2007-12

# Ad Hoc Network Architecture for Multi-Media Networks

Mak, Wai Y.

Monterey California. Naval Postgraduate School

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**AD HOC NETWORK ARCHITECTURE FOR MULTI-MEDIA NETWORKS**

by

Wai Yen Mak

December 2007

Thesis Advisor: Weilian Su
Second Reader: John C. McEachen

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

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

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

| | |
|---|---|
| **4. TITLE AND SUBTITLE**<br>Ad Hoc Network Architecture for Multi-media Networks | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Wai Yen Mak | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>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**<br>Approved for public release; distribution is unlimited | **12b. DISTRIBUTION CODE**<br>A |

**13. ABSTRACT**

The desire for more intelligence in the battlefield has given rise to the idea of routing video images over wireless sensor networks. This would apprise combat decision makers with actual images of battlefield developments and allow them to make sound decisions. To achieve this objective, the characteristics of video traffic must be studied and understood. This thesis focuses on evaluating the possibility of routing video images over a wireless sensor network. Video traffic is modeled and simulations are performed via the use of the Sun Small Programmable Object Technology (Sun SPOT) Java development kits configured in three different network topologies: the star topology, binary tree topology and chain topology. It is known that video traffic is self-similar and can be obtained by aggregating a large number of On-Off message sources. Hence, an On-Off model using Pareto distribution function is used to model video traffic over the network. In this thesis, four self-similar shaping parameters, i.e., $\alpha_{On} = 1.2, 1.4, 1.7 \, and \, 1.9$ are used in the simulations. The performance of each topology is evaluated based on parameters like mean throughput, mean interarrival time, mean packet drop, and mean delay.

| | |
|---|---|
| **14. SUBJECT TERMS**<br>Pareto distribution, Wireless Sensor Network, Architecture, Self-similar traffic | **15. NUMBER OF PAGES**<br>87 |
| | **16. PRICE CODE** |

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UU |

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# AD HOC NETWORK ARCHITECTURE FOR MULTI-MEDIA NETWORKS

Wai Yen Mak
Major, Republic of Singapore Armed Forces, Army
B.Eng. (Hons), University of Southampton, United Kingdom, 2001

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
(MSEE)**

from the

**NAVAL POSTGRADUATE SCHOOL
December 2007**

Author: Wai Yen Mak

Approved by: Weilian Su
Thesis Advisor

John C. McEachen
Second Reader

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

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The desire for more intelligence in the battlefield has given rise to the idea of routing video images over wireless sensor networks. This would apprise combat decision makers with actual images of battlefield developments and allow them to make sound decisions. To achieve this objective, the characteristics of video traffic must be studied and understood. This thesis focuses on evaluating the possibility of routing video images over a wireless sensor network. Video traffic is modeled and simulations are performed via the use of the Sun Small Programmable Object Technology (Sun SPOT) Java development kits configured in three different network topologies: the star topology, binary tree topology and chain topology. It is known that video traffic is self-similar and can be obtained by aggregating a large number of On-Off message sources. Hence, an On-Off model using Pareto distribution function is used to model video traffic over the network. In this thesis, four self-similar shaping parameters, i.e., $\alpha_{On} = 1.2, 1.4, 1.7 \; and \; 1.9$ are used in the simulations. The performance of each topology is evaluated based on parameters like mean throughput, mean interarrival time, mean packet drop, and mean delay.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

**I. INTRODUCTION....1**
- **A. BACKGROUND....1**
- **B. MOTIVATION....2**
- **C. THESIS GOALS AND METHODOLOGY....2**
- **D. RELATED WORK....3**
- **E. THESIS ORGANIZATION....3**

**II. WIRELESS SENSOR NETWORKING....5**
- **A. INTRODUCTION TO SENSOR NETWORKS....5**
  - **1. Sensor Node Hardware Components....6**
    - ***a. Processor....6***
    - ***b. Memory....7***
    - ***c. Transceiver....7***
    - ***d. Sensors....7***
    - ***e. Power Supply....8***
  - **2. Performance Metrics for Wireless Sensor Networks....8**
    - ***a. Latency....8***
    - ***b. Accuracy....8***
    - ***c. Fault Tolerance....9***
    - ***d. Scalability....9***
    - ***e. Energy Efficiency....9***
- **B. ROUTING PROTOCOL....9**
  - **1. Table-driven Routing Protocols....10**
    - ***a. Destination-sequenced Distance Vector (DSDV)....10***
    - ***b. Cluster-head Gateway Switch Routing (CGSR)....11***
    - ***c. Wire Routing Protocol (WRP)....12***
  - **2. On-demand Routing Protocols....13**
    - ***a. Ad Hoc On-demand Distance Vector (AODV)....14***
    - ***b. Dynamic Source Routing (DSR)....15***
    - ***c. Temporally Ordered Routing Algorithm (TORA)....18***
    - ***d. Associativity-Based Routing (ABR)....20***
    - ***e. Signal Stability Routing (SSR)....21***
- **C. NETWORK TOPOLOGY....22**
  - **1. Star Topology....23**
  - **2. Mesh Topology....23**
  - **3. Tree Topology....24**
- **D. SUN SPOT SENSOR....24**
  - **1. eSPOT Main Board....25**
  - **2. eSPOT Daughterboard (eDEMO)....26**
  - **3. Battery....26**
- **E. SUMMARY....26**

**III. EXPERIMENTAL SETUP....27**

A. GENERAL ....27
B. TRAFFIC MODELING ....27
C. SENSOR NETWORK ARCHITECTURE....29
1. Star Topology ....29
2. Binary Tree Topology....30
3. Chain Topology ....31
D. SPOT APPLICATIONS....31
1. Base Station (BS)....32
2. Free-range Sun SPOTs (SPOTs) ....33
E. PERFORMANCE PARAMETERS ....35
1. Mean Throughput....35
2. Mean Interarrival Time ....35
3. Mean Packet Drop ....36
4. Mean Delay....36
F. SIMULATION PROCEDURE....37
G. SUMMARY ....37
IV. PERFORMANCE ANALYSIS....39
A. APPROXIMATION METHOD FOR IMPLEMENTING PARETO ON-OFF ....39
1. Approximating Pareto Distribution....39
2. Accuracy of the Approximation Method ....42
B. SIMULATION RESULTS ....46
1. Star Topology ....46
2. Binary Tree Topology....49
3. Chain Topology ....52
4. Summary....56
V. CONCLUSION AND RECOMMENDATION ....59
A. CONCLUSIONS ....59
B. RECOMMENDATIONS....60
LIST OF REFERENCES....63
INITIAL DISTRIBUTION LIST ....67

# LIST OF FIGURES

Figure 1. Star Topology........................................................................................ xiv
Figure 2. Binary Tree Topology.........................................................................xv
Figure 3. Chain Topology ................................................................................. xvi
Figure 4. Sensor Nodes Scattered in a Sensor Field. (After [5])......................................6
Figure 5. Components of a Sensor Node. (After [5])........................................................6
Figure 6. Categorization of Ad Hoc Routing Protocols. (From [10]) ............................10
Figure 7. CGSR: Routing from Node 1 to Node 8. (From [6])......................................12
Figure 8. Propagation of RREQ to N8. (From [10]) .......................................................14
Figure 9. Path of RREP to N1. (From [10]) ....................................................................15
Figure 10. DSR Route Discovery. (From [10])................................................................17
Figure 11. Propagation of Route Reply with Route Record. (From [10])........................17
Figure 12. Route Creation, Using a “Height” Metric to Establish DAG (From [10]) ......19
Figure 13. Route Maintenance Process. (From [10]) .......................................................20
Figure 14. Star, Mesh and Tree Topology. (After [21])...................................................23
Figure 15. Photo of a Sun SPOT (From [7]) ...................................................................24
Figure 16. a. Sun SPOT BS Unit, b. Free-range Sun SPOT Unit and c. eSPOT Board Configuration and Connectors. (From [7]) .....................................................25
Figure 17. On-Off Traffic Model and its Parameters. (From [23])..................................28
Figure 18. Star Topology...................................................................................................30
Figure 19. Binary Tree Topology.......................................................................................31
Figure 20. Chain Topology ................................................................................................31
Figure 21. Base Station Flow Chart ..................................................................................33
Figure 22. SPOT Flow Chart..............................................................................................34
Figure 23. SPOT Packet Format.........................................................................................35
Figure 24. Error Plot for $\alpha_{on} = \alpha_{off} = 1.2$ ..........................................................42
Figure 25. Error Plot for $\alpha_{on} = 1.4\, and\, \alpha_{off} = 1.2$ ...............................................42
Figure 26. Error Plot for $\alpha_{on} = 1.7\, and\, \alpha_{off} = 1.2$ ...............................................43
Figure 27. Error Plot for $\alpha_{on} = 1.9\, and\, \alpha_{off} = 1.25$ ...........................................43
Figure 28. Close-up Error Plot of $\alpha_{on} = 1.9\, and\, \alpha_{off} = 1.25$ for *r* between 0.999 and 1.....43
Figure 29. Error Plot for $\alpha_{on}$ between 1 and 2 at *r*=0.9. .................................................44
Figure 30. Error Plot for $\alpha_{on}$ between 1 and 2 at *r*=0.999. .............................................45
Figure 31. Error Plot for $\alpha_{on}$ between 1 and 2 at *r*=0.999999 .........................................45
Figure 32. Star Topology - Mean Throughput (bytes/second) Plot. ................................46
Figure 33. Star Topology - Mean Interarrival Time (sec) Plot. .......................................47
Figure 34. Star Topology - Mean Packet Drop Plot..........................................................47
Figure 35. Star Topology - Mean Delay (sec) Plot. .........................................................48
Figure 36. Binary Tree Topology – Mean Throughput (bytes/second) Plot.....................50
Figure 37. Binary Tree Topology – Mean Interarrival Time (sec) Plot............................50
Figure 38. Binary Tree Topology – Mean Delay (sec) Plot.............................................51

Figure 39. Chain Topology (3SPOTs) - Mean Throughput (bytes/second).......................53
Figure 40. Chain Topology (3SPOTs) - Mean Interarrival Time (sec) Plot. .....................53
Figure 41. Chain Topology (3SPOTs) - Mean Delay (sec) Plot. .......................................54
Figure 42. Chain Topology (4SPOTs) - Mean Throughput (bytes/second) Plot. .............54
Figure 43. Chain Topology (4SPOTs) - Mean Interarrival Time (sec) Plot. .....................55
Figure 44. Chain Topology (4SPOTs) - Mean Delay (sec) Plot. .......................................55

# LIST OF TABLES

Table 1. Architecture Recommendation for IEEE 802.15.4 and the Zigbee Standard xvi
Table 2. Approximate Value of Respective $t_{on}$ and $t_{off}$ . ..............................................41
Table 3. Architecture Recommendation for IEEE 802.15.4 and the Zigbee Standard. ...........................................................................................56
Table 4. Topology Recommendation with Respect to Shaping Parameter....................57

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

The need for intelligence in the battlefield has grown tremendously. Today, with the help of technology, especially wireless communications, the desire to have more intelligence has given rise to the idea of routing video images over wireless sensor networks. In the future, wireless sensor networks will not only fulfill the need for real-time sensing of the environment, they will also perform the task of routing video images from the intelligence collector (the source), to the decision maker (the destination). This would apprise the combat decision maker with actual images of battlefield development and allow them to make sound decisions.

The operation of the network is divided into three phases. In phase one the intelligence collectors capture the video images using any video capturing device, and the video images go through a video compression algorithm. For phase two, the compressed video images are sent through the wireless sensor network to the destination. Finally, in phase three the destination decompresses the video images before presenting them.

This thesis focuses on evaluating the possibility of routing video images over a wireless sensor network. Video traffic will be modeled and simulations will be carried out via the use of the Sun Small Programmable Object Technology (Sun SPOT) Java development kits in three different network topologies: the star topology, binary tree topology and chain topology. Simulation data will be collected in text format and imported into Matlab for computation of performance parameters such as mean throughput, mean packet interarrival time, mean packet drop, and mean delay.

The Sun SPOT development kit is an experimental technology from Sun Labs. Each developmental kit consists of one base station (BS) and two free-range SPOTs. The main processor is an Atmel AT91RM9200 system-on-chip (SOC) integrated circuit. The main processor runs the Java virtual machine (VM) “Squawk” and serves as an IEEE 802.15.4 wireless network node.

It is known that video traffic is self-similar and can be obtained by aggregating a large number of On-Off message sources. Hence, an On-Off model using the Pareto

distribution function is useful to characterize a video application. In this thesis, an approximation method was adapted to implement the Pareto distribution function. This is due to the limitation imposed by the Java "Squawk" VM. The Java Squawk VM is a reduced set of standard Java language that does not contain the required math functions to implement the actual Pareto distribution function. The approximation method uses the integer power to compute the On-Off period of the Pareto distribution function instead of the floating point power. Using the approximation method, the maximum error between it and the actual Pareto distribution function is approximately $2.7\times10^{-5}$. Four different Pareto distribution shaping parameters, namely $\alpha_{On} = 1.2, 1.4, 1.7, 1.9$ were examined. The parameter $\alpha_{On} = 1.2$ characterizes Ethernet traffic while $\alpha_{On} = 1.9$ characterizes video traffic. These four parameters are in the range of self-similarity (i.e. $0.5 \le H \le 1$, $H$ is the Hurst parameter which is the self-similarity parameter ) with $\alpha_{On} = 1.2$ being the most self-similar and the $\alpha_{On} = 1.9$ being the less self-similar.

The star topology is configured such that all SPOTs are one hop away from the base station (BS) (see Figure 1). This topology provides very wide breadth but limited depth coverage. The simulation results show that the mean throughputs of all shaping parameters have identical characteristics but wide variations. These results do not permit a conclusive analysis. Hence, a best shaping parameter for this topology could not be clearly ascertained.

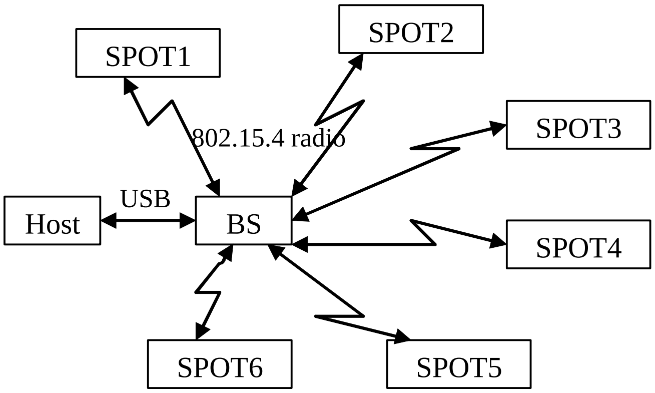


Figure 1. Star Topology

The binary tree topology is configured in two levels as depicted in Figure 2. This topology provides good breadth and some depth coverage. The simulation results show that the mean throughput of $\alpha_{on} = 1.2$ and $\alpha_{on} = 1.9$ are the lowest and their mean interarrival time are among the highest. Both parameters lie on the opposite edge of the self-similarity range ($H_{1.2} = 0.9\, and\, H_{1.9} = 0.55$, respectively), whereas, $\alpha_{on} = 1.4$ and $\alpha_{on} = 1.7$ ($H_{1.4} = 0.8\, and\, H_{1.7} = 0.65$, respectively) are in the mid-range of the self-similarity range. These two parameters produce higher mean throughput with lower mean interarrival times. From these results, it is assessed that the best-performing shaping parameter is 1.4. The mean throughput is 124 bytes/second, the mean interarrival time is 0.19 second (sec) and the mean delay of 4.76 sec. However, by interpolation a better performance parameter could be found between $\alpha_{On} = 1.5 - 1.6$. A point to note is that a three-level binary tree topology could not be established because data could not be captured and hence could not be evaluated.

