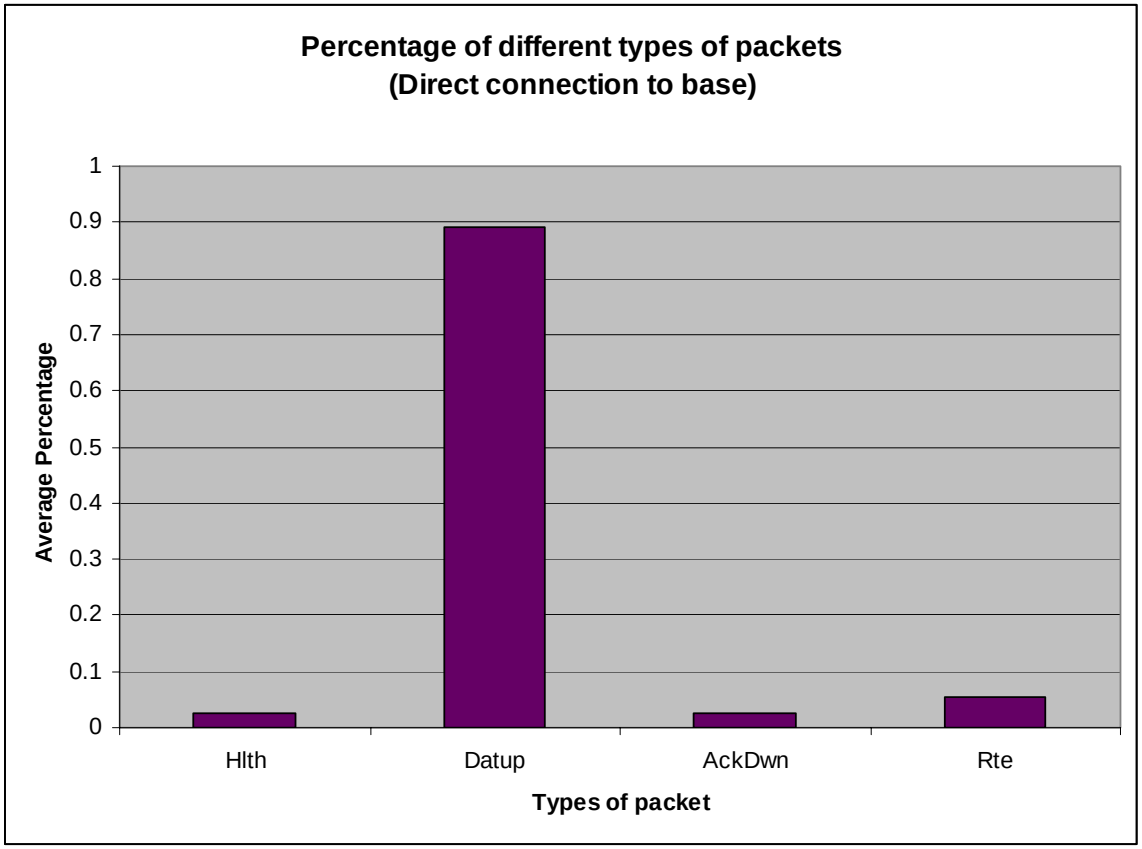

Figure 23. Percentage of Different Types of Packets in a Transmission (Direct Connection to Base Mode).

2. Daisy-chain Connection to Base Setup

A total of three runs were conducted in this setup, with each run taking about eighteen hours. The average number of packets collected is about 208,000 per run over the three simulations. The actual data is shown in Table 4.

	Total	Hlth	Datup	AckDwn	Rte
First run	143850	3933	130347	7367	2203
Second run	208304	5752	188214	11119	3219
Third run	272410	7470	246765	14018	4157
Average	208188	5718.333	188442	10834.67	3193
Percentage(average)		0.027467	0.905153	0.052043	0.015337

Table 4. Number of Packets Captured for Six Motes Connecting in Daisy-chain to Base Setup

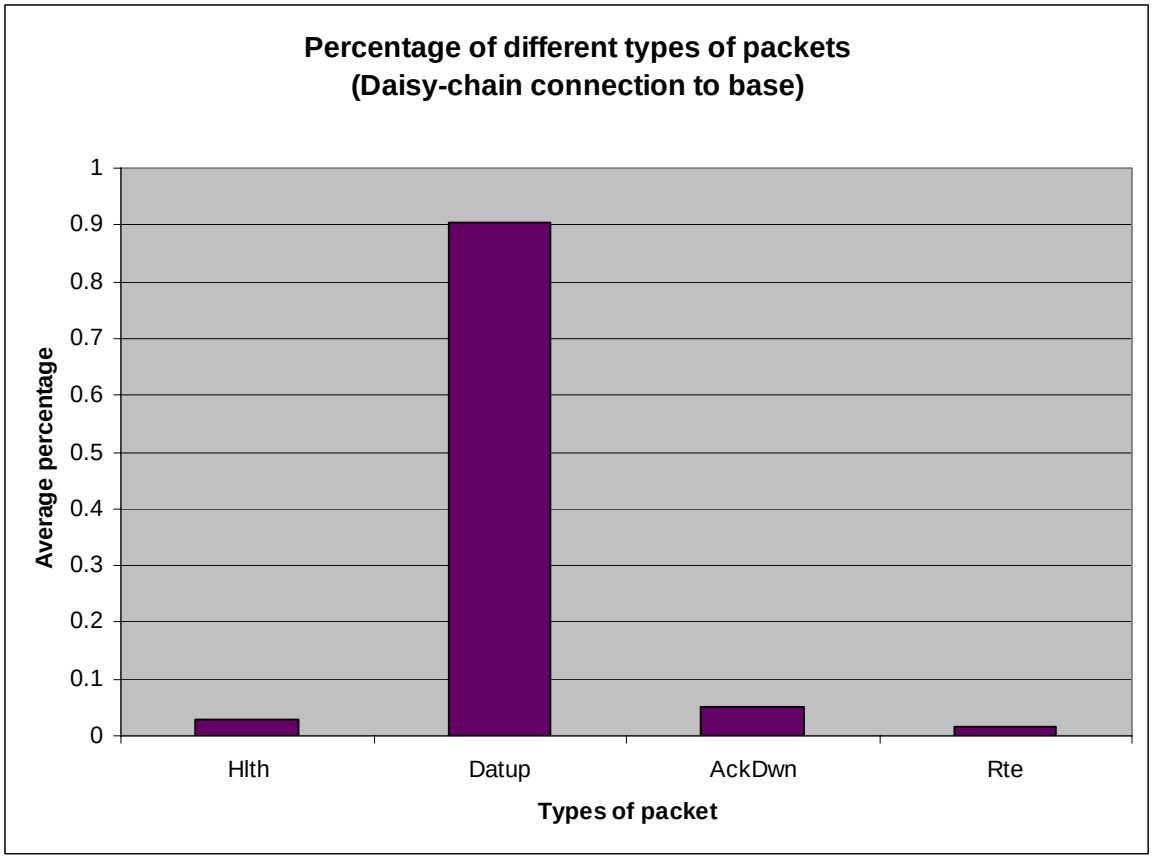

Figure 24. Percentage of Different Types of Packets in a Transmission (Daisy-chain Connection to Base Mode).

Figure 24 shows the average percentage of the types of packets captured during the experiment. From Figure 24, it is shown that health packets (AM = 3) take up 2.75% and AckDwn packets (AM = 246) have a traffic of 5.20% out of the total transmission. Data packets (AM = 11) take up 90.52% of the transmission bandwidth, while 1.53% of the packets transmitted are for the Rte packets (AM = 250).

3. Comparison of Direct and Daisy-chain Connection to Base Setup

Figure 25 below shows all six runs of direct connection and daisy-chain connection to the base station. It is observed that there are not many differences in the Hlth and Datup type of packets in terms of percentage of the total transmission. However, it shows that direct connection shows a lower percentage of AckDwn packets when compared with the daisy-chain connection. In direct connection mode, the percentage of

AckDwn packets is only 2.70% where in daisy-chain mode, it rises to 5.32% on average. This difference is due to the additional AckDwn packets from mote 1 to mote 2 in the daisy-chain connection mode.

Another difference is in the Rte packet where in direct mode, it occupied 5.32% whereas in daisy-chain mode, Rte packets only occupied 1.53% of the total transmission. This is due to the fact that in direct connection mode, the Rte packets are broadcast from all six sensor motes and base. In the daisy-chain connection mode, only sensor mote 1 and base's Rte packets are captured.

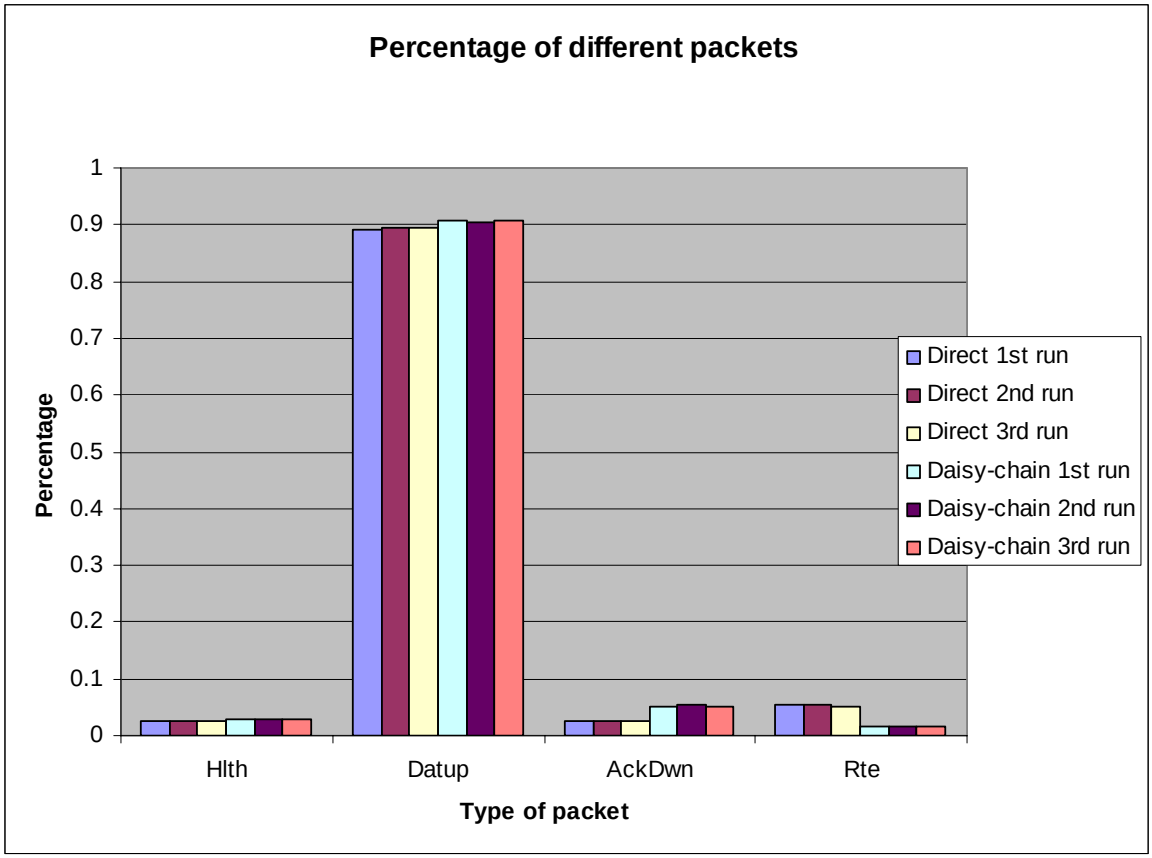

Figure 25. Percentage of Different Types of Packets in a Transmission (Combination of Direct Mode and Daisy- chain Mode).

B. BEHAVIOR OF PACKETS

A few patterns were observed during the analysis of the data collected. Detailed analyses are described in the following paragraphs for both topology setups. Data are taken from the elapsed time, destination address, AM type, length, source and originating fields in the captured file.

1. Direct Connection to Base Setup

During the initial wireless sensor network setup, sensor motes start by sending broadcast messages to build their routing tables. The active message type is DatUp (AM = 11) with a length of 27 bytes. The source and originating address was sent from the respective sensor mote which broadcasts packets. In this topology with six motes, it took about thirty three seconds for the first motes to start sending packets to base and about one minute for all six motes to start sending packets to the base. One thing to note is that once the route to base is known, no more broadcasts using DatUp (AM = 11) will be sent.

Rte (AM = 250) packets are broadcast from the base after about thirty seconds. They are sent at an average of every 34.5 seconds, both from base and sensor motes to a broadcast address. All Rte packets are sent with a packet length of 27 bytes. This is a good indication to detect if a mote is still operational as it serves as a keep-alive signal.

Two packets need to be sent before the sensor motes can start sending DatUp (AM = 11) packets. They are Rte packets broadcast from the base and Rte packets broadcast from the sensor motes. DatUp (AM = 11) will then be sent to base with the packet length of 27 bytes from respective sensor motes. They are sent at an interval of about 1.829 seconds in this application.

There are two types of Hlth (AM = 3) packets in this application. They are of two different packet lengths of 25 bytes and 29 bytes sent in an alternate fashion. The time interval between the two packets is one minute. These statistics are sent to the base from the respective sensor motes.

The last type of AM in this application is the AckDwn (AM = 246) packet. This message is sent after every Hlth packet. It is sent from the base to the respective sensor motes that sent the Hlth packet. As it is responding to the Hlth packet motes, AckDwn

packets are sent at an interval of one minute in this application. They will be sending at the same interval as the Hlth packets if the Hlth packet parameters are changed in the application.