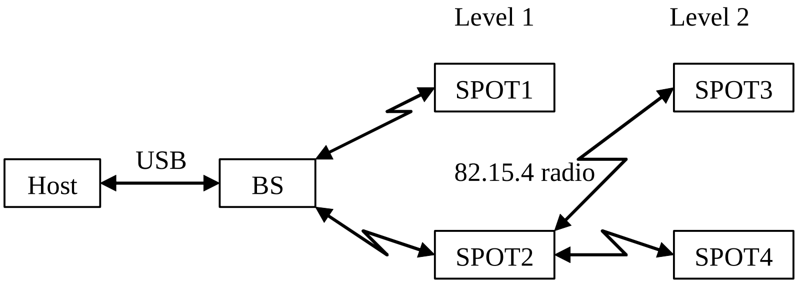


Figure 2. Binary Tree Topology

The chain topology is configured in a straight line (see Figure 3), and as such it is good in depth but poor in breadth of coverage. The simulation results reveal that the mean throughput performance is the inverse of the binary tree topology. The shaping parameters with higher self-similarity, i.e., $\alpha_{on} = 1.2, 1.4$, suffer a lower mean throughput than those at the lower self-similar range, i.e., $\alpha_{on} = 1.7, 1.9$. In this topology, $\alpha_{on} = 1.7$ has emerged to be the best performer with the highest mean throughput vale of 142 bytes/second, lowest mean interarrival time of 0.12 sec and mean delay of 3.33 sec. It is also observed that mean throughput reduces by more than half when there are four SPOTs in the chain topology.

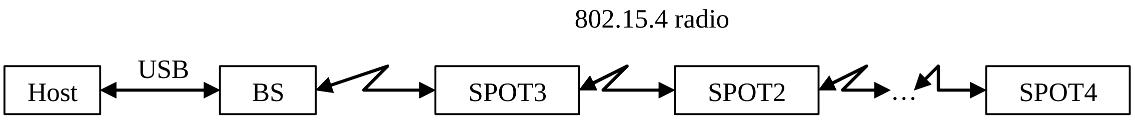


Figure 3. Chain Topology

In conclusion, Table 1 provides a recommendation of the topologies and their shaping parameter that will provide the desired breadth and depth network coverage.

| Topology | Recommended | | | Mean throughput (bytes/second) | Mean interarrival time (sec) | Mean delay (sec) |
|---|---|---|---|---|---|---|
| | $\alpha_{On}$ | Max breadth coverage | Max depth coverage | | | |
| Star | 1.4 | 2 | 1 | 917 | 0.049 | 0.26 |
| Binary Tree | 1.4 | 2 | 2 | 124 | 0.194 | 4.76 |
| Chain | 1.7 | 1 | 3 | 142 | 0.116 | 3.33 |

Table 1. Architecture Recommendation for IEEE 802.15.4 and the Zigbee Standard

The decision to select the type of topology is based on the trade-off between breadth and depth coverage. If it is desired to have depth coverage then the chain topology should be adopted. If the decision is to have good breadth and some depth coverage than a binary tree topology will be good. Otherwise, if only breadth coverage is desired, a star topology is excellent.

# ACKNOWLEDGMENTS

For my advisor Professor Weilian Su, I would like to thank him for the research advices, guidance, support and encouragement. His though provoking questions have made the thesis interesting and challenging. I have learned a lot from this thesis.

I would like to express my sincere appreciation to Professor John C. McEachen for his helpful inputs.

I would like to thank Miss Regine Oh, CPT Gil Nachmani and Mr. Koh Kim Leng who have taken time off from their own thesis work to help me in making this thesis a success.

To my beloved wife, Janet, who has been the most supportive person throughout my time at NPS. She has constantly encouraged me even when things are not going well. I would like to take this opportunity to give a big thanks to her.

Finally, I would like to thank God for giving me this opportunity to pursue a Master of Science in Electrical Engineering at NPS in Monterey, California. Throughout my stay in NPS, Psalm 23 has been my word of encouragement and guidance.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND

Information technology systems have provided the capability to collect, process, and distribute relevant data to thousands of locations, which has allowed the military to gain dominant battlespace awareness. In the age of Network Centric Warfare (NCW), the NCW architecture will most likely leverage small, low-power, and inexpensive network computing and wireless devices. Such wireless devices can be widely deployed to robustly and efficiently self-organize under dynamic conditions [1]. The infrastructures formed by the untethered wireless computing devices, such as wireless mesh sensor networks, can be adapted to serve a number of critical mission tasks.

Wireless mesh sensor networks consist of a large number of wireless sensor nodes that are strategically deployed in or near the area of interest. These networks, though strategically deployed, are formed in an ad hoc manner. The wireless sensor nodes are used to sense and capture physical and/or environmental information of the area of interest. The captured information is a result of real-time sampling of the physical and/or environmental information. The captured data is then disseminated through multi-hop broadcasting to the destination node. Due to the ad hoc nature of the deployment, there is normally no fixed base access point or switch exchange. Every node acts as a repeater to transmit data from nearby nodes to peers that are too far away to reach. As a result, the effective range of the wireless sensor network can be extended to a large span of distance, especially over rough or difficult terrain.

The advantages of wireless mesh sensor networks could potentially enhance the situational awareness of the fighting force greatly. This can be achieved if video or still images at the forefront of the battlefield could be transported to the command center via wireless mesh sensor networks. Essentially, the goal is to route the video or still images collected by intelligence sources such as unmannned arial vehicles (UAVs), special forces and scouts through the ad hoc wireless mesh sensor network deployed in the battlefield to the command center.

## B. MOTIVATION

The study will exploit the many advantages of wireless mesh sensor networks. There are a host of technical difficulties with transmitting video or still images over a sensor network. The file size of a video image creates a challenge for the sensor device. The sensor device is usually small and therefore does not have a large enough buffer to store and process video images. Furthermore, the received video images are to be transferred through multiple-hops to the destination. Thus, the study focuses on performing the necessary research to examine the performance of the wireless mesh sensor network when video images are being transferred over the network.

## C. THESIS GOALS AND METHODOLOGY

The goal of this thesis is to perform the necessary research to validate and implement a wireless mesh sensor network that will route video and still images from the intelligence source to the destination nodes (i.e., the command center). It has been observed by [2] that video applications exhibit self-similar characteristic. Modeling of self-similar traffic can be obtained by aggregating a large number of On-Off message sources. The Pareto distribution function, which fit well to this kind of traffic [3], will be used to model video traffic on the wireless mesh sensor network. Various performance behaviors of different network topologies will be investigated such as mean throughput, mean interarrival, mean packet drop and mean delay time.

The experiments for performance analysis will be carried out using the SUN Small Programmable Object Technology (SPOT) Java development kit. Sun SPOT is a sensor hardware developed by Sun Microsystems that has a small, flexible Java virtual machine (Project Squawk) on the sensor platform. Each Sun SPOT Java development kit consists of three sensor hardware SPOTs. These SPOTs are configured into three network topologies, namely star, binary tree and chain topology. The performance behaviors of each topology will be investigated.

## D. RELATED WORK

As mentioned that the goal is to perform the necessary research to validate and implement a wireless mesh sensor network that will route video and still images from the intelligence source to the destination nodes. This section presents related work pertaining to transmitting video and still images over wireless sensor network.

Culuriello, Joon and Savvides [4] developed a zero-computation inexpensive methodology for compressing frame-difference video. Video is encoded using address-event representation (AER) and uses custom image sensors capable of detecting intensity-difference (motion) information. This compression enables wireless sensor nodes to stream temporal frame-difference video over wireless networks at higher rates. The work focuses on keeping a constant low bit rate over sensor network channels based on Zigbee radios, so that the motion information in an image can degrade gracefully when the resolution or image content varies over time. This thesis investigates the network topologies that will support as well as extend the distance of transmitting video images over wireless sensor network.

## E. THESIS ORGANIZATION

This chapter provided the introduction, thesis goals, methodology and motivation for studying multimedia ad hoc wireless mesh sensor networking. Chapter II provides an overview of wireless sensor networking and an introduction to Sun SPOT sensor hardware. The chapter will also provide an introduction to mesh networks and the Ad hoc On Demand Distance Vector (AODV) routing algorithm. Chapter III is dedicated to the experiment setup. This chapter will present the simulation of the Pareto On-Off distribution model and the sensor networking architecture adapted in the study. Chapter IV will present the performance analysis, the approximation method for simulating the Pareto On-Off distribution model and discussion of the simulation results. Chapter V will conclude the study and provide recommendations for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. WIRELESS SENSOR NETWORKING

This chapter introduces the characteristics and the components of a wireless sensor network (WSN) and highlights the performance evaluation metrics for a wireless sensor network. This chapter will also discuss various routing protocols, network topologies and the hardware used in this thesis. An overview of the capability of a wireless sensor network will also be discussed along with its various applications.

## A. INTRODUCTION TO SENSOR NETWORKS

A WSN is a network of numerous inexpensive devices (or sensor nodes) with sensing, computation and wireless communication capabilities. The sensor nodes are small, lightweight and are identical. They can be deployed randomly or deterministically in a large quantity, of up to hundreds or thousands, in an area of interest. These sensor nodes are usually placed in close proximity, approximately 10 meters apart (see Figure 4). They are capable of communicating with each other for self-organizing, broadcasting and multi-hop routing, or communicating directly with the base station (BS). Due to advancements in technology, the manufacturing cost of sensor nodes has reduced dramatically such that large quantities can be purchased at low cost and be networked to operate cooperatively unattended for a variety of applications. These applications can range from inventory control, temperature control, acceleration measurement, signal strength, light level measurement, intrusion detection, target field imaging and tactical surveillance. This section introduces the general components and the operating system of a sensor node. The performance of a WSN and its topology will also be discussed.

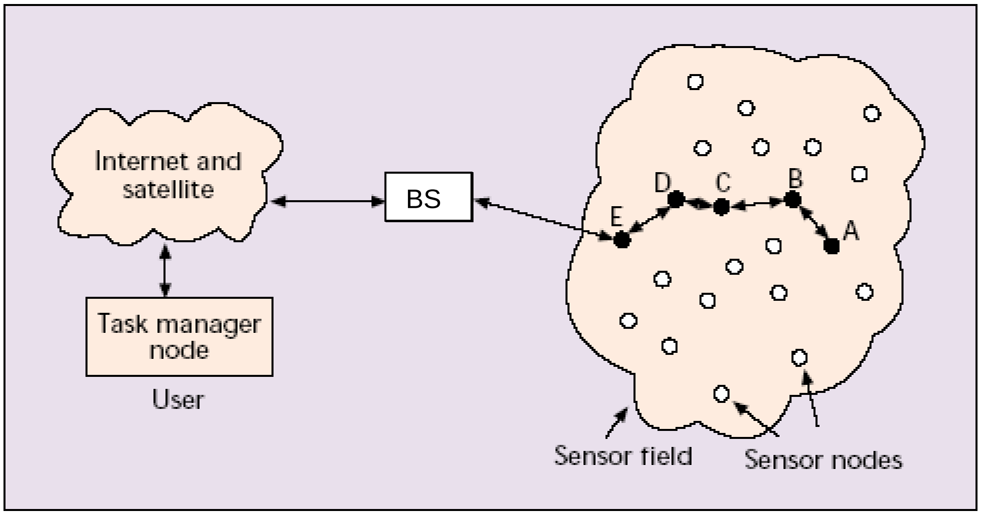


Figure 4. Sensor Nodes Scattered in a Sensor Field. (After [5])

### 1. Sensor Node Hardware Components

A sensor node consists of the following five main components [6]: the processor, memory, sensor, communication device, and power supply. Figure 5 shows the schematic diagram of the components of a sensor node.

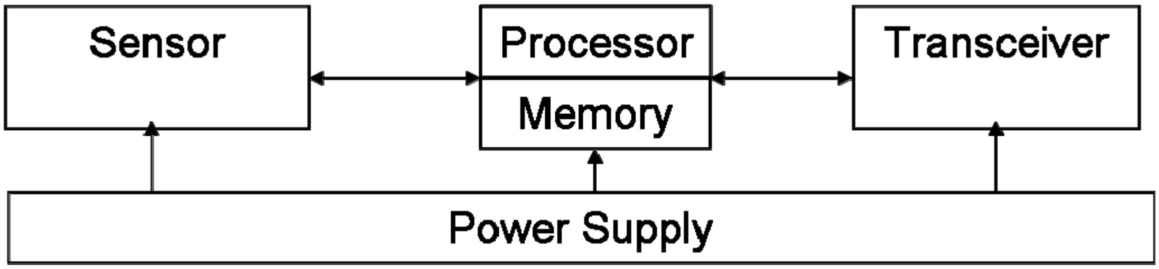


Figure 5. Components of a Sensor Node. (After [5])

#### *a. Processor*

The processor is the brain of the sensor node. It performs tasks, processes data and controls the functionality of other components in the sensor node. It processes the data that were collected from its own sensor and decides when and what action is needed to be carried out. It receives data from other sensor nodes and decides on the necessary action. The processor also executes various programs ranging from signal processing to communications protocols and application programs. In essence, the processor manages procedures that make the sensor node collaborate with other sensor nodes to carry out the assigned sensing task.

#### *b. Memory*

The memory component is made up of random access memory (RAM) and a flash memory. The RAM stores sensor readings and packets sent by other sensor nodes. However it loses its contents once the power supply is interrupted. Flash memory retains its contents even when the power supply is cutoff and can also be used like RAM if there is insufficient RAM. However, flash memory takes a longer time to read and/or write and it also consumes more energy.

#### *c. Transceiver*

The transceiver is used to transmit and receive data between sensor nodes, (i.e. connecting the sensor node to the network). It uses radio frequency (RF) communication that requires modulation, filtering and multiplexing circuitry. It may seem that adding this circuitry increases the complexity and manufacturing cost; however, it is the most preferred communication method as RF does not require a line of sight (LOS) between the transmitter and receiver. Data conveyed in the sensor network is separated into packets that are small in size. Also, frequency reuse is high due to the close proximity between the sensors and their small broadcast range. The sensor node transceiver is sometimes IEEE 802.15.4 compliant and consequently operates in the 2.4GHz to 2.4835 GHz ISM unlicensed bands [7].

#### *d. Sensors*

The sensors used in a WSN are categorized into three categories as follows: passive omni-directional sensors, passive narrow beam sensors and active sensors. A passive omni-directional sensor measures the quantity at the point of the sensor nodes without manipulating the environment by active probing. Examples of such sensors are temperature and light sensors. A passive narrow beam sensor has a well defined direction of measurement. An example of such sensors is a camera. Finally, an active sensor actively probes the environment. Examples of this kind of sensor are laser or sonar systems.

#### *e. Power Supply*

Sensor nodes are usually deployed in areas that are physically difficult to access. Hence, the lifetime of the sensor node is dependent on its own power sources. Intuitively, the node’s power supply is a crucial component. Typically, the energy source of a sensor node is often traditional batteries, such as AA size batteries or rechargeable batteries which have a fixed lifespan. Since sensor nodes are usually deployed in areas that are inaccessible, it is necessary to extend the lifespan of a node by recharging the battery from the environment. This type of lifespan extension is called energy scavenging [8], and an example of it is a technique such as using solar cells to collect solar energy.

### 2. Performance Metrics for Wireless Sensor Networks

The following is a list of metrics recommended by [9] as a measure of the effectiveness of a sensor network.

#### *a. Latency*

It is necessary that the information captured by a sensor node is sent back to the designated receiver as soon as the data is available. Hence, the sooner the information is made known to the designed receiver the better.

#### *b. Accuracy*

Data accuracy is important as it allows less sensor nodes to be deployed in an area of interest. That is, when deploying sensor nodes, more areas of interest can be surveyed with the same amount of sensor nodes, which results in cost savings.

#### *c. Fault Tolerance*

Sensor nodes can be deployed in areas with harsh environmental conditions. Hence, they are usually very prone to failure as a result of a lack of power, physical damage, environmental interference, and man-made interference such as electronic jamming. Fault tolerance is the ability to sustain sensor network functionalities without interruption when sensor nodes exhibit failure [5]. Therefore, it is essential that networks are resilient to node failure. In a scenario where there are several nodes that fail, the routing protocol must be able to establish alternative routes to route data to the designated destination. The effect of node failure can be reduced by introducing node redundancy or collaborative processing and communication, or by applying both.