Table 5 below summarizes the traffic flow discussed in the above paragraph.

Destination	Active Message	Packet Length	Source Mote	Originating Mote
Broadcast	DatUp (AM = 11)	27	Base	Base
Broadcast	DatUp (AM = 11)	27	Sensor motes	Same as source mote
Broadcast	Rte (AM = 250)	27	Base	Base
Broadcast	Rte (AM = 250)	27	Sensor motes	Same as source mote
Base	DatUp (AM = 11)	27	Sensor motes	Same as source mote
Base	Hlth (AM = 3)	29	Sensor motes	Same as source mote
Base	Hlth (AM = 3)	25	Sensor motes	Same as source mote
Respective sensor mote which sent the Hlth packet	AckDwn (AM = 246)	10	0	Respective sensor mote which sent the Hlth packet

Table 5. Traffic Flow Matrix for Direct Connection Setup.

2. Daisy Connection to Base Setup

In daisy-chain connection setup, as the sniffer only has "visibility" from the base and sensor mote one, broadcast messages with DatUp message and packet length of 27 are captured in the initial phase. In this topology with the same number of motes as in the first setup, it took about forty seconds for mote one to start sending packets to base. Similar to the direct connection setup, broadcasts using DatUp (AM = 11) are not sent after the initial setup.

There are two types of Rte message packet lengths for a daisy-chain connection. Rte messages from base are sent with a length of 15 bytes and Rte messages from mote one are 18 bytes long. It takes about thirty seconds, the same time as direct connection, for the base to send out the first Rte message. Base Rte messages are sent at an interval of 37 seconds and mote one Rte messages are sent at an average interval of 34.5 seconds.

Similar to direct connection setup, Rte packets broadcast from the base and Rte packets broadcast from the sensor motes need to be sent before transmission of DatUp. DatUp are sent to base with the packet length of 27 bytes from mote one. One distinctive

clue to detecting daisy-chain motes is in the originating field, where it indicates the packet source originator. If the data indicates a different mote in the source and origin field, it is sending a multi-hopped packet. There is no change in the DatUp message interval from each sensor mote when compared to the direct connection setup.

Hlth packets in a daisy-chain mode have a variation of packet length in this application. There are four types of packet lengths, namely 15, 20, 25 and 29 bytes. The Hlth packets are sent at a time interval of one minute. These statistics are sent to the base through mote one, with an indication of the source from the originating field.

The same behavior is observed for AckDwn packets when compared with the direct connection setup. AckDwn packets are sent after every Hlth packet. They are sent from the base to the respective sensor motes that sent the Hlth packet through mote one. They are also sent at an interval of one minute to each sensor mote as it responds to the Hlth packet sent by each sensor mote.

Table 6 below summarizes the traffic flow discussed in the above paragraph.

Destination	Active Message	Packet Length	Source Mote	Originating Mote
Broadcast	DatUp (AM = 11)	27	1	1
Broadcast	Rte (AM = 250)	15	Base	Base
Broadcast	Rte (AM = 250)	18	1	1
Base	DatUp (AM = 11)	27	1	Sensor mote(1-6)
Base	Hlth (AM = 3)	15	1	Sensor mote(1-6)
Base	Hlth (AM = 3)	20	1	Sensor mote(1-6)
Base	Hlth (AM = 3)	25	1	Sensor mote(1-6)
Base	Hlth (AM = 3)	29	1	Sensor mote(1-6)
1	AckDwn (AM = 246)	10	0	Respective sensor mote which sent the Hlth packet

Table 6. Traffic Flow Matrix for Daisy-chain Connection Setup.

3. Observation and Analysis

A few observations are noted in this analysis. New mote(s) joining the network can be detected when broadcasts of DatUp packets are seen. This will help in cases where detection of new or authorized motes joining the network is important. Rte packets can also be used as an indication of the mote's operational status. Once Rte packets are not received in the expected time interval, further action can be triggered or further correlation of the data can be performed.

The experiment has shown that by observing the Rte packet length, it can indicate if the mote is in daisy-chain or direct-to-base connection. This can be used to build more accurate topology maps for network management.

Another thing to note is that once the route to base is known, no more broadcasts using DatUp (AM = 11) will be sent. This is a good indication to detect if a mote is still operational as it serves as a keep-alive signal. The difference in different Hlth packet lengths will also enable the system to detect daisy-chain to direct connection setup.

C. SELF-SIMLIARITY IN WSNS

The captured data are processed through a Mathcad script as described in Chapter III to determine if the traffic is self-similar.

1. Direct Connection to Base Setup

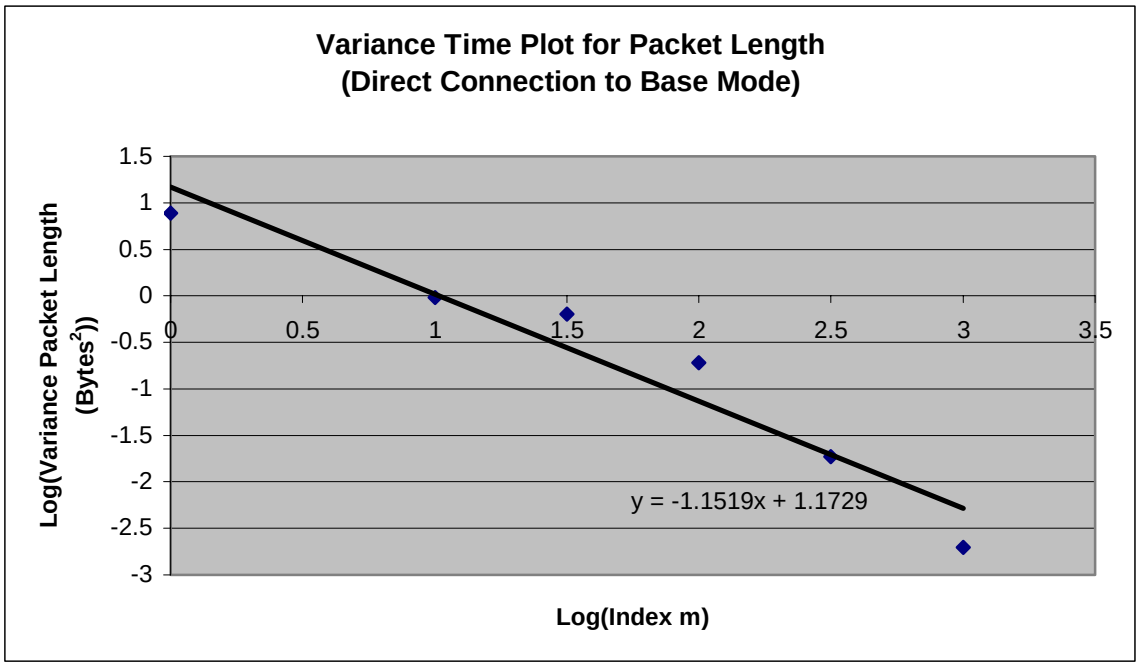

Figure 26. Variance Time Plot for Packet Length (Direct Connection to Base Mode).