#### *d. Scalability*

The number of sensor nodes deployed in an area of interest may range in the order of hundreds to thousands in which the distance between nodes can be 10 meters or less in diameter. Thus it is essential that a routing algorithm can be scaled for large number of nodes.

#### *e. Energy Efficiency*

As mentioned earlier, a sensor node is usually operated on batteries with a limited lifespan. In a multi-hop ad hoc sensor network, the sensor node plays the role of both data originator and data router. The malfunctioning of few nodes can cause significant changes to the topology and require rerouting of packets and reorganization of the network [3]. Thus, it is essential that the applications running on the nodes are energy efficient. Efficient use of energy resources can prolong the lifetime of the network.

### B. ROUTING PROTOCOL

This section introduces some of the existing routing protocols that are used in a wireless sensor network [10]. As depicted in Figure 6, these routing protocols may be categorized into table-driven or source initialed (demand-driven) [10]. The solid line in the figure represents a direct descendent while the dotted line represents a logical descendent.

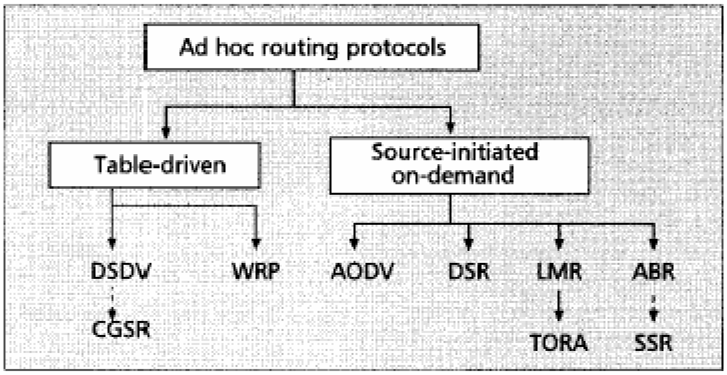


Figure 6. Categorization of Ad Hoc Routing Protocols. (From [10])

### 1. Table-driven Routing Protocols

Table-driven routing protocols strive to ensure that the routing information is the most up-to-date from each node to other nodes within the network. The nodes exchange routing information to obtain the latest snapshot of the topological information. This results in less delay for the information to reach a specific destination. However, there is a price to pay for a shorter delay. Control messages have to be sent even though the nodes may not be transmitting to each other.

#### *a. Destination-sequenced Distance Vector (DSDV)*

Destination-Sequenced Distance Vector (DSDV) is described as a table-driven protocol in [11]. It is a table-driven algorithm based on the Bellman-Ford routing mechanism. It is an improvement to the Bellmen-Ford algorithm as it ensures that the routing table is free from loops. This is accomplished by assigning a unique sequence number to each route. Since the sequence number is unique to each route, it distinguishes new routes from the old routes, and thereby prevents the existence of routing loops.

Every node in the network maintains a routing table, which records the destination nodes that it is connected to. The routing table also indicates the number of hops needed to arrive at the destination nodes. Thus the routing table consists of a destination, route, the hop count, and a sequence number. The updated routing table is transmitted periodically throughout the network in order to maintain table consistency. This can be done via two types of packets, namely a full dump and incremental packet. In the full dump packet, the entire routing table of a given node is transmitted to its directly connected neighbors. These packets are transmitted infrequently. In incremental packets, only routes that have changed as compared to the last full dump are sent to the destination nodes. Each node also maintains an additional table which stores the data that were sent in the incremental packets.

The broadcasted new routes contain the address of the destination nodes, the hop counts needed to arrive at the destination nodes, the old sequence number and the new sequence number. The route with the most recent sequence number is always updated onto the routing table. In the event that two routes have the same sequence number, the route with the least number of hops is always adopted. This is to ensure that the shortest path is always adopted.

Nodes also keep track of the settling time of routes. Settling time is the weighted-average time that routes to the destination will fluctuate before route with the least number of hops is received. Nodes will delay the broadcast of routing information based on the settling time.  This avoids sending extra updates, which would increase the traffic in the network before a better route arrives in the near future.

#### *b. Cluster-head Gateway Switch Routing (CGSR)*

The CGSR uses a hierarchical cluster-head-to-gateway routing approach to route traffic [12] instead of a flat network which was used by DSDV. The protocol selects a cluster head within a group of nodes. Packets designed for a destination node are routed to their cluster head and then the packet is routed from the cluster head to the gateway to another cluster head and so on until the cluster head of the destination node is reached. The packet is then transmitted to the destination node. Figure 7 depicts the routing protocol.

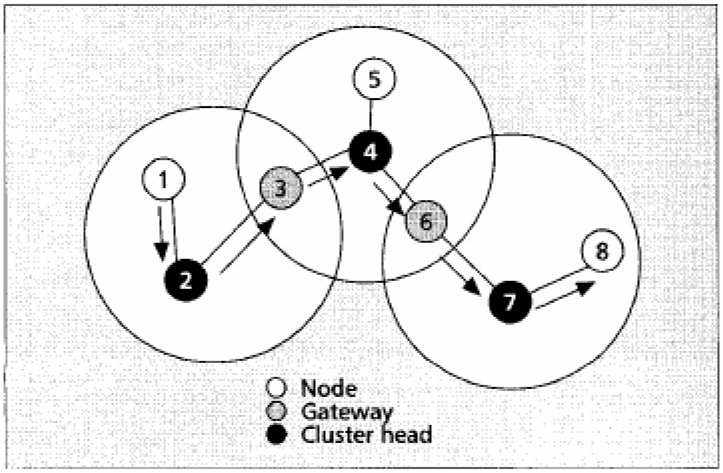


Figure 7. CGSR: Routing from Node 1 to Node 8. (From [6])

Similar to DSDV, in CGSR each node maintains a "cluster member table" which lists the destination cluster heads of each node. The "cluster member table" is broadcast using the DSDV algorithm and the receiving nodes update their own table upon receiving it from their neighbors.

In addition, each node has to maintain a routing table which is used to determine the next hop in order to reach the destination node. That is, when a node receives a packet, it will have to consult the cluster member table and the routing table before the packet is routed to the destination node. Thus, the selected path to reach its destination node may not necessarily be the most optimal route.

The disadvantage of this scheme is that frequent changes of a cluster head can badly affect the performance of the routing protocol. This is because more time is spent reselecting the cluster head instead of relaying messages. To avoid this problem, the least cluster change (LCC) clustering algorithm is used. Using this algorithm, a cluster head will be reselected only when two cluster heads come in contact with each other or when one of them moves out of range of any existing cluster head.

#### *c. Wire Routing Protocol (WRP)*

WRP is a table-based protocol [13]. It attempts to maintain routing information of all nodes within the network. Each node maintains four tables, namely a distance table, routing table, link-cost table, and message retransmission list (MRL) table. The MRL contains the sequence number, a retransmission counter, acknowledgement-request flag vector for each neighbor, and a list of updates sent in the update message. It also contains a record of which update in an update message needs to be retransmitted and which neighbors should acknowledge the retransmission.

In WRP, each node uses update messages to update neighboring nodes of any link changes. An update message contains the destination, the distance to the destination, the predecessor of the destination, and a list of responses indicating which nodes should acknowledge the update. Nodes will send update messages after processing updates from neighbors or detecting a link change. If there is a lost link between two neighbors, update messages are sent to their neighbors. The neighbors will update their distance table and search for new paths through other nodes. The original node will be notified if new paths exist. Connectivity between nodes is maintained by sending a hello message when there is no update message to be sent. When a new node joins the network, it sends a hello message to each node in the network. Upon receiving the hello message, the node will update its own routing table and forward a copy of its routing table information to the new node.

WRP prevents routing loops and avoids the “count-to-infinity” problem [14] by forcing each node to perform consistency checks of predecessor information reported by the neighboring nodes.

### 2. On-demand Routing Protocols

The on-demand approach is different from the table-driven approach in that, the initiating nodes initiate a route discovery process within the network only when there is a need to transmit data to a destination. The process is said to be completed only when a route is found or when all possible route permutations have been examined. Once a route is established, it will be maintained by the maintenance procedure. It will only be disconnected when the destination node becomes inaccessible along every path from the source or when the route is no longer needed.

#### *a. Ad Hoc On-demand Distance Vector (AODV)*

The AODV routing protocol is an improvement to the previously mentioned DSDV algorithm. In AODV, a node is not required to consistently maintain the routing information, as the path discovery process begins only when there is data to be sent to the destination node. Opposite of DSDV, during the entire routing process only nodes that are along the direct path, between the initiating and the destination nodes, have to maintain the routing information or participate in the routing table exchange [15]. Thus AODV seeks to minimize the number of required broadcasts between the nodes.

To illustrate the process, if N1 needs a route to N8, it broadcasts a route request (RREQ) message to its neighbors which then forward the request to their neighbors, and so on, until either N8 is reached or an intermediate node with a route to N8 is discovered. Figure 8 depicts the routing process.

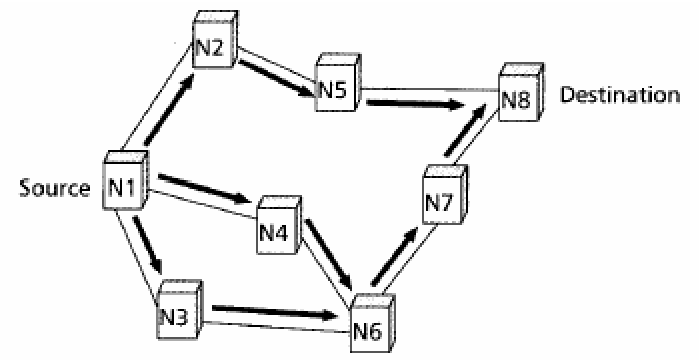


Figure 8. Propagation of RREQ to N8. (From [10])

Intermediate nodes can reply to the RREQ only when they have a route to N8. If a route to N8 exists, the node will update its route table with the address of the neighbor who sent the broadcast packet, thus establishing a reverse path. In the event that it received the same RREQ again, the second copy will be discarded. Once the RREQ arrives at N8 or an intermediate node that has a route to N8, the N8/intermediate node will respond by unicasting a route reply (RREP) packet back to the sender node. The RREP is subsequently route back to N1 via the same route that RREQ is routed to N8/intermediate node. The nodes along this path then set up the forward path in their route table pointing to the chain of nodes that unicast the RREP, which eventually points to either N8 or the intermediate node with a route to N8. Figure 9 depicts the path of the RREP to N1.

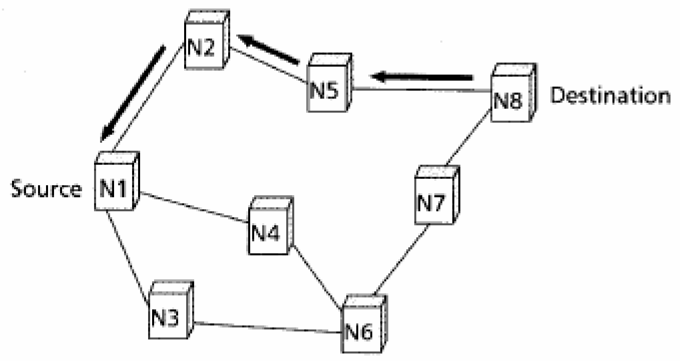


Figure 9. Path of RREP to N1. (From [10])

Each node maintains the following four pieces of essential information: its own sequence number, a broadcast ID which is incremented for every RREQ that is initiated, its own IP address, and the destination node's sequence number. An RREQ is uniquely identified by the broadcast ID and the node IP address. Each node uses the destination sequence number to ensure all routes are loop-free and contain the most recent route information.

The route maintenance is carried out as follows. If N1 is repositioned, it will reinitiate a route discovery protocol to find a new route to N8. If a node along the path is repositioned, its upstream neighbor (if it notices the changes) will initiate a link failure notification message to inform each of its upstream neighbors. The link failure notification will eventually arrive at N1, which may reinitiate route discovery to N8 if the route is still needed. In addition, each node broadcasts hello messages to inform each node of the existence of other nodes in its neighborhood; thus, maintaining the local connectivity of the node.

### *b. Dynamic Source Routing (DSR)*

DSR is an on-demand routing protocol that is based on the concept of source routing [16]. In this protocol, each node is required to maintain route caches which contain the source routes that are known to it. The route caches are continuously updated by the node as it continues to discover new routes.

The DSR protocol consists of two major phases of operation, namely route discovery and route maintenance. When there is a packet to send to a destination node, the node will consult its own route cache for an existing unexpired route to the destination. If a route exists it will use the existing route to send the packet. However, if a route does not exist, it will send a RREQ packet to every node in the network. This is where the route discovery phase begins. The RREQ packet consists of the destination address, source address, and a unique identification number.

Each node that receives the RREQ packet checks its own route table for an existing route to the destination. If a route does not exist, it will add its address to the route record and then forward the packet along with its outgoing link. The process will go on until the destination node or an intermediate node that has a route to the destination is reached.

To prevent a large number of RREQ packets from being propagated into the network, the protocol limits the number of route requests that a node can send. A node will only forward the route request if it has not received the RREQ before and there is no record of it in the route record. Once the destination node or the intermediate node with a route to the destination node receives the RREQ, it will generate a route reply [17]. A packet arriving at either of these nodes will have a route record that contains the sequence of hops it has taken. Figure 10 shows the route record as the packet propagates through the network. For example, a packet sent from N1 to N8 will have a route record of N1-N2-N5 when it arrived at N8. The destination node will place this route record contained in the RREQ onto the route reply. An intermediate node will append its cached route to the route record and then generate the route reply. If the replying node has a route (in its route cache) to the initiator, the route reply will then be sent back to the initiator. Otherwise, the node may do one of the following. If the protocol supports symmetric links, the node may reverse the route in the route record and send the route reply back to the initiator. If symmetric links are not supported, the node will have to initiate its own route discovery to the initiator and the route reply will be piggybacked on the RREQ. Figure 11 illustrates the transmission of the route reply with its associated route record back to the initiator. For example, route reply sent from N8 to N1 will have route reply of N8-N5-N2-N1.

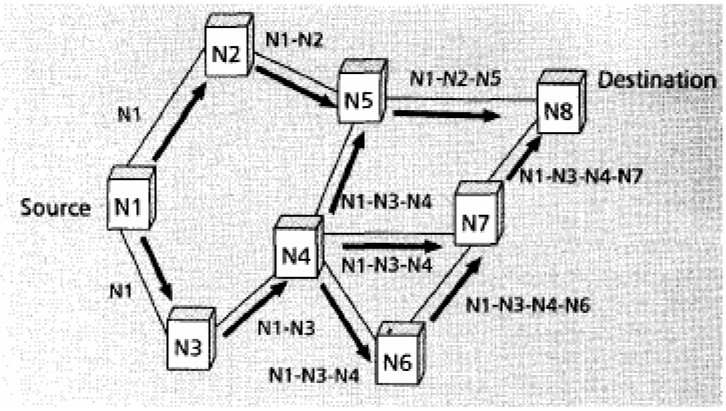


Figure 10. DSR Route Discovery. (From [10])

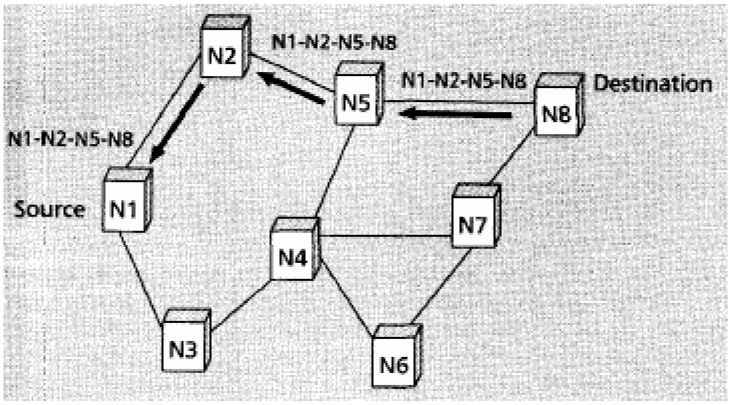


Figure 11. Propagation of Route Reply with Route Record. (From [10])

Each node performs route maintenance through the use of route error packets and acknowledgements. When a fatal transmission problem is encountered at the data link layer of a node, route error packets will be generated and sent to the neighboring nodes. On receiving the route error packet, the hop error is removed from the node’s route cache and all routes containing the hop are truncated at that instant. As for acknowledgements, it is used to verify that the route links are correctly operated.

### *c. Temporally Ordered Routing Algorithm (TORA)*

The TORA is a highly adaptive loop-free distributed routing algorithm developed based on the concept of link reversal [18]. It is designed to operate in a networking environment that is highly dynamic. It is an on-demand routing protocol that provides multiple routes for any desired source/destination pair. The key design concept of TORA is that it confines the sending of control messages to nodes where the topological change occurs [10]. To achieve this, every node has to maintain routing information about adjacent nodes, i.e., nodes that are one hop away.

The protocol performs basic functions such as route creation, route maintenance, and route erasure. In the first two functions, the node establishes a directed acyclic graph (DAG) (that is rooted to the destination) based on a height metric. The height metric consists of the following five elements: logical time of a link failure, unique ID of the node that defined the new reference level, a reflection indicator bit, a propagation ordering parameter, and the unique ID of the node. Subsequently, each link is assigned a direction (i.e., upstream or downstream) based on the relative height metric of neighboring nodes. Figure 12 depicts the use of a height metric to establish a DAG. The accuracy of the height metric is dependent upon the logical time of a link failure. Thus, clock synchronization between nodes is an important factor for TORA. Usually clock synchronization is achieved through an external clock source such as GPS.

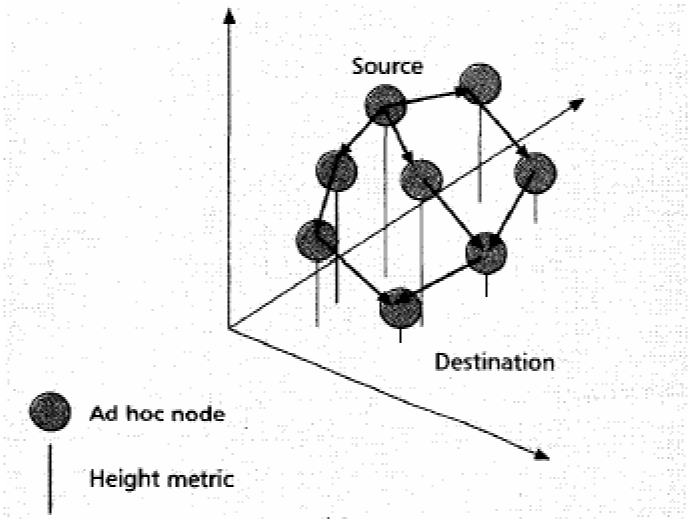


Figure 12. Route Creation, Using a “Height” Metric to Establish DAG. (From [10])

When a node is displaced from its original location, the DAG route is broken and hence route maintenance is carried out to re-establish a DAG that is rooted to the same destination. The displaced node generates a new reference level which results in the propagation of that reference level by its neighboring nodes. This effectively coordinates a structured reaction to reestablish the DAG. In adapting to the new reference level, links are reversed to reflect the changes. Effectively, this is equivalent to reversing the direction of one or more links when a node has no downstream links. Figure 13 shows the route maintenance process in TORA. When the link from D to F fails, the link from D to B and D to C is reversed (see Figure 13 (2)). This subsequently causes the link from B to C and A to B to reverse as well (see Figure 13 (3) and Figure 13 (4)).

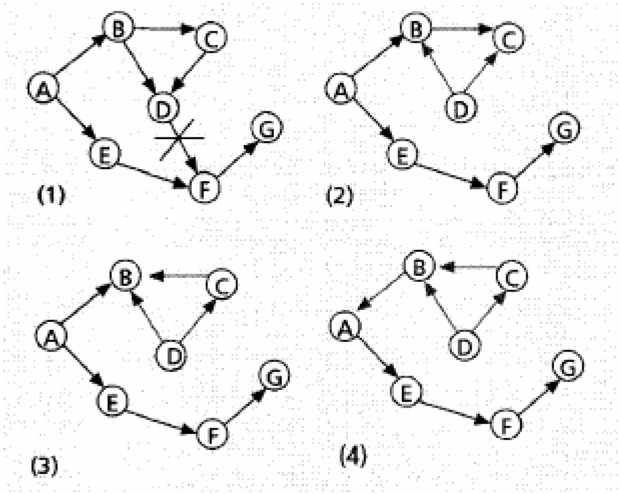


Figure 13. Route Maintenance Process. (From [10])

Finally, the route erasure function broadcasts clear packets (CLR) throughout the network to erase invalid routes.

### *d. Associativity-Based Routing (ABR)*

ABR defines a new routing metric, known as the degree of association stability, for ad hoc mobile networks. The protocol is free from loops, deadlock, and packet duplicates. The fundamental objective of ABR is to derive longer-lived routes for ad hoc mobile networks.

In ABR, route selection is based on the association stability of nodes. Association stability is defined by connection stability of one node with respect to another node over time and space [10]. A high degree of association stability implies low state of node mobility while a low degree of association stability implies a high state of node mobility.

Each node periodically generates a beacon to inform its neighboring nodes of its existence. The beacon updates the associativity tables of the receiving node by incrementing its associativity tick with respect to the sending node. The associativity tick is reset when the neighbors of a node or the node itself moves out of proximity.

ABR operates in the following three phases: route discovery, route reconstruction (RRC), and route deletion. During the route discovery phase, an initiator node will broadcast a broadcast query (BQ) message to search for nodes that have a route to the destination. All intermediate nodes that receive the BQ message will append their addresses and their associativity ticks with their neighbors along with quality of service (QoS) information to the query packet. If a node receives the same BQ message twice, the message will be discarded. When the packet arrives at the destination node, it examines the associativity ticks for multiple possible paths to select the optimal route. If multiple paths have the same degree of stability, the route with the minimum number of hops will be selected. The destination then sends a reply packet back to the source via the same path. Nodes along the path will propagate the reply packet and mark their routes as valid.

The RRC is initiated to reconstruct the routing table when there is a change in the network topology. This phase consists of partial route discovery, invalid route erasure, valid route updates, and new route discovery. RRC can be initiated by the source node, destination node or intermediate node. When a route is no longer needed, the source node initiates the route delete phase, which broadcasts a route delete broadcast to all nodes (regardless of whether they are along the route or not). This is to inform all nodes of the route deletion so that they will update their routing tables.

#### *e. Signal Stability Routing (SSR)*

The SSR is a signal stability-based adaptive routing protocol. Unlike the aforementioned algorithms, it selects routes with stronger connectivity between nodes and a node’s location stability [10].

SSR is further divided into two cooperative protocols, namely the Dynamic Routing Protocol (DRP) and the Static Routing Protocol (SRP). The DRP is responsible for maintaining the signal stability table (SST) and routing table (RT). The SST keeps track of the signal strength of neighboring nodes. The signal strength of neighboring nodes is obtained by periodic beacons from their link layer.

After updating all appropriate table entries, the DRP hands over the received packet to the SRP. When the SRP receives a packet and if it is the intended receiver, it will pass the packet to the stack. Otherwise, it will forward the packet to the destination based on the route to the destination found in its RT. If the RT does not contain a route to the destination, it will initiate a route search process to find a route to the destination.

Route requests are propagated throughout the network. However, only nodes with the strongest connectivity with the sender and that have not processed the route request message before will receive the route request message. At the destination node, only the first arriving route-search packet is replied to because it is assumed to be the shortest and/or less congested path [10].

The DRP then reverses the selected route and sends a route-reply message back to the initiator. During this time, the DRP of the nodes along the return path will also update their RTs accordingly. If there is no route-reply message received at the initiator within a stipulated timeout period, the initiator will change the PREF field in the header to indicate that weak channels are also acceptable.

When a link failure within the network is detected, the intermediate nodes send an error message to the source indicating which channel is the failed channel. The source then initiates another route-search process to locate a new path to the destination. It will also send an erase message to notify all nodes of the broken link.

## C. NETWORK TOPOLOGY

This section provides an introduction to network topologies deployed in WSNs. The term topology refers to the way in which the end point, or stations, attached to the network are interconnected [19]. There are three common topologies deployed for WSNs, namely: star, mesh and tree. Each topology has its own set of advantages and disadvantages and supports a different power profile. Thus, the power consumption of each topology has to be assessed before a suitable topology can be selected to be deployed into the desired WSN.

### 1. Star Topology

In a star topology, all nodes are one hop away from the central node. Figure 14 shows a star topology. This topology consumes the least amount of power as the nodes do not need to synchronize or route other nodes' packets. However, the transmission distance of the radio, between node and central node, is the limiting factor to the size of the coverage area of the sensor network. The quality of the radio link defines the transmission distance.

A star topology is most effective when the area of coverage of the network is small and requires extended period of operation. In terms of power consumption, if operating on two AA size batteries, it is said to be able to operate up to 5 years [20].

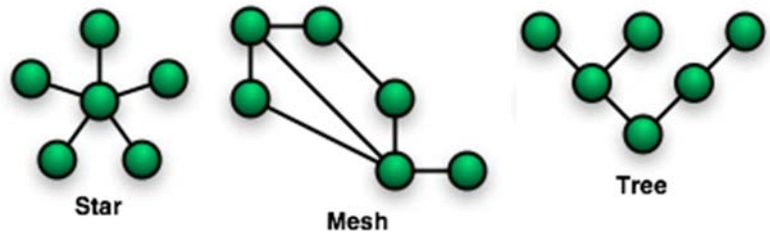


Figure 14. Star, Mesh and Tree Topology. (After [21])

### 2. Mesh Topology

The mesh topology as shown in Figure 14 represents a highly distributed network that allows transmission only to nodes that are one hop away [22]. Packets are multi-hopped through numerous nodes before arriving at the destination. Thus, it is good for large-scale networks of wireless sensors that are distributed over a wide geographical area. In the event that a node failure occurs in the network, the packet will still be routed to the destination via other links making the network highly tolerant to failure. The topology is suitable if the network is required to operate over a long period of time. The battery operational life of the node is expected to last between one to three years [20].

### 3. Tree Topology

In a tree topology, as shown in Figure 14, a central root node is connected to one or more other nodes that are located at the second level. The second level nodes will also have one or more nodes that are located at the third level connected to it. The tree topology is symmetrical in nature; each node in the network has a specific fixed number of nodes connected to it at the next level. The largest drawback of the topology is that when one of the nodes at the second level fails the entire chain of nodes connected to that node will be rendered inoperative.

## D. SUN SPOT SENSOR

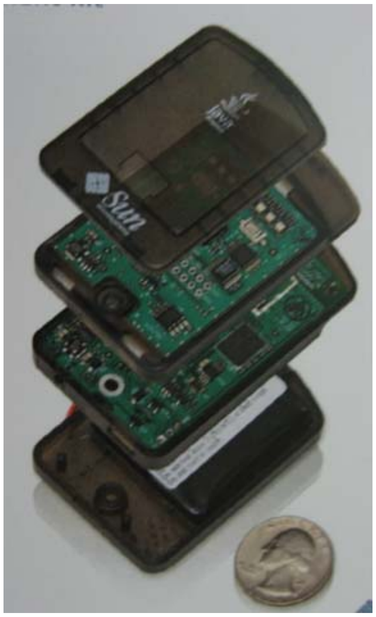
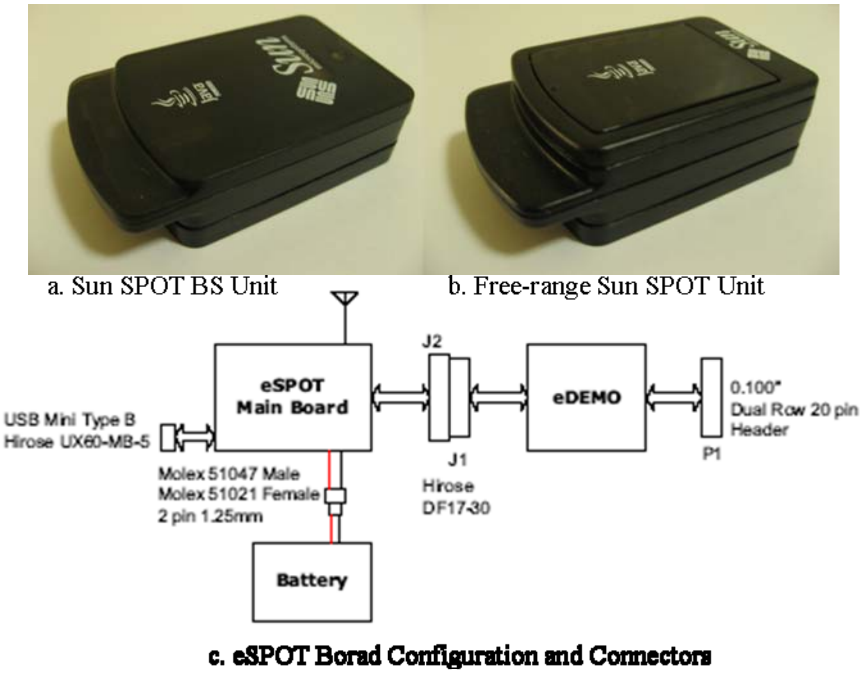


Figure 15. Photo of a Sun SPOT. (From [7])

In this section the actual hardware, namely the Sun Small Programmable Object Technology (SPOT) development kit, used in this study will be presented. The Sun SPOT development kit consists of the following three sensors: one base station (BS) Sun SPOT unit and two free-range Sun SPOT units. Figure 15 shows a photo of a Sun SPOT.



J2
eSPOT Main Board
eDEMO
0.100" Dual Row 20 pin Header
USB Mini Type B Hirose UX60-MB-5
J1
P1
Molex 51047 Male Molex 51021 Female 2 pin 1.25mm
Hirose DF17-30
Battery

a. Sun SPOT BS Unit

b. Free-range Sun SPOT Unit

c. eSPOT Borad Configuration and Connectors

Figure 16. a. Sun SPOT BS Unit, b. Free-range Sun SPOT Unit and c. eSPOT Board Configuration and Connectors. (From [7])

Figure 16 shows the photos of the BS, the free-range Sun SPOT unit, and the components that make up the two units. The BS has an eSPOT main board without a battery or an application board. Power is supplied by a USB connection to a host workstation. The base station serves as a radio gateway between other Sun SPOTs (and theoretically other 802.15.4 devices) and the host workstation. The eSPOT unit contains the main board with a rechargeable Li-Ion prismatic battery and an example of an eSPOT daughterboard, the eDEMO board.

The free-range Sun SPOT unit has an eSPOT main board, an eDEMO board and is powered by a battery.

### 1. eSPOT Main Board

The eSPOT main board contains the main processor, memory, and 802.15.4 radio transceiver and antenna. The main processor is an Atmel AT91RM9200 system-on-chip (SOC) integrated circuit. The memory is a single Spansion S71PL032J40, a multichip package (MCP) that consists of a 4 MByte NOR Flash memory and a 512 Kbyte pseudostatic random access memory (pSRAM). The pSRAM is self-refreshing and automatically enters power down mode when not selected for access. Memory contents are maintained as long as the power supply USB, external power supply or battery is connected (even during periods of off or “deep-sleep”). The wireless network communications use an integrated radio transceiver, namely the TI CC2420. The transceiver is IEEE 802.15.4 compliant and operates in the 2.4 GHz to 2.4835 GHz ISM unlicensed bands. The IC contains a 2.4 GHz RF transmitter/receiver with digital direct sequence spread spectrum (DSSS) baseband modem with MAC support. The effective bit rate is 250 kbps and chip rate is 2000 kchips/second.