Figure 26 shows the results of the variance-time plot for packet length for the direct connection to base mode. From the trend line, it is shown that the β parameter is equal to 1.1519. Using Equation (3), the H parameter is calculated to be 0.42405, which indicates that the packet length distribution does not have any indication of self-similarity.

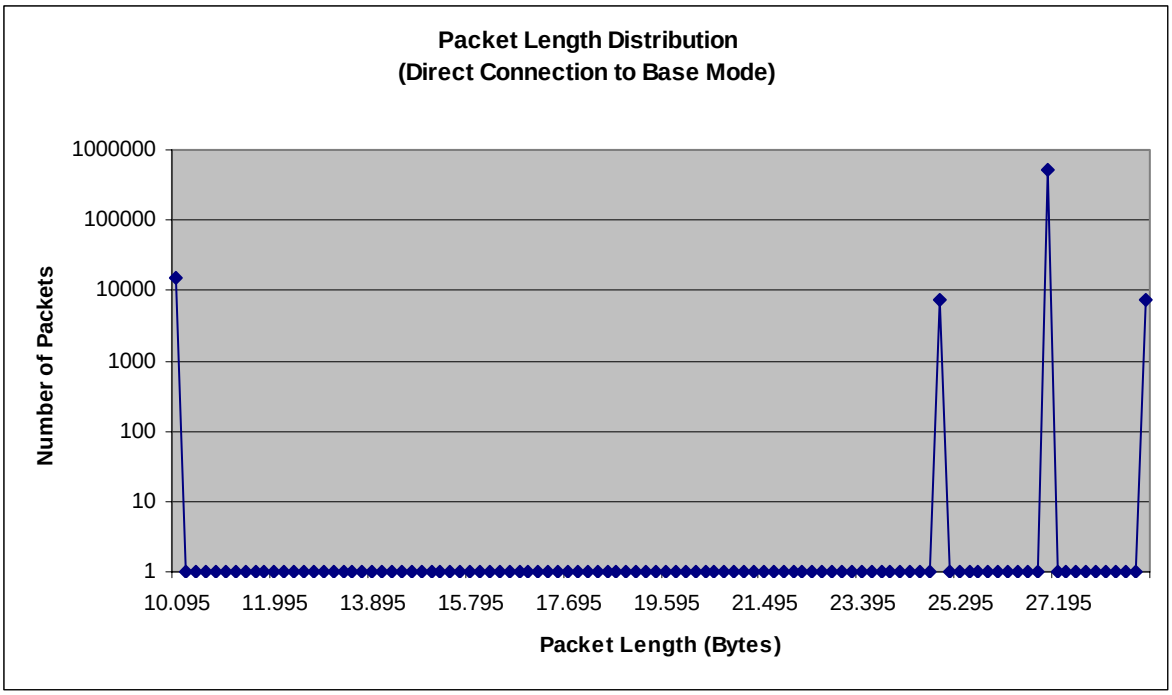

Figure 27. Distribution of Packet Based on Packet Length (Direct Connection to Base Mode).

Figure 27 shows the packet length distribution of the direct connection to base setup. Four types of packet length size are captured for the whole simulation run. There are altogether 15,104 packets with a packet length of 10 bytes, 7,539 packets with a packet length of 25 bytes, 527,449 packets with a packet length of 27 bytes and 7,536 packets with a packet length of 29. It is shown that the dominant packet is the one with a packet length of 27 bytes, which essentially is the sensor data, DatUp packet.

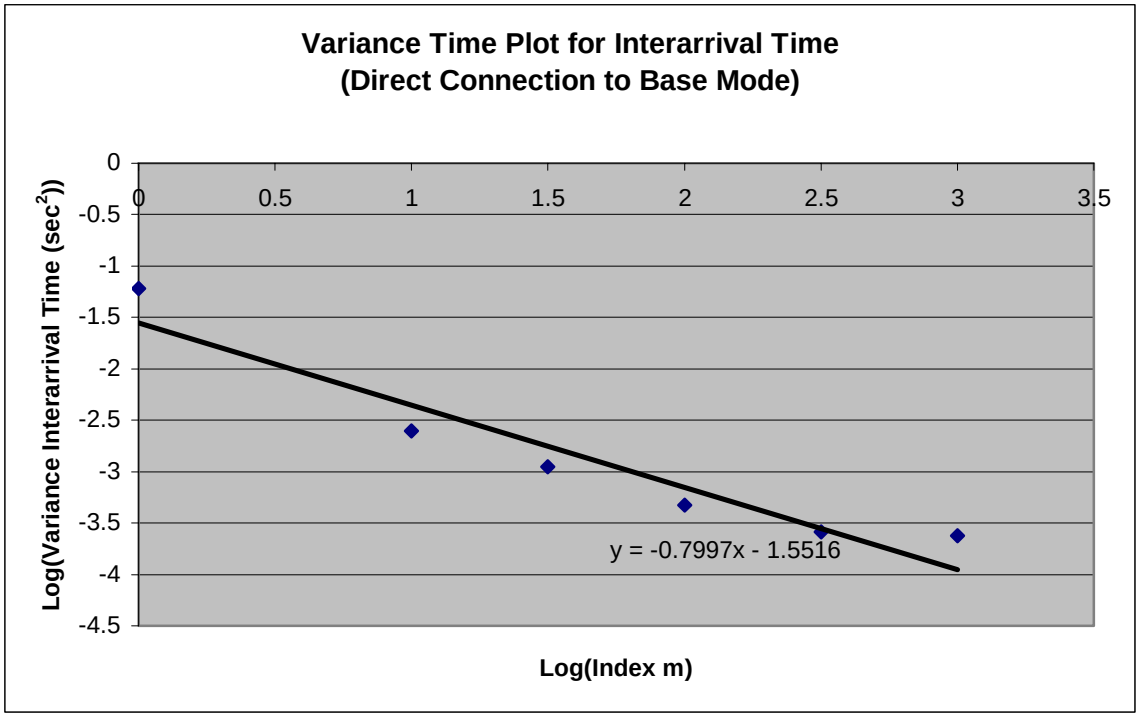

Figure 28. Variance Time Plot for Interarrival Time (Direct Connection to Base Mode).

Figure 28 shows the results of the variance time plot for interarrival time for the direct connection to base mode. From the trend line, it shows that the β parameter is equal to 0.7997. Using Equation (3), the H parameter is calculated to be 0.60015, which indicates that the packet length distribution is self-similar. This indicates that if there is an increasing (or decreasing) trend in the past, it will have an increasing (or decreasing) trend in the future. However, the strength of the correlation is not high as seen from the Hurst value.