#### 2. eSPOT Daughterboard (eDEMO)

The supplied eDEMO board is an example from the possible universe of eSPOT daughterboards. It contains an Atmega88 processor, flash memory, a light sensor, an accelerometer, and temperature sensor. The Atmega88 microcontroller communicates with the main board as a slave device. The flash memory is a series of EEPROM memory for storing configuration information.

#### 3. Battery

The internal battery is a 3.7V 720maH rechargeable lithium-ion prismatic cell. The battery has an internal protection circuit to guard against over-discharge, under voltage and overcharge conditions. The battery can be charged from either the USB type mini-B device connector or from an external source with a 5V $\pm 10\%$ supply. Typical shelf life losses at room temperature are about 2% of the battery’s capacity per month and the rate can increase with the rise in temperature.

### E. SUMMARY

This chapter introduces the characteristics and the components of a WSN and highlights the performance evaluation metrics for a WSN. Some of the routing protocols and network topologies were discussed to better appreciate the potential of the technology. This chapter also provided an introduction to Sun SPOT, which is a sensor used in this thesis. The next chapter will provide an overview of the experimental setup.

# III. EXPERIMENTAL SETUP

This chapter provides an overview of the experiment setup. It will present the traffic modeling methodology, the network topologies adapted for the experiment and an outline to the Sun SPOT application. A detailed description on the experimental procedure will be presented at the end of this chapter.

## A. GENERAL

The equipment involved in the experiment is a development PC (a Dell laptop) installed with Java programming software IDE NetBeans version 5 and Sun SPOT development kits. Each Sun SPOT development kit consists of a base station (BS) and two free-range Sun SPOTs (SPOT). Six Sun SPOT development kits are available for this study.

The BS and the SPOTs are Java platform; hence, Java bytecodes are written to program the BS as a receiver and the SPOTs as transmitters. The structure of an On-Off process for traffic modeling of video traffic is implemented in Java bytecode for downloading onto the SPOTs. The On-Off process will be described in detail in the next section. In the experiment, the WSN is deployed in three network topologies and the performance parameters such as mean throughput, mean interarrival time, mean packet drop and mean delay are computed from the collected data for analysis. The results and analysis will be discussed in the next chapter.

## B. TRAFFIC MODELING

To examine the possibility of routing video traffic in a WSN, it is important to model the traffic flow in the network. An appropriately selected traffic model will reveal the possible performance of the network, in terms of mean throughput, mean packet interarrival time, mean packet drop and mean delay, when subject to the video application characteristics.

In modeling traffic flow, there are options of using Poisson and self-similar message injection distributions. Traffic flow in MPEG-2 video applications has been observed to be self-similar [2]. Self-similar traffic can be modeled by aggregating a large number of On-Off message sources [3]. The length of time each packet spends in either the On or the Off state should be selected according to a distribution which exhibits long-range dependence [3]. The general structure of an On-Off process is comprised of alternate periods of traffic generation activity (On state) and inactivity (Off state), as depicted in Figure 17.

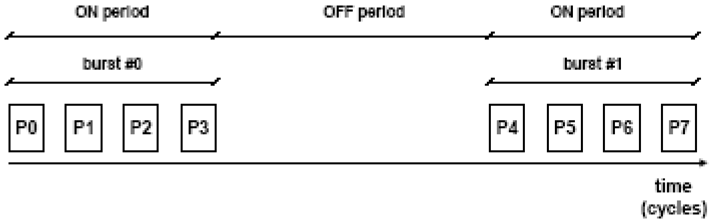


Figure 17. On-Off Traffic Model and its Parameters. (From [23])

During the On period, the traffic sources produce a fixed-length packet at a regular interval, while in the Off period there is no packet generation. The size of the On and Off periods varies. According to [3], a Pareto distribution satisfies all the requirements. The Pareto distribution is a heavy-tailed distribution with a probability density function given by Equation (3.1). The corresponding cumulative distribution function is given by Equation (3.2). The parameter *k* specifies the minimum value that the random variable can take. In this experiment, *k*=1. The shaping parameter $\alpha$ determines the mean and variance of the random variable.

$$f(x) = \frac{\alpha}{k}\left(\frac{k}{x}\right)^{\alpha+1}, where\, \alpha > 0, k > 0. \tag{3.1}$$

$$F(x) = P[X \le x] = 1 - \left(\frac{k}{x}\right)^{\alpha} \tag{3.2}$$

$$t_{On} = (1-r)^{\frac{-1}{\alpha_{On}}}, t_{Off} = (1-r)^{\frac{-1}{\alpha_{Off}}} \tag{3.3}$$

Equation (3.3) describes Pareto On-Off distributions used to model traffic. The parameter *r* is a random number uniformly distributed between 0 and 1. It is randomly chosen so that the size of the On and Off periods can be dynamically generated. The shaping parameter $\alpha$ is a constant, which shapes the On-Off period. It is a user-defined parameter to adapt the traffic generation to the application characteristics. For self-similar traffic, $\alpha$ falls in the range of $1 < \alpha < 2$ [24], which is related to $0.5 < H < 1$. $H$ is the Hurst parameter given in Equation (3.4) and it describes the degree of self-similarity. In modeling self-similar traffic for applications such as MPEG-2 video and Ethernet traffic, [3] recommend using $\alpha_{On} = 1.9$ and $\alpha_{Off} = 1.25$ and [24] recommended using $\alpha_{On} = 1.2$ and $\alpha_{Off} = 1.2$, respectively.

$$H = \frac{3-\alpha}{2} \tag{3.4}$$

In this study, $\alpha_{On} = 1.9$, $\alpha_{Off} = 1.25$ and $\alpha_{On} = 1.2$, $\alpha_{Off} = 1.2$ along with two other values ($\alpha_{On} = 1.4$, $\alpha_{Off} = 1.2$ and $\alpha_{On} = 1.7$, $\alpha_{Off} = 1.2$), will be examined. The performance parameters will be collected, analyzed and presented in the next chapter.

## C. SENSOR NETWORK ARCHITECTURE

In this experiment the WSN is deployed in three different network topologies. These topologies are the star topology, binary tree topology, and chain topology. Depending on the topology, the mesh router function of the SPOTs will be enabled so that it is able to mesh route packets that it received to the desired destination. The routing protocol adopted by Sun SPOT is the AODV routing protocol.

### 1. Star Topology

In the star network topology, the simulation begins with one SPOT connected wirelessly to the BS and an increment of one SPOT after every simulation until six SPOTs are connected to the BS. To obtain a reasonable sample size, each simulation for a particular shaping parameter $\alpha_{On}$ is repeated twenty times. The duration of each simulation is five minutes. The mean throughput, mean interarrival time, mean packet drop, and mean delay are computed from the collected data. Figure 18 depicts the star topology for the experiment. Every SPOT is connected to the PC so that the number of packets sent can be captured for computing mean packet drop.

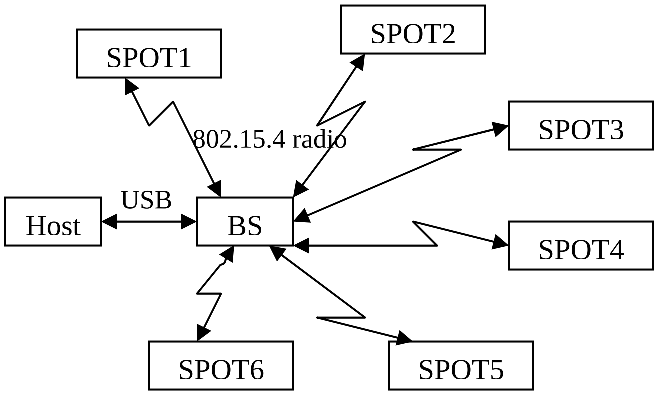


Figure 18. Star Topology

### 2. Binary Tree Topology

A two-level binary tree topology involving one BS and four SPOTS is deployed. Level one SPOTs are configured as mesh routers so that packets transmitted by level two SPOTs will be routed to the BS. A sample size of twenty for each simulation of a particular $\alpha_{On}$ is recorded. The duration of each simulation is five minutes. In this topology, the leaf SPOTs are deployed 40 centimeters (cm) apart from each other such that the SPOTs could not be in radio contact with the host PC. Thus, packets transmitted by the leaf SPOTs could not be captured by the PC via a USB cable. The onboard memory is also too small to support the capturing of the transmitted packets. Due to these reasons, only mean throughput, mean interarrival time and mean delay are computed. Figure 19 depicts the binary tree topology for the experiment.

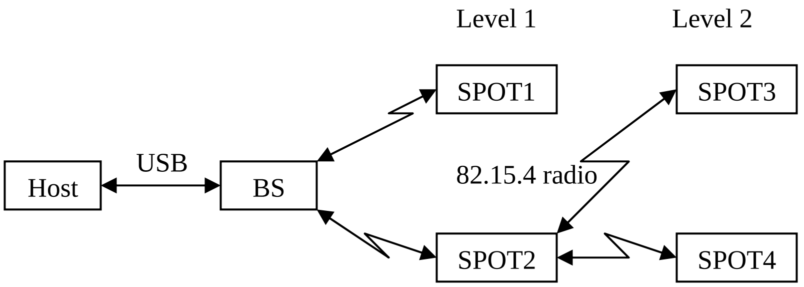


Figure 19. Binary Tree Topology

### 3. Chain Topology

In this network deployment, all the SPOTs are deployed in a line (i.e. one after another). In this topology, all SPOTs are configured as mesh routers so that packets transmitted from the last SPOT can be routed by the SPOT before it and be delivered to the BS. Twenty simulations per shaping parameter are performed and the mean throughput, mean interarrival time and mean delay are computed based on the collected data. The duration of each simulation is five minutes. For the same reasons as explained in the binary tree topology, the SPOTs are located too far (40 cm between each SPOTs) apart to be observed by the PC. Hence, mean packet drop is not computed. Figure 20 depicts the chain topology for this experiment.

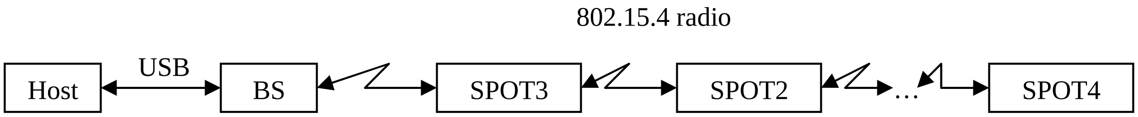


Figure 20. Chain Topology

## D. SPOT APPLICATIONS

The Sun SPOT is a Java platform powered by the Java ‘Squawk” virtual machine (VM). Squawk VM is a VM for Java language that examines better ways of building VMs. It is meant to be used in small, resource-constrained devices [7]. Hence, applications are implemented in Java bytecode. Both BS and SPOTs establish communications using the radiostream protocol provided by the radio communications library in Squawk VM. The radiostream protocol is a socket-like peer-to-peer protocol that provides reliable, buffered, stream-based communication input/output (I/O) between two devices [7]. When the radiostream protocol is implemented, the SPOT will automatically be enabled as a mesh router. It will act as a router to route any packets received from remote SPOTs, which are out of range from the BS, to the BS. The following subsections describe the flow of events in the Java bytecode.

### 1. Base Station (BS)

The BS is programmed to receive packets transmitted from the transmitter (SPOTs). The flow of events in a BS is depicted in Figure 21. It begins with establishing a radiostream connection, which is a bi-directional connection, with all the available SPOTs. This is followed by opening a data input stream and output stream to receive and send data. The BS then checks for available packets at the input stream and if there is a packet available, it will receive the packet, add a receive-time stamp to it, and print it on the screen. The BS will continue to monitor for inputs and terminate only when it is being reset.

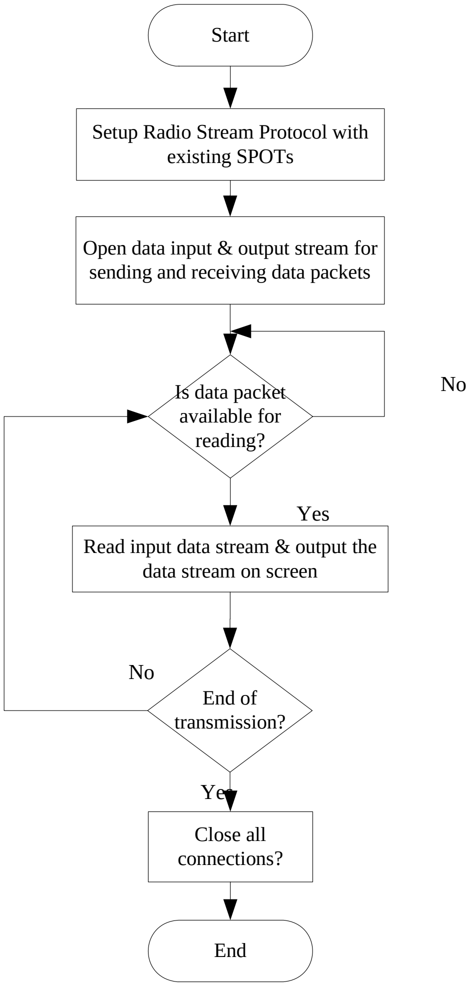

Start
Setup Radio Stream Protocol with existing SPOTs
Open data input & output stream for sending and receiving data packets
Is data packet available for reading?
No
Yes
Read input data stream & output the data stream on screen
No
End of transmission?
Yes
Close all connections?
End

Figure 21. Base Station Flow Chart

### 2. Free-range Sun SPOTs (SPOTs)

The SPOTs are programmed to be transmitters transmitting simulated video traffic to the BS. This is where the Pareto On-Off distribution, as described by Equation (3.3), is implemented to generate simulated video traffic. Packets of 128 bytes each are generated and sent at the line rate of 250 kbps within the computed $t_{on}$ milliseconds. When $t_{on}$ is up, the transmitter will be turned off for the duration of $t_{off}$ milliseconds. When $t_{off}$ is up, a new $t_{on}$ will be computed and the process will repeat itself until the transmitter is reset. This flow of events is captured in Figure 22. Figure 23 illustrates the format of one packet and the fields that make up the packet. As specified by the IEEE

802.15.4 standard, the maximum radio packet size is 128 bytes. In the radiostream protocol, the Sun SPOT uses 49 bytes as packet header which leaves 79 bytes for application data. Hence, the 79 bytes are divided into 4 byte sequence numbers, an 8 byte node ID, an 8 byte sender node time stamp, 1 byte field separators and 56 bytes for artificial data. In actual implementation the 1 byte field separators are not included. However in this thesis, the 1 byte field separators are included for easy of visualization and for Matlab's extraction of data for computing the performance parameters.

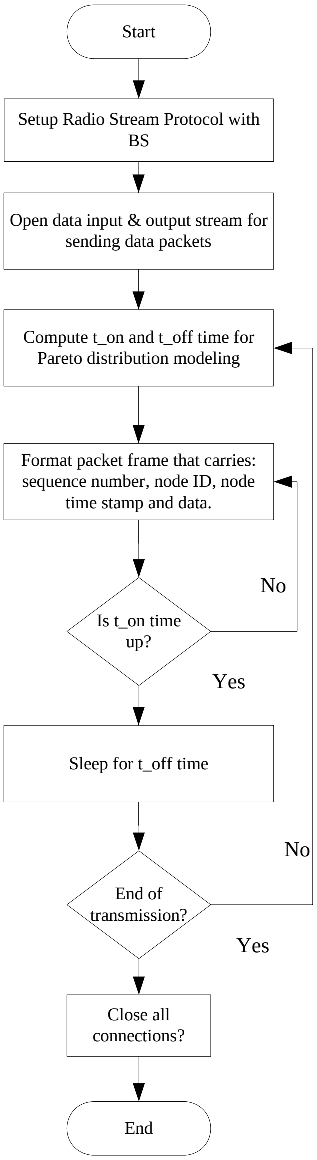

Start
Setup Radio Stream Protocol with BS
Open data input & output stream for sending data packets
Compute t_on and t_off time for Pareto distribution modeling
Format packet frame that carries: sequence number, node ID, node time stamp and data.
Is t_on time up?
No
Yes
Sleep for t_off time
End of transmission?
No
Yes
Close all connections?
End

Figure 22. SPOT Flow Chart.

| Sequence number | Separator | Sender Node ID | Separator | Sender Node Time Stamp | Separator | Artificial Data |
|---|---|---|---|---|---|---|
| 4 bytes | 1 byte | 8 bytes | 1 byte | 8 bytes | 1 byte | 56 bytes |

Figure 23. SPOT Packet Format.

## E. PERFORMANCE PARAMETERS

### 1. Mean Throughput

Throughput is defined as the total amount of useful data received per unit of time. This is a fundamental measure of the performance of a network. In this study, statistical measurement is used to analyze the system mean throughput. The mean throughput in this study is given by Equation (3.5).

$$Mean\,Throughput = \frac{\sum_{i=1}^{Total\,No.of\;simulations}\left(\dfrac{\sum No.of\;\;pkts\,received\;by\;BS \times 128Byte\;per\;\;pkt}{Total\,transmission\,time}\right)_i}{Total\,number\,of\;\;simulations} \quad (3.5)$$

### 2. Mean Interarrival Time

Interarrival time measurement is important as it is an integral part of traffic management. It can be used to investigate abnormal or unexpected network activities. Interarrival time is defined in Equation (3.6). The interarrival time of each packet (pkt) is computed using this equation. The mean interarrival time is given by Equation (3.7).

$$\Delta t_j = t_{j+1} - t_j \quad (3.6)$$

$$Mean\ InterarrivalTime = \frac{\displaystyle\sum_{i=1}^{Total\ No.of\ simulations} \left( \frac{\displaystyle\sum_{j=1}^{No.of\ pkts\ received\ by\ BS-1} \Delta t_j}{No.of\ pkts\ received\ by\ BS-1} \right)_i}{Total\ No.of\ simulations} \qquad (3.7)$$

### 3. Mean Packet Drop

Packet drop can be caused by a number of factors, including the following: signal degradation of the network medium, corrupted packet rejected by the receiver, and an oversaturated network link. A high amount of packet drops reveal poor network performance. In this study, mean packet drop is given by Equation (3.8).

$$Mean\ Packet\ Drop = \frac{\displaystyle\sum_{i=1}^{Total\ No.of\ simulations} \left( \left( \sum_{j=1}^{No.of\ SPOTs} \left( No.of\ pkts\ sent\ by\ SPOTs \right)_j \right) - No.of\ pkts\ received\ by\ BS \right)_i}{Total\ no.of\ simulations} \quad (3.8)$$

### 4. Mean Delay

The delay is the overall time taken from the time that data is transmitted by the source to the time that the designated destination receives. Delay affects applications in many ways. Applications such as video streaming and voice that are sensitive to delay cannot function properly when the delay becomes too long. In this study, mean delay is given by Equation (3.9)

$$mean\ Delay = \frac{\displaystyle\sum_{i=1}^{Total\ No.of\ simulation} \left( \frac{\displaystyle\sum_{j=1}^{No.of\ pkts\ received\ per\ simulation} \left( delay\ per\ pkt \right)_j}{Total\ No.of\ pkts\ received\ per\ simulation} \right)_i}{Total\ No.of\ simulations} \qquad (3.9)$$

## F. SIMULATION PROCEDURE

The following steps describe the simulation procedure:

- Open the Java bytecodes applications on IDE NetBeans.
- Connect the BS and SPOTs to the PC via USB cable.
- Deploy applications onto the BS and SPOTs one at a time. During the process of deploying the applications, the clock of the BS and SPOTs will synchronize with the PC.
- After deploying the application onto the BS and SPOTs, use the run command in NetBeans to activate the simulation. During the simulation, the output of BS and SPOTs will be printed on the output windows of NetBeans.
- After the simulation has stopped, save the outputs.
- After repeating the simulation twenty times, run the Matlab code to compute the mean throughput, mean interarrival time and mean packet drop (for star network topology).

## G. SUMMARY

This chapter provided an over view of the experimental setup. The chapter also provided a detailed discussion on traffic modeling methodology, sensor network architecture, Sun SPOT applications and performance parameters adapted in this thesis. A detailed step-by-step simulation procedure was also presented. The next chapter will present the implementation of Pareto On-Off distribution and follows with discussion and analysis of the simulation results.

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. PERFORMANCE ANALYSIS

This chapter provides a performance analysis of the shaping parameters with respect to the star, binary tree and chain WSN topologies. The analysis will present results in terms of the performance parameters such as mean throughput, mean interarrival time, mean packet drop and mean delay.

As mentioned in previous chapter, the Sun SPOT is a Java platform and is powered by Squawk VM. Therefore, it is not the standard Java and does not contain the standard library for Java programming. In the case of the experiment, the math library in Squawk VM does not contain the math operation required for implementing the Pareto distribution. This thesis uses an alternative, which is the approximation method, to implement the Pareto distribution. The approximation method and its accuracy will be discussed in the following subsection.

## A. APPROXIMATION METHOD FOR IMPLEMENTING PARETO ON-OFF

The implementation of Equation (3.3), which describes the Pareto distribution, requires the math method pow(double a, double b) found in the standard Java math class. This method returns the value of the first argument raised to the power of the second argument (double represents a floating point value). However, Squawk VM is a reduced version of the standard Java. Thus, it does not provide the required pow method.

### 1. Approximating Pareto Distribution

To overcome this limitation, an alternative method has been developed. This alternative method finds an approximate value of $-\frac{1}{\alpha_{on}}$ and $-\frac{1}{\alpha_{off}}$, which is the power of $\left(1-r\right)$. It uses Equation (4.1) to find the integer value of *x* and *y* that will give the minimum difference between the approximate value and the actual value of $-\frac{1}{\alpha}$.

$$
\begin{aligned}
&\left(\tilde{x},\tilde{y}\right)=\underset{(x,y)}{\arg\min}\, g\left(x,y\right)\\
&where\, g\left(x,y\right)=\left|\frac{x}{2^{y}}-\frac{1}{\alpha}\right| \qquad (4.1)\\
&s.t.\, x,y\in\left\{\text{positive integer}\right\}
\end{aligned}
$$

The integer value $\tilde{x}$ and $\tilde{y}$ are obtained by solving Equation (4.1) in Matlab. The equation $\frac{x}{2^{y}}$, which is the exponent of $\left(1-r\right)$, can be separated into $\left(\frac{1}{2^{y}}\right)x$. Thus, Equation (4.2) is the new equation used to compute $t_{on}$ and $t_{off}$. In these equations, $\tilde{y}$ is the number of square root to be applied to $\left(1-r\right)$ and $\tilde{x}$ is the integer power of $\left(\left(1-r\right)^{\frac{1}{2^{\tilde{y}}}}\right)$. The method uses the square root function available in Squawk VM to compute the value of $\left(\left(1-r\right)^{\frac{1}{2^{\tilde{y}}}}\right)$ and the result raised to the power of $\tilde{x}$. To obtain the result of the integer power, a fast version of math.pow written by Patricia Shanahan [25] was used to compute the approximated value of $t_{on}$ and $t_{off}$.

$$t_{on}=\frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{\tilde{y}}}}\right)^{\tilde{x}}}\, or\, t_{off}=\frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{\tilde{y}}}}\right)^{\tilde{x}}} \qquad (4.2)$$

Table 2 shows the value of $\tilde{x}$ and $\tilde{y}$ for computing $t_{on}$ and $t_{off}$ for the respective $\alpha_{on}$ and $\alpha_{off}$ .

| $\alpha$ | $\tilde{x}$ | $\tilde{y}$ | $t_{on}$ |
|---|---|---|---|
| $\alpha_{on} = 1.2$ | 218453 | 18 | $t_{on} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{18}}}\right)^{218453}}$ |
| $\alpha_{off} = 1.2$ | 218453 | 18 | $t_{off} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{18}}}\right)^{218453}}$ |
| $\alpha_{on} = 1.4$ | 93623 | 17 | $t_{on} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{17}}}\right)^{93623}}$ |
| $\alpha_{off} = 1.2$ | 218453 | 18 | $t_{off} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{18}}}\right)^{218453}}$ |
| $\alpha_{on} = 1.7$ | 77101 | 17 | $t_{on} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{17}}}\right)^{77101}}$ |
| $\alpha_{off} = 1.2$ | 218453 | 18 | $t_{off} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{18}}}\right)^{218453}}$ |
| $\alpha_{on} = 1.9$ | 137971 | 18 | $t_{on} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{18}}}\right)^{137971}}$ |
| $\alpha_{off} = 1.25$ | 209715 | 18 | $t_{off} = \frac{1}{\left(\left(1-r\right)^{\frac{1}{2^{18}}}\right)^{209715}}$ |

Table 2. Approximate Value of Respective $t_{on}$ and $t_{off}$ .

### 2. Accuracy of the Approximation Method

To verify the accuracy of the approximation method, the difference between the approximation method and the actual method at all values of *r* were computed. The computation was based on Table 2. Figures 24, 25, 26 and 27 show the error between the approximation method and the actual method. It can be seen that the error is very small and it increases minimally as the value of *r* approaches 1. This is a legitimate result as it can be seen from Equation (3.3) that as *r* approaches 1, the value of $t_{on}$ or $t_{off}$ approaches infinity. Hence, the error between the two methods will also approach infinity. Figure 28 shows a close-up error plot for the range of *r* between 0.999 and 1 for $\alpha_{on} = 1.9$ and $\alpha_{off} = 1.25$.

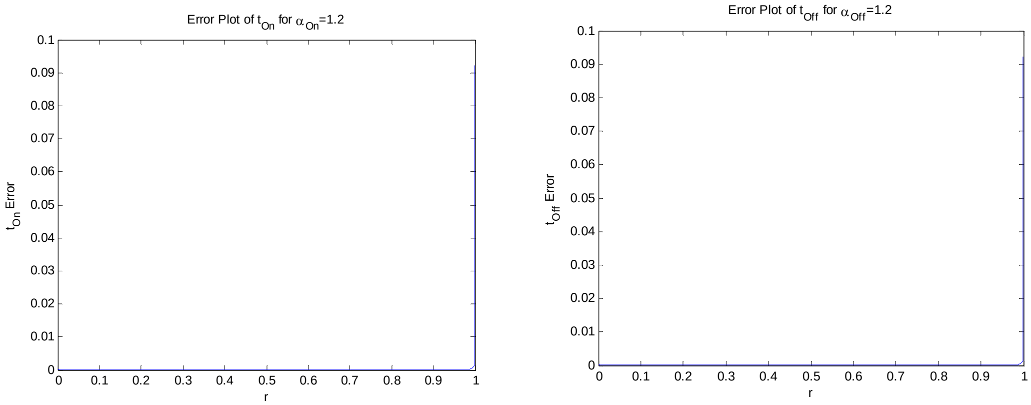


Figure 24. Error Plot for $\alpha_{on} = \alpha_{off} = 1.2$.

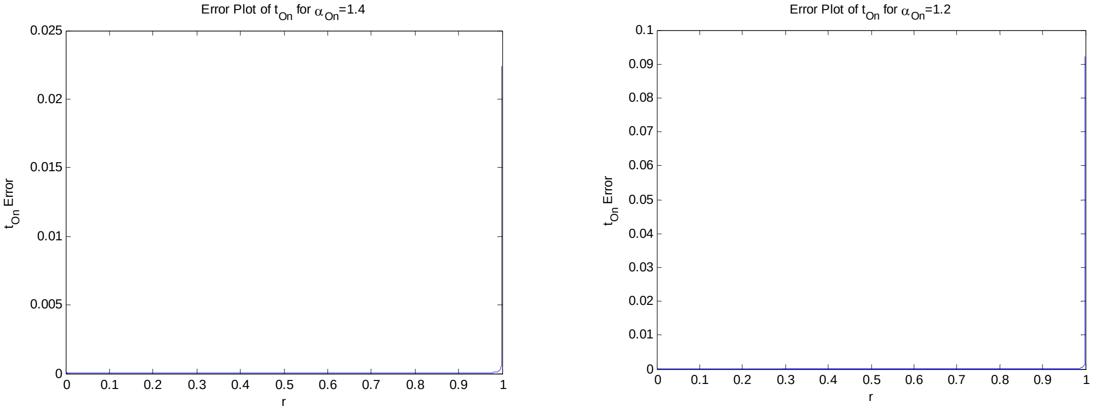


Figure 25. Error Plot for $\alpha_{on} = 1.4\, and\, \alpha_{off} = 1.2$.

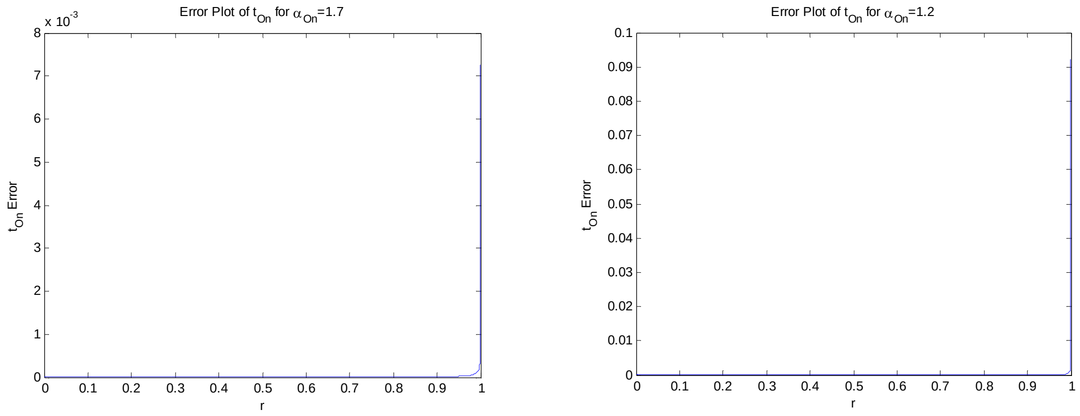


Figure 26. Error Plot for $\alpha_{on} = 1.7$ *and* $\alpha_{off} = 1.2$.

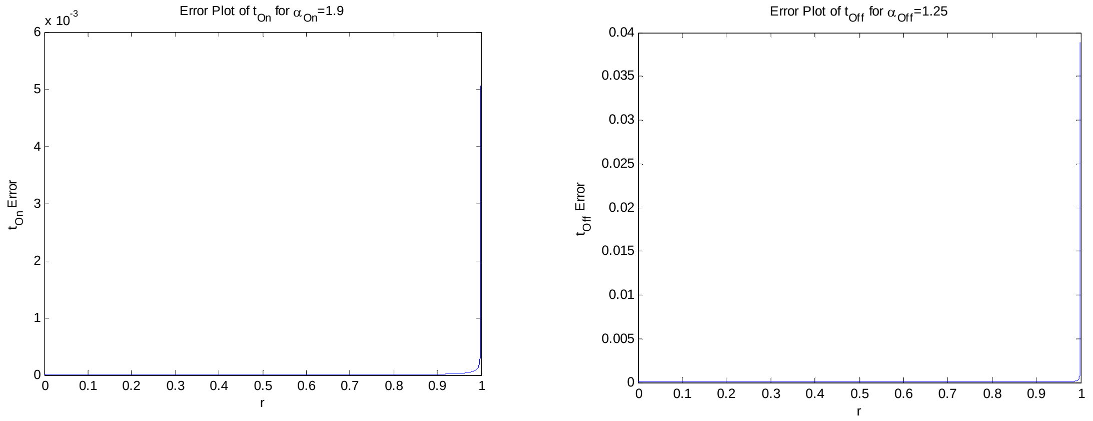


Figure 27. Error Plot for $\alpha_{on} = 1.9$ *and* $\alpha_{off} = 1.25$.