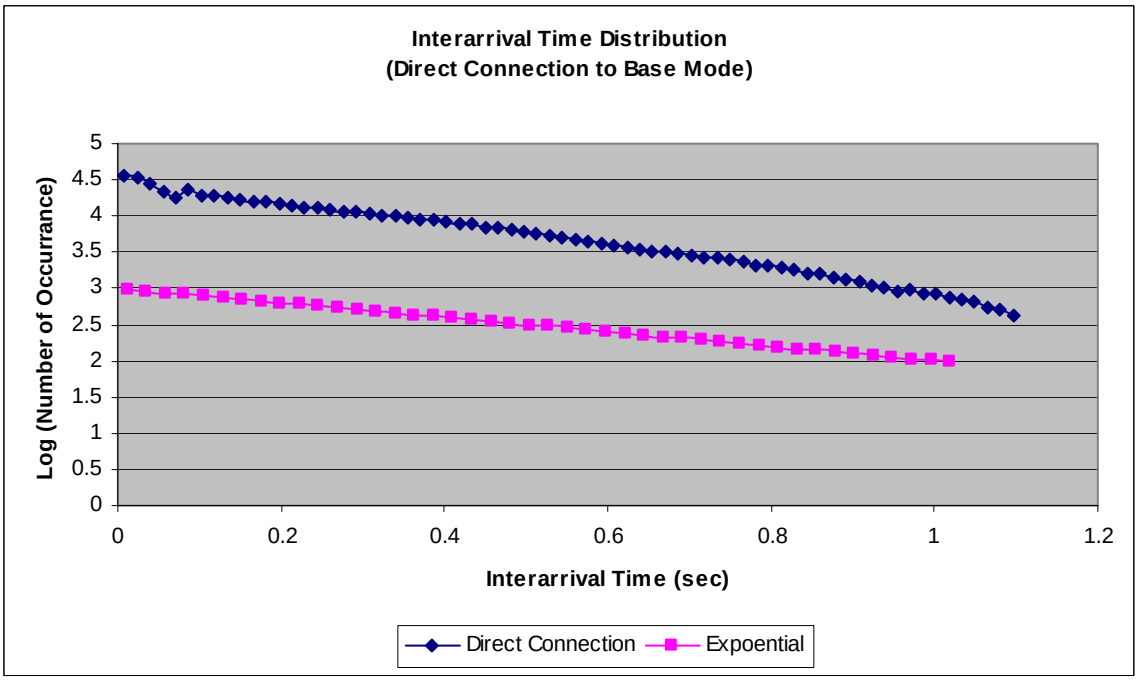

Figure 29. Distribution of Packet Based on Interarrival Time (Direct Connection to Base Mode).

Figure 29 shows the plot for interarrival time for the direct connection to base mode. By comparing the trend line with an exponential line, it shows that the interarrival time is approximately an exponential distribution.

2. Daisy-chain to Base Mode

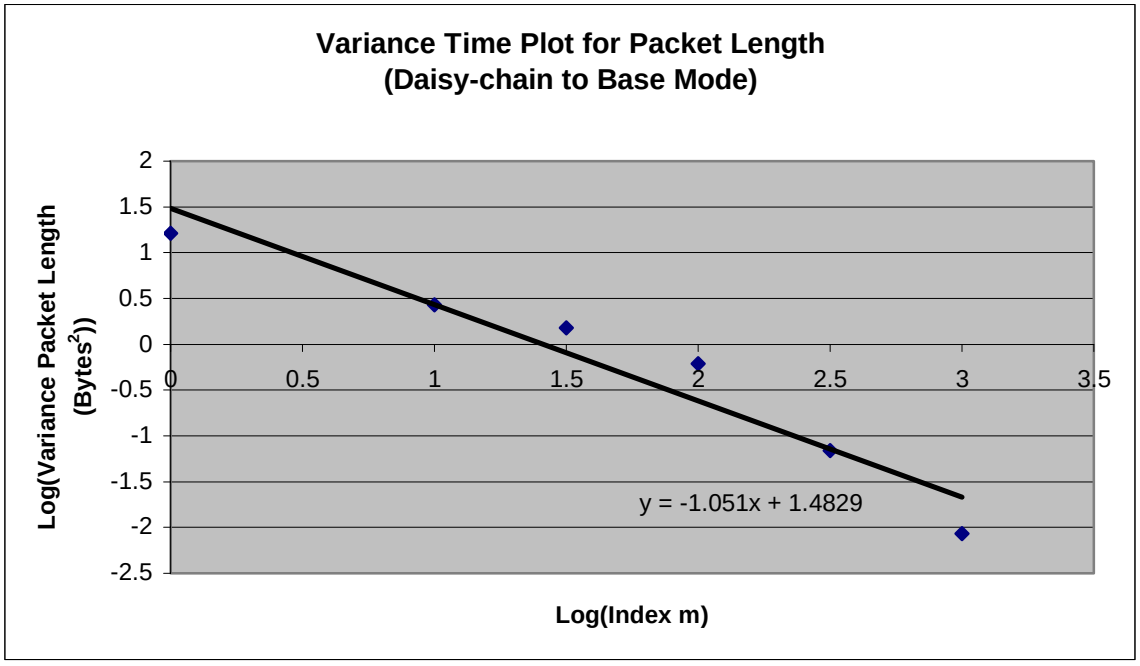

Figure 30. Variance Time Plot for Packet Length (Daisy-chain to Base Mode)

Figure 30 shows the results of the variance-time plot for packet length for the daisy-chain to base setup. From the trend line, it shows that the β parameter is equal to 1.051. Using Equation (3), the H parameter is calculated to be 0.4745, which indicates that the packet length distribution is not self-similar and not long range dependent.