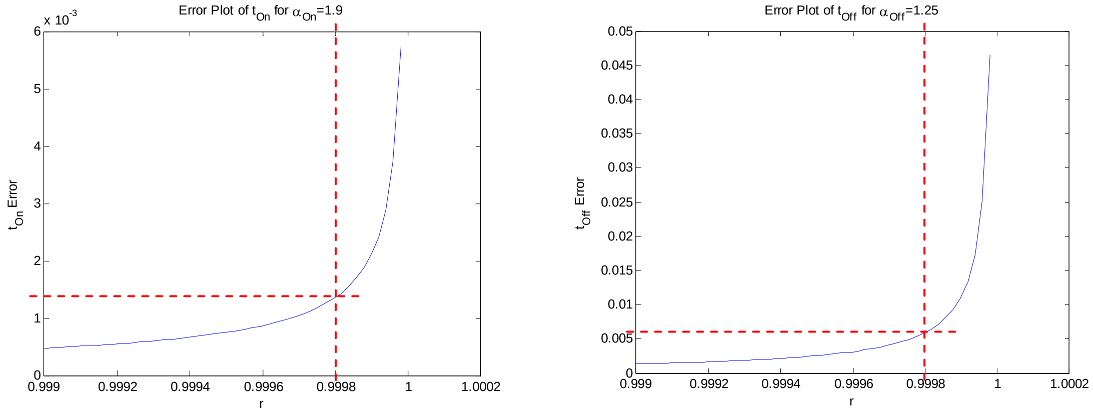


Figure 28. Close-up Error Plot of $\alpha_{on} = 1.9$ *and* $\alpha_{off} = 1.25$ for $r$ between 0.999 and 1

As $r$ is a random number uniformly distributed between 0 and 1, errors of such magnitude will only happen once in every one thousand or more simulations. Due to this, it is assessed that an error of such magnitude will not affect the characteristics of the Pareto distribution. To further prove the error is not enough to affect the accuracy of Pareto distribution, error plots for all value of $\alpha$ at three values of $r$ are shown in Figures 29, 30 and 31.

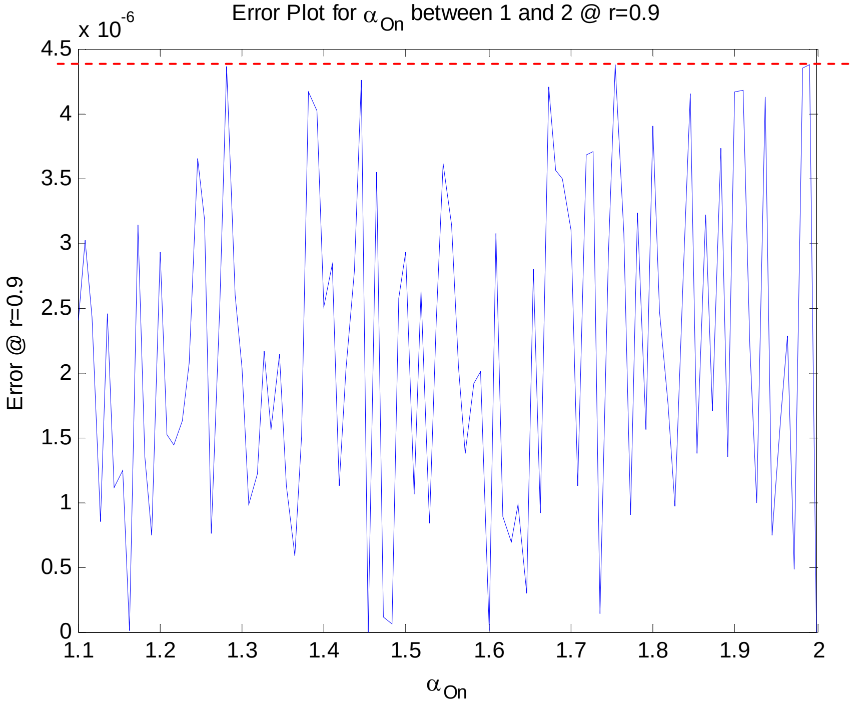


Figure 29. Error Plot for $\alpha_{on}$ between 1 and 2 at $r$=0.9.

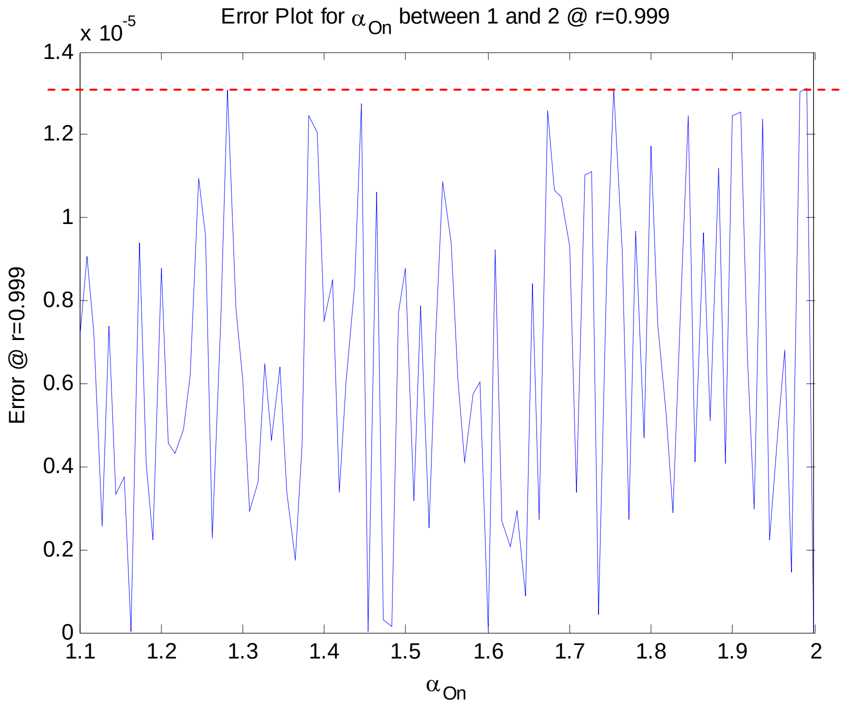


Figure 30. Error Plot for $\alpha_{on}$ between 1 and 2 at *r*=0.999.

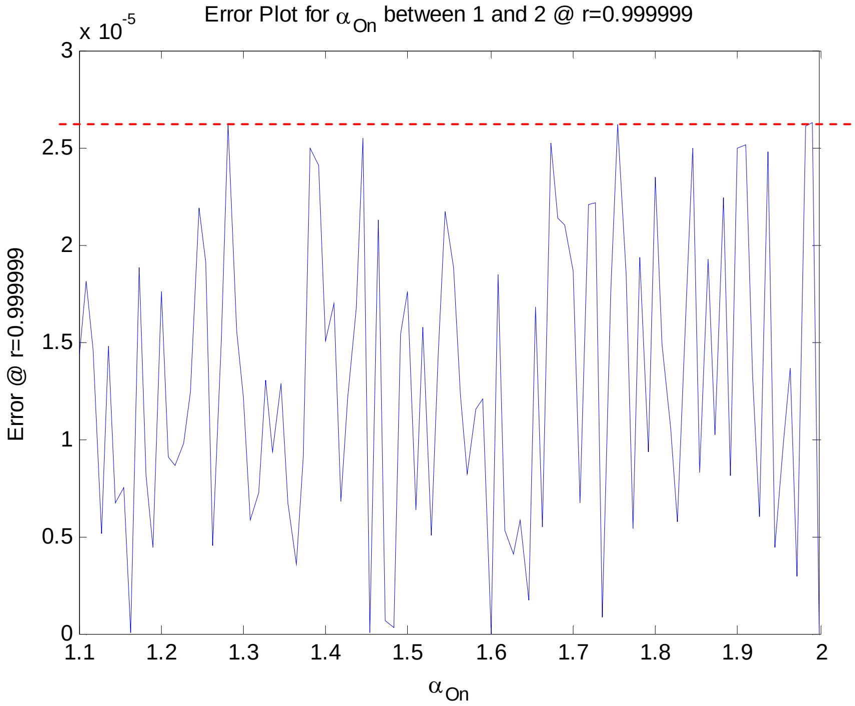


Figure 31. Error Plot for $\alpha_{on}$ between 1 and 2 at *r*=0.999999

Again it is observed that the error value increases as *r* approaches 1. From Figure 31, it can be seen that the highest error is approximately $2.6 \times 10^{-5}$.

## B. SIMULATION RESULTS

This section presents the simulation results and analysis of the three topologies.

### 1. Star Topology

The simulation results of the star topology are shown in Figures 32, 33, 34 and 35.

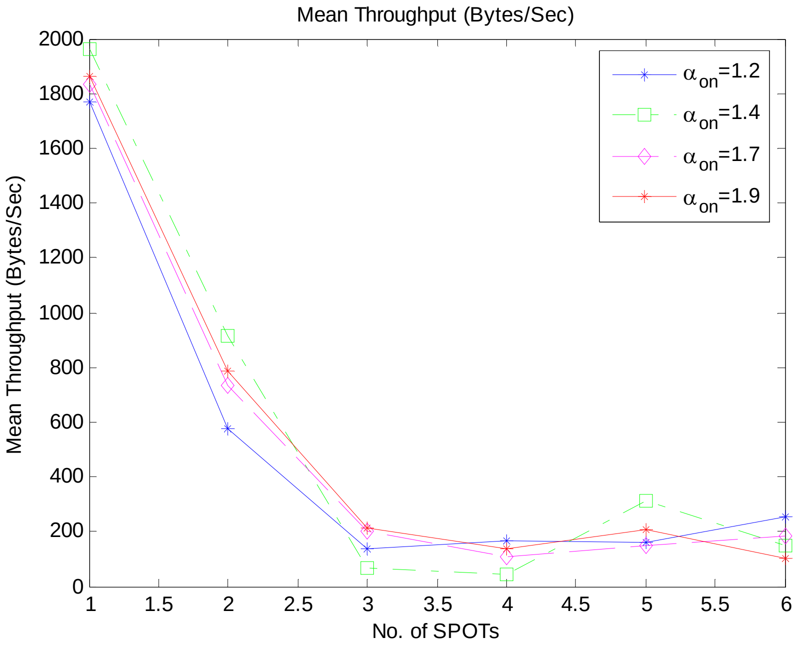


Figure 32. Star Topology - Mean Throughput (bytes/second) Plot.

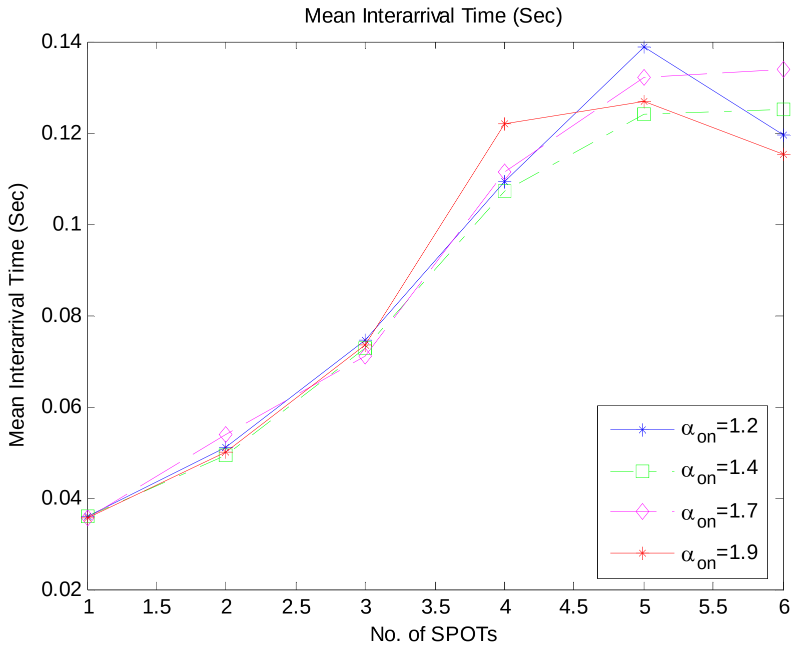


Figure 33. Star Topology - Mean Interarrival Time (sec) Plot.

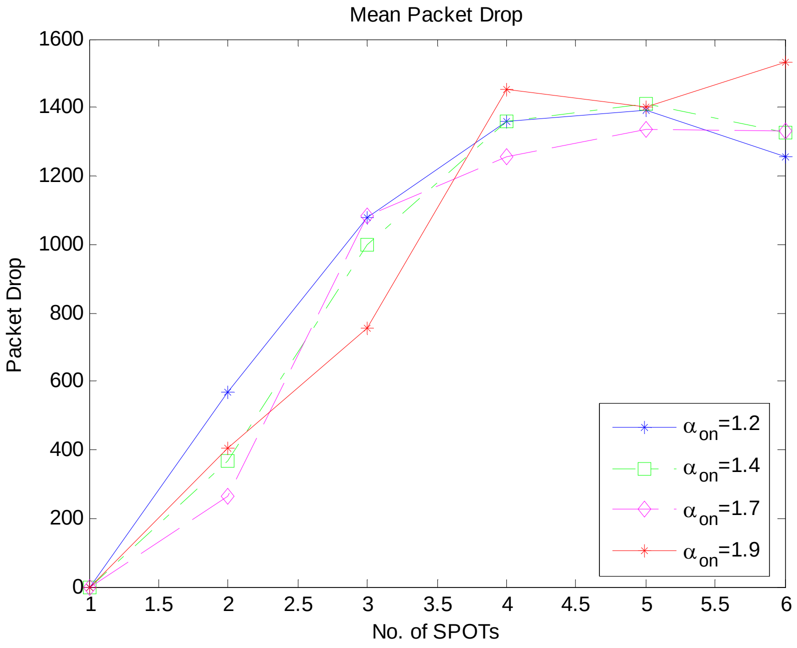


Figure 34. Star Topology - Mean Packet Drop Plot.

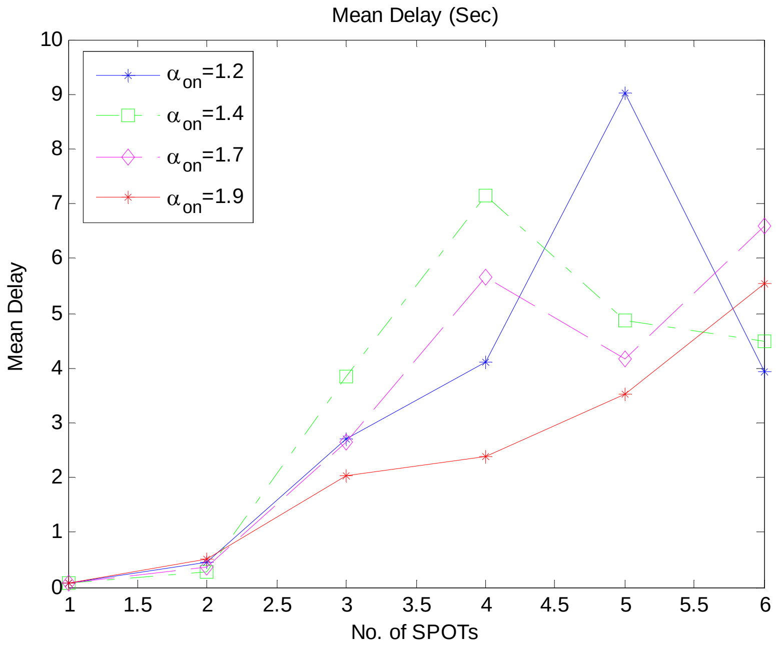


Figure 35. Star Topology - Mean Delay (sec) Plot.

From Figure 32, it can be seen that when there was only one SPOT connected to the BS, the highest throughput was provided by $\alpha_{On} = 1.4$, and the lowest throughput was provided by $\alpha_{On} = 1.2$ with a spread of 195 bytes/second. The mean throughput continued to reduce by half when two SPOTs were connected to the BS. When three or more SPOTs were connected to the BS the throughput reduced to 211 bytes/second and below.