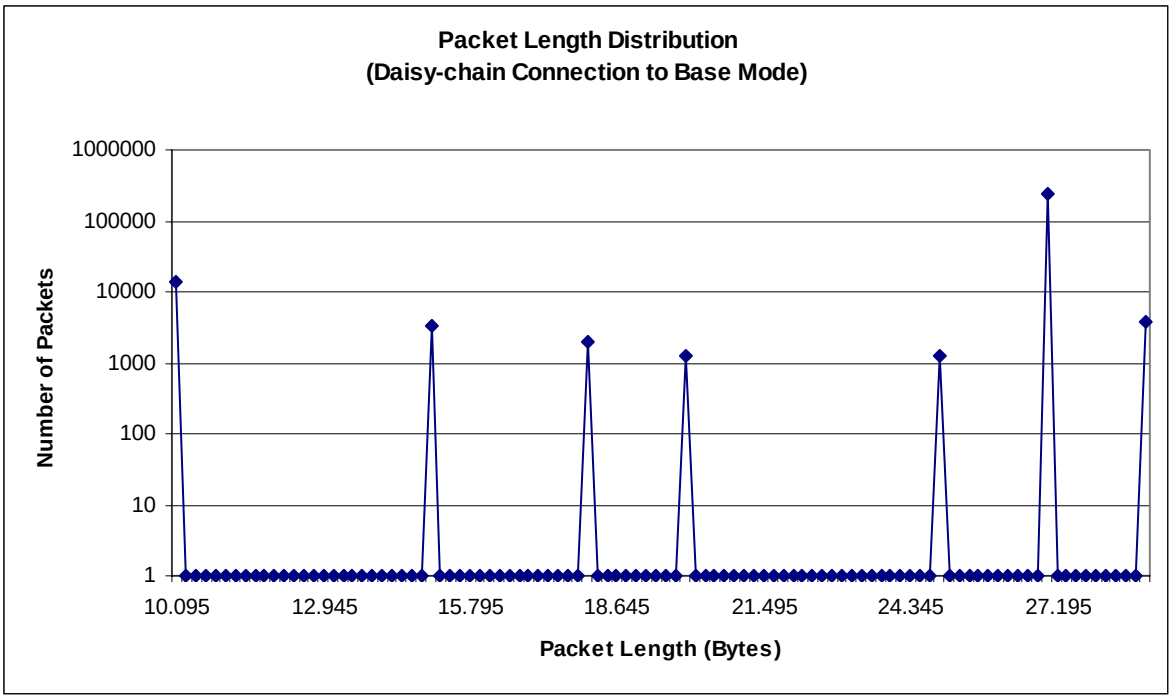

Figure 31. Distribution of Packet Based on Packet Length (Daisy-chain Connection to Base Mode).

Figure 31 shows the packet length distribution of the daisy-chain connection to base mode. It again shows the packet with length of 27 bytes dominating the distribution. However, there is a noticeable difference in the number of different types of packet length in this setup when compared to direct connection to base mode. In this setup, there are a total of seven different types of packet length instead of four different packet lengths in direct connection mode. Packet lengths of 15, 18 and 20 are not found in the direct connection setup. In summary, there are 7,516 packets with a packet length of 10 bytes, 3,355 packets with a packet length of 15 bytes, 1,987 packets with a packet length of 18 bytes, 1,250 packets with a packet length of 20 bytes, 1,270 packets with a packet length of 25 bytes, 246,635 packets with a packet length of 27 bytes, and 3,752 packets with a packet length of 29.

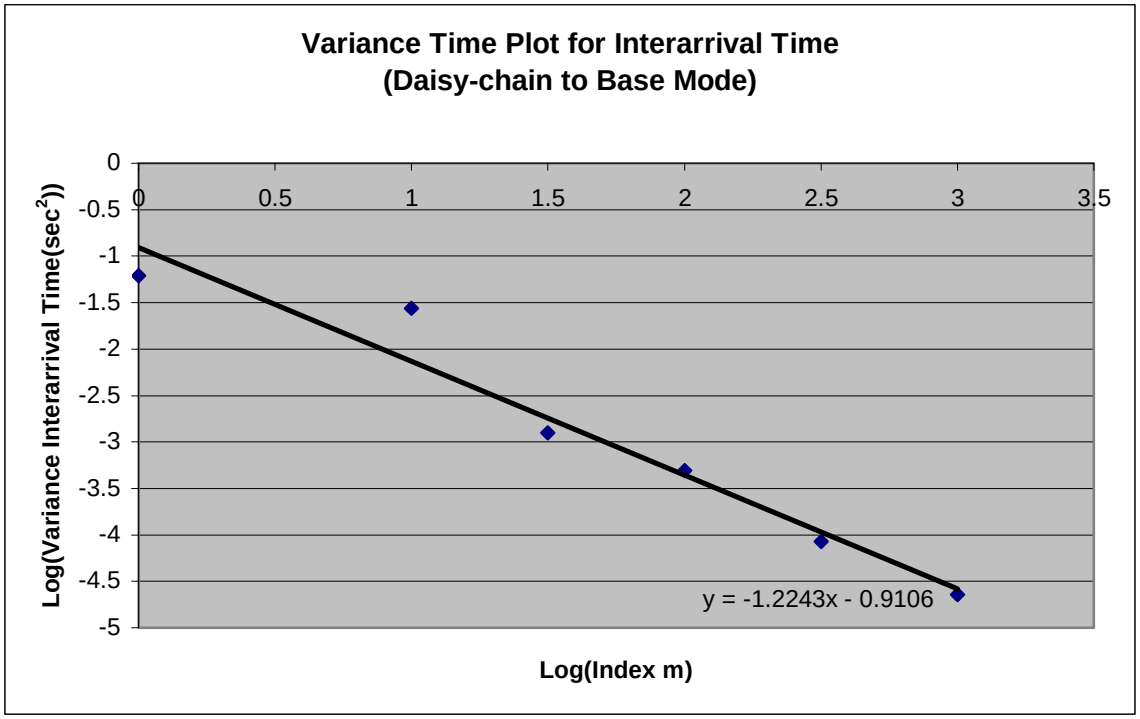

Figure 32. Variance Time Plot for Interarrival Time (Daisy-chain Connection to Base Mode).

Figure 32 shows the results of the variance time plot for interarrival time for the daisy-chain connection to base setup. From the trend line, it shows that the β parameter is equal to 1.2243. Using Equation (3), the H parameter is calculated to be 0.38785, which indicates that the interarrival time distribution does not show self-similar characteristics.

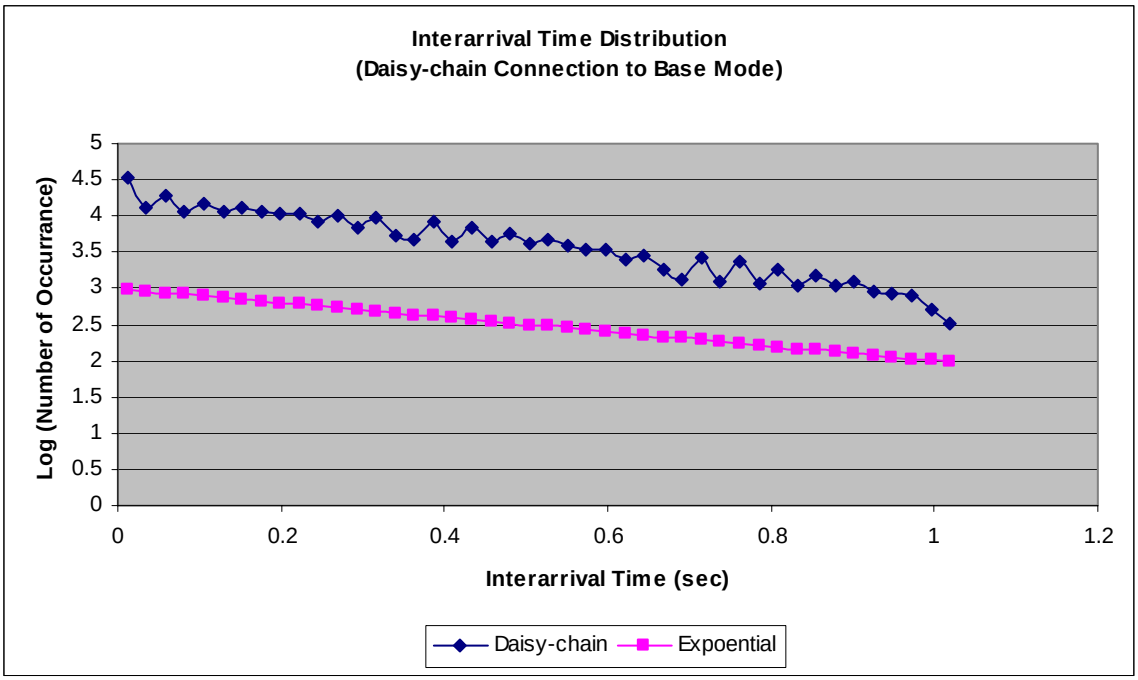

Figure 33. Distribution of Packet Based on Interarrival Time (Daisy-chain Connection to Base Mode).

Figure 33 shows the plot for interarrival time for the daisy-chain connection to base mode. From the trend line, it has the same indication with direct mode setup that the interarrival time is approximately exponentially distributed in nature.

3. Summary for Self-similar Analysis

In this section, data captured were analyzed using two Mathcad scripts for self-similarity characteristics. Variance time plots were constructed for both packet length and interarrival time to determine if WSN traffic is self-similar. The results obtained show that all except one shows no resemblance of self-similarity. Interarrival time for direct connection to base setup shows slight self-similarity characteristics as shown in the variance time plot.

Histograms were also plotted to see the distribution of packet length and interarrival time in both direct connection and daisy-chain setup. The DatUp packet has shown to be the dominant packet of the packet length distribution as in agreement with the early results in Chapter V. The interarrival time has shown a distinctively exponential distribution in both direct and daisy-chain connection setup.

D. CONCLUSION

In this chapter, the captured data were presented and analyzed. By analyzing the distribution of packets in the captured data, it is shown that it is possible to identify a direct connection from a daisy chain connection topology.

The captured packets show that it is possible to identify nodes joining the network by detecting DatUp broadcast packets. The nodes' operational status can be tracked by using Rte packets. Observing the packet length of Rte and Hlth packets will also help in differentiating between the direct connection and daisy-chain setup.

Analyses were done to determine if there is self similar characteristic in the packet length and interarrival time for both topologies. There is no resemblance of a self similar traffic in the packet length distribution. However, inter-arrival times for the direct connection setup show a slight self similarity characteristic. Packet length distribution in both topologies show DatUp packet dominates the distribution. In both configurations, the interarrival time distribution shows an exponential distribution on the inter-arrival time distribution.

In the next chapter, the results will be summarized and future work will be proposed.

VI. CONCLUSION

A. CONCLUSION

Wireless sensor networks have been proven to have great potential for many applications. As the number of WSNs deployed increase exponentially, users are faced with many unexpected and peculiar problems which are not seen in the computer simulations or lab experiments. Though computer simulations have proven useful, they are not able to capture the complex dynamics of wireless signal propagation and interference in a wireless sensor network. The difference in underlying hardware components has also been a challenge for the simulations to mimic. A physical, albeit scaled down simulation, is still desired by implementers.

Though traffic analysis in wired and wireless (IEEE 802.11) areas has matured, the unique characteristics of wireless sensor networks present a new set of challenges to the researchers and developers. Moreover, troubleshooting a wireless sensor network is several times harder and more expensive than in a traditional network given the little knowledge about the implementation.

This thesis studied the traffic generated by the wireless sensor network by setting up two network topologies, direct connection to the base and daisy-chain connection to the base. Six experiments were conducted. The data traffic among the six sensor nodes and the base node are captured over forty hours for the direct setup and twenty hours for the daisy-chain setup. Analyses are done to categorize and identify the information through anomalies and variation of traffic patterns. Data are also analyzed for self-similarity and statistical distribution.

Hlth packets in both topologies take up about 2.7% of the whole transmission. However, it shows that in direct connection, it transmits a lower percentage of AckDwn packets (2.7%) compared with daisy-chain connection percentage of 5.2%. Another noticeable difference is in the Rte packet where in direct mode, it occupied 5.32% whereas in daisy-chain mode, Rte packets only occupied 1.53% of the total transmission. This statistic can help to identify a direct connection from a daisy-chain node.

In the analysis of AM type messages, it is observed that new mote(s) joining the network can be detected when broadcasts of DatUp packets are captured. Rte packets can also be used as an indication of the mote's operational status. When Rte packets are not received in the expected time interval, further action can be triggered or further correlation of the data can be performed. The Rte packet length can also indicate if the mote is in daisy-chain or direct-to-base connection. Direct connection Rte packet length is fixed at 27 bytes whereas daisy-chain Rte packet length varies between 15 and 18 bytes. Another distinctive clue to identify daisy-chain motes is in the originating field. If the data indicates a different mote in the source and origin field, it is sending a multi-hopped packet. Another distinction between the two modes is that direct connection sends Hlth packets with a length of 25 and 27 bytes, whereas in daisy-chain nodes, Hlth packets have packet lengths of 15, 20, 25 and 27 bytes.

In terms of statistical analysis, variance-time plots were constructed for both packet length and interarrival time to determine if wireless sensor network traffic is self-similar. The results obtained show that all except one measured parameter showed no resemblance of self-similarity traffic. The only one showing self similarity of $H = 0.60$ is the interarrival time for direct connection to base setup. Histograms were also plotted to see the distribution of packet length and interarrival time in both direct connection and daisy-chain setup. The DatUp packet has shown to be the dominant packet of the packet length distribution as in agreement with the early results. The interarrival time has shown a distinctive exponential distribution in both direct and daisy-chain connection setup.

These results are important as they allow identification of the topology of a network using the captured data. The analysis also shows that new nodes joining the network can be identified and connectivity of the nodes can be monitored using Rte packets. The self-similarity study allows developers to use the exponential model with confidence. Using the right modeling for traffic distribution allows better evaluation of network capacity and allows determination of battery power based on the forecasted traffic workload.

B. FUTURE WORK

This study only covered a portion of the wireless sensor network traffic characteristics. More work can be done to further explore the inner-workings of the traffic generated by sensor networks. Some examples of future work are proposed in the following paragraphs.

1. Changing of Parameters

Given more time to conduct the research, different parameters can be changed to see if it changes or affects the traffic characteristics. Any variation to the traffic can be used to identify that particular network. One of the simple changes is to increase the number of nodes to study scalability issues in wireless sensor networks. More traffic will be generated and it will further examine the scalability of the network. Distance between the nodes can be varied to see if there is any variation to the traffic statistic to identify mobile nodes or node failure. Weak batteries can be used to simulate node failures and to study if there is any early sign that can be detected. Although there are two non-overlapping channels in 802.15.4 over 802.11, tests can be conducted in the overlapping channel to see if the interference will cause any changes in the traffic statistic. Path loss scenario due to absorbing, refracting, scattering, and reflecting can also be simulated to see if it influences the transmission of a packet.

2. Security

Various studies have been made on the security vulnerability of a WSN [32]. These concerns include eavesdropping, forgery of sensor data, denial of service attacks, and the physical compromise of sensor nodes. These attacks can be simulated and traffic captured to analyze the behavior of these attacks. With these traffic profiles and signatures, intrusion detection systems deployed in a wireless sensor network can identify known attacks accurately and quickly.

3. Correlation of Data

This thesis has focused on the packets transmitted by the wireless sensors. However, the sensor data sent can be used to correlate an event or incident together with the traffic captured. A weak battery will show low voltage in the health message and probably an end-to-end acknowledgement might not be sent during the last minute of the

failing node. With the correlation of the data comes the ability to identify a dead node due to starvation of energy instead of a damaged node in a battlefield.

With a more comprehensive set of data, networks can be identified more accurately. Design of the next generation wireless sensor nodes will also benefit from these research efforts.

LIST OF REFERENCES

[1] Information Sciences Institute, *The Network Simulator- ns-2*. University of Southern California, 2003. http://www.isi.edu/nsnam/ns/. Last accessed 15 Feb 2007.

[2] L. F. Perrone and D. M. Nicol, "A scalable simulator for TinyOS applications," in Simulation Conference, 2002. Proceedings of the 34th conference on Winter simulation: exploring new frontiers, December 08-11, 2002, San Diego, California pp 679- 687, 8-11 Dec. 2002.

[3] D. Cavin, Y. Sasson, and A. Schiper, "On the Accuracy of MANET Simulators," in *Proceedings of the Workshop on Principles of Mobile Computing (POMC'02)*, Toulouse, France, pp. 38-43, 30-31 October 2002.

[4] Cheng Kiat Amos, Teo, **"**Performance Evaluation of a Routing Protocol in Wireless Sensor Network," Master's Thesis, Naval Postgraduate School, Monterey, California, December 2005.

[5] Georgios Kirykos, "Traffic Profiling of Wireless Sensor Networks," Master's Thesis, Naval Postgraduate School, Monterey, California, December 2006.

[6] Qilian Liang, "Ad Hoc Wireless Network Traffic—Self-Similarity and Forecasting," Communications Letters, IEEE, Volume 6, Issue 7, pgs. 297-299, Jul 2002.

[7] M. Haenggi, (2005), "*Opportunities and Challenges in Wireless Sensor Networks"*. In Ilayas, M., Mahgoub, I. (Eds), "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems". Boca Raton, CRC Press, 2004.

[8] M. Ilyas (Ed.), "*Handbook of Sensor Networks: Compact Wirelessand Wired Sensing Systems"*, CRC Press, 2004.

[9] F. L. Lewis, "Wireless Sensor Networks," Smart Environments: Technologies, Protocols, and Applications ed. D.J. Cook and S.K. Das, John Wiley, New York, 2004.

[10] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, Volume 40, Issue 8, pp. 102-114, Aug 2002.

[11] Kay Romer and Friedemann Mattern, "The Design Space of Wireless Sensor Networks," IEEE Wireless Communications, Volume 11, Issue 6, pp. 54-61, Dec 2004.

[12] Karl Holger and Andreas Willig, *Protocols and Architectures for Wireless Sensor Networks*, Chapter 2 pp. 18-53, Chapter 4, pp. 86-108, John Wiley and Sons, West Sussex, England, 2005.

[13]A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", In *WSNA*, Atlanta, USA, 28 September 2002.

[14] K. Martinez, R. Ong, J. K. Hart, and J. Stefanov, "Sensor Web for Glaciers", *Proc. EWSN 2004*, Berlin, Germany, pp 56-62, January 19-21 2004.

[15] F. Michahelles, P. Matter, A. Schmidt, and B. Schiele, "Applying Wearable Sensors to Avalanche Rescue", Computers and Graphics, Volume 27, Number 6, pp. 839-847, December 2003.

[16] H. Baldus, K. Klabunde, and G. Muesch, "Reliable Set-Up of Medical Body-Sensor Networks", *Proc. EWSN 2004*, Berlin, Germany, pp. 353-363, January 19-21, 2004.

[17] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks", Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, September 23-28, 2002, Atlanta, Georgia, USA, pp. 148–159, ACM Press, 2002.

[18] The 29 Palms Experiment: Tracking vehicles with a UAV-delivered sensor network, http://robotics.eecs.berkeley.edu/~pister/29Palms0103/. Last accessed 11 Feb 2007.

[19] Crossbow Technology Inc., "Wireless Sensor Networks Seminar," San Jose, February 7-9, 2006.

[20] Crossbow Technology Inc., "MTS/MDA Sensor and Data Acquisition Board User's Manual Rev. A," January 2006.

[21] Crossbow Technology Inc, "XMesh User's Manual Revision A," March 2006.

[22] Bob Heile, "ZigBee Alliance Tutorial" ZigBee Alliance, http://www.zigbee.org/en/resources/presentations.asp. Last accessed 11 Feb 2007

[23]. IEEE Computer Society, "IEEE Standard 802.15.4-2003 (draft), Part 15.4 : Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks," IEEE Standards, October 2003.

[24] ZigBee: "Wireless Control That Simply Works," William C. Craig Program Manager Wireless Communications ZMD America, Inc., http://www.zigbee.org/en/resources/whitepapers.asp, Last accessed 11 Feb 2007.

[25] Tim Reilly, "Mote Software Development for Wireless Sensor Networks," presented at the 2nd Annual IEEE Communications Society Conference on Sensor and Ad-hoc Communications and Networks, Santa Clara, California, 26 September 2005.

[26] Crossbow Technology Inc., "XServe Users Manual Revision A," March 2006.

[27] Linington P.F., “Everything you always wanted to know about self-similar network traffic and long-range dependency, but were ashamed to ask.”, University of Kent, http://www.cs.kent.ac.uk/people/staff/pfl/presentations/longrange. Last accessed 15 Jan 2007.

[28]. Kihong Park, Gitae Kim, and Mark E. Crovella, “On the Effect of Traffic Self-Similarity on Network Performance,” *Proceedings of SPIE International Conference on Performance and Control of Network Systems*, pp. 989-996, November 1997.

[29]. William Stallings, *High-Speed Networks and Internets: Performance and Quality of Service*, Second Edition, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[30]. Matthias Grossglauser and Jean Bolot, “On the Relevance of Long Range Dependence in Network Traffic,” Networking, IEEE/ACM Transactions on, Vol.7, Iss.5, pp. 629-640*,* Oct 1999.

[31]] M. Crovella and A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes,” IEEE/ACM Transactions on Networking, Vol. 5, No. 6, pp. 835-846, December 1997.

[32] D. Djenouri, L. Khelladi, and A. N. Badache, “A survey of security issues in mobile ad hoc and sensor networks,” Communications Surveys & Tutorials, IEEE, Vol.7, Iss. 4, pp. 2- 28, 4th quarter 2005.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

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

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

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

6. Wang Wei Beng
   Defence Science & Technology Agency
   Singapore