Figures 33 and 34 show the mean interarrival time and packet drop performance of the star topology. The mean interarrival time increased linearly between one SPOT to four SPOTs; however, it plateaued from four SPOTs onwards. On a similar note, the mean packet drop also increased when more than one SPOTs were connected to the BS. However, the increase in mean packet drop was not linear and spreading of approximately 300 packets are observed. The mean packet drop also plateaued from four SPOTs onwards with a spreading of approximately 300 packets, except for five SPOTs. These results reveal that the BS was still receiving packets even when four or more SPOTs were connected to it. However, the packets were received at a very low rate. This is due to the frequent request for retransmission as a result of high occurrences of traffic collision between SPOTs. This inevitably increase the mean interarrival time and packet drop.

During the process of the simulations it was observed that occasionally the BS will run out of memory when there were three or more SPOTs in the network (this phenomenon also happened in binary tree and chain topology). Thus, the simulation time reduced to about two to three minutes instead of five minutes. According to technical advice from the Sun SPOT forum [26], in theory the BS will turn off the radio when it had received 1,000 packets backed up waiting for the host application's attention. In this thesis, during each simulation, all SPOTs were programmed with the same traffic characteristic and transmitted at the same rate. Although all the SPOTs were activated to begin traffic generation and transmission one at a time, due to the present of *r* in equation (4.2), which is a random number, there will be times where their On period overlaps with each other. During the overlapping of the On period, the buffer of the BS may become saturated and the host application unable to de-queue the received packets. Thus results in the BS running out of memory and turning off the radio. This also resulted in the BS dropping all subsequent packets. Hence, increases the mean packet drop.

Figure 35 shows the mean delay plot of the star topology. It can be seen that the spread of delay from three SPOTs onwards was quite large. As a result of such large variation of delay, the results are inconclusive. In order to have better understanding of the phenomena, an in-depth investigation is required.

In summary for the star topology, $\alpha_{On} = 1.2 - 1.9$ produces self-similar and heavy-tailed traffic. However, the results of the simulations are inconclusive and therefore are unable to pinpoint which shaping parameter has better performance. Hence, an in-depth investigation is required to establish some understanding of this aspect. Finally, the topology provides a very wide coverage in term of the breadth of the network.

### 2. Binary Tree Topology

The simulation results of the binary tree topology are shown in Figures 36, 37 and 38.

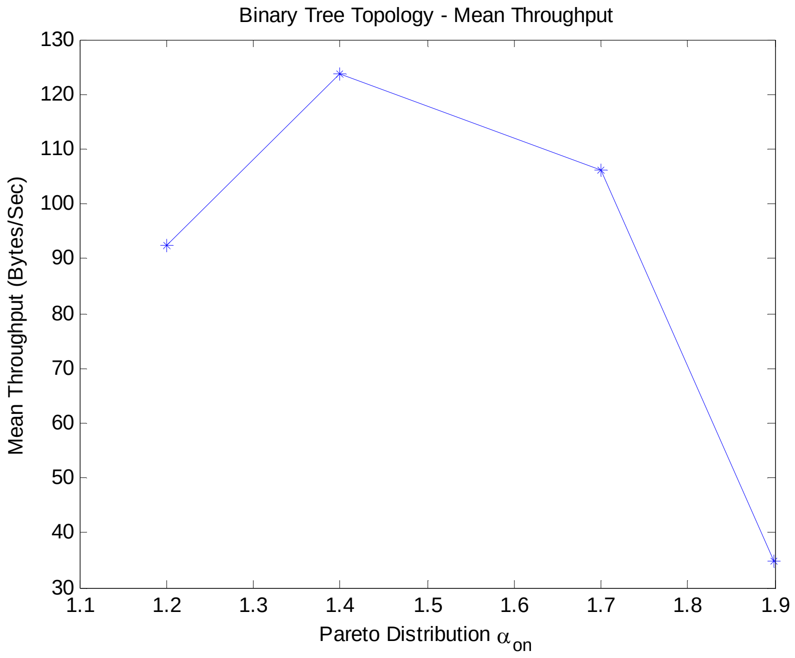


Figure 36. Binary Tree Topology – Mean Throughput (bytes/second) Plot.

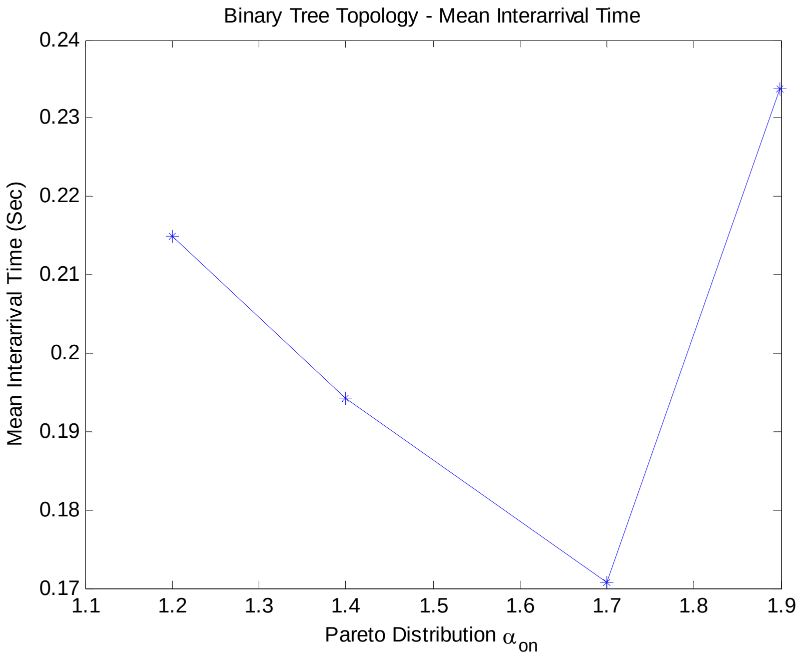


Figure 37. Binary Tree Topology – Mean Interarrival Time (sec) Plot.

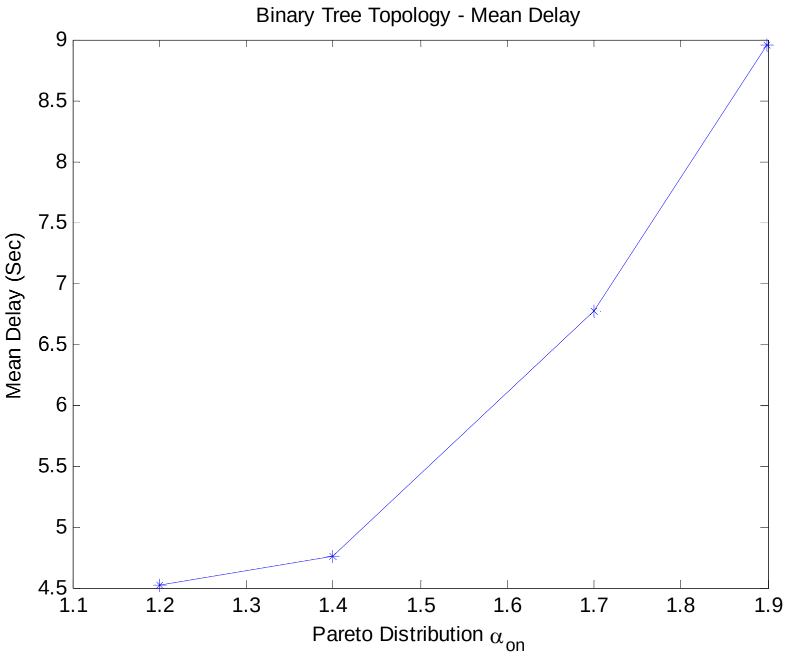


Figure 38. Binary Tree Topology – Mean Delay (sec) Plot.

The binary tree network topology was set up in two levels with one BS and four SPOTs. The topology has a very similar throughput performance as compared to the four SPOT star topology. In this topology, the leaf SPOTs are deployed 40 cm apart from each other, thus they could not be connected to the host PC. For this reason, packets transmitted by the SPOTs could not be captured for analysis. The onboard memory is also too small to support the capturing of the transmitted packets. Hence, mean packet drop could not be computed. A point to note is that a three-level binary tree topology could not be established because data could not be captured and hence could not be evaluated.

It is interesting to note that in this topology (see Figures 36 and 37), the mean throughput values of $\alpha_{on} = 1.2$ and $\alpha_{on} = 1.9$ were the lowest and their mean interarrival times were among the highest. Both parameters were on the opposite edge of the self-similarity range ($H_{1.2} = 0.9$, $H_{1.9} = 0.55$). Whereas, $\alpha_{on} = 1.4$ and $\alpha_{on} = 1.7$ ($H_{1.4} = 0.8$, $H_{1.7} = 0.65$) are in the mid range of the self-similarity range. These two parameters produced higher throughput with lower interarrival times. Figure 38 shows the mean delay of the topology. It is observed that the mean delay increases as the level of self-similarity reduces. To understand this phenomenon, further research has to be performed to identify the root course of the increasing delay.

In summary, for the binary tree topology the results revealed that the performance of shaping parameter selected at the mid-range of the self-similarity range were better than those at both edges of the range. From these results, it is assessed that the best performing shaping parameter is 1.4. However, via interpolation, a better performance parameter could be from the range of 1.5 to 1.6. More investigation is necessary to confirm this assessment. Finally, the two-level binary tree topology provides good breadth as well as some depth coverage of the network.

### 3. Chain Topology

The simulation results of the chain topology with three SPOTs and four SPOTs are shown in Figures 39, 40, 41 and 42, 43, 44, respectively. Both sets of figures reveal an inverse in mean throughput performance as compared to the binary tree topology. From Figure 39 the shaping parameters with higher self-similarity, i.e. $\alpha_{on} = 1.2, 1.4$, suffered a lower throughput than those at the lower self-similar range, i.e. $\alpha_{on} = 1.7, 1.9$. Figure 42 shows that $\alpha_{on} = 1.4, 1.9$ have similar performance. These results show that, in this topology, shaping parameters with lower self-similarity characteristic performance better. In this topology, $\alpha_{on} = 1.7$ emerged as the best performer with the highest mean throughput, lowest mean interarrival time and mean delay. It was also observed from Figure 42 that the mean throughput reduced by more than half when there were four SPOTs in the chain topology. However for mean delay (see Figure 44), the results are not coherent with that of Figure 41. In theory, the mean delay for a 4 SPOTs chain system should have a larger mean delay as compared to a 3 SPOTs system. However, the results show otherwise hence, further research need to be performed to investigate the phenomenon.  For the same reasons as mentioned in the binary tree topology, the SPOTs are deployed too far apart from the host PC such that the SPOTs could not be connected to the PC via USB cable. Hence, mean packet drop was not computed as the transmitted packets could not be captured for analysis.

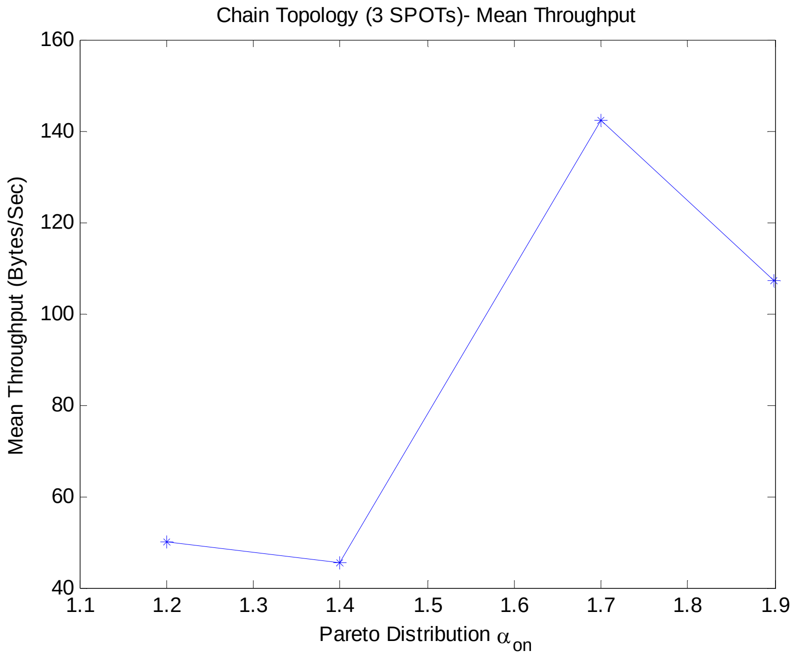


Figure 39. Chain Topology (3SPOTs) - Mean Throughput (bytes/second).

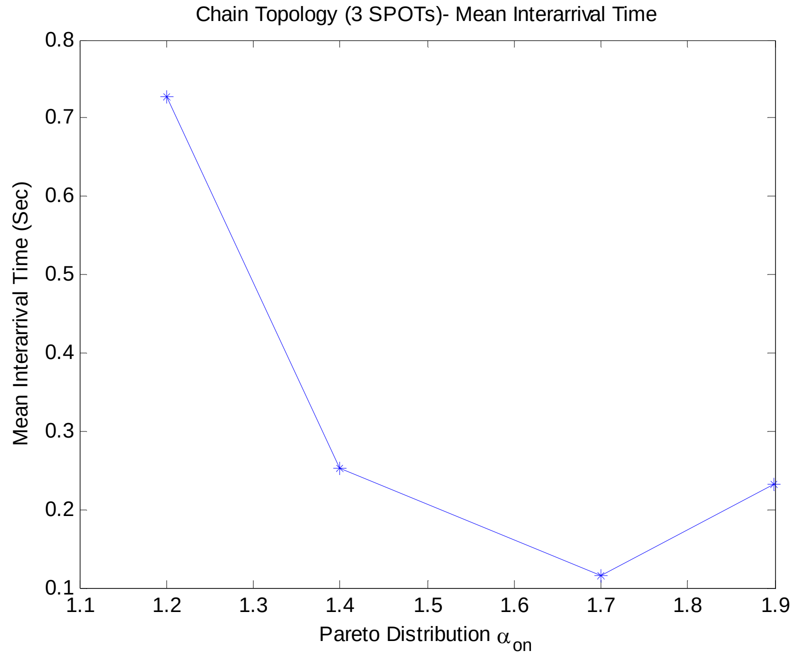


Figure 40. Chain Topology (3SPOTs) - Mean Interarrival Time (sec) Plot.

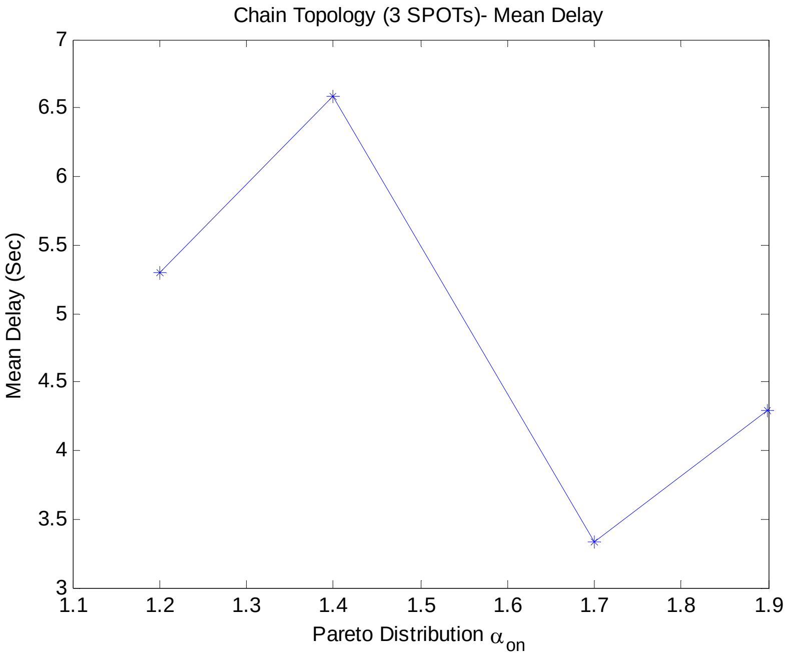


Figure 41. Chain Topology (3SPOTs) - Mean Delay (sec) Plot.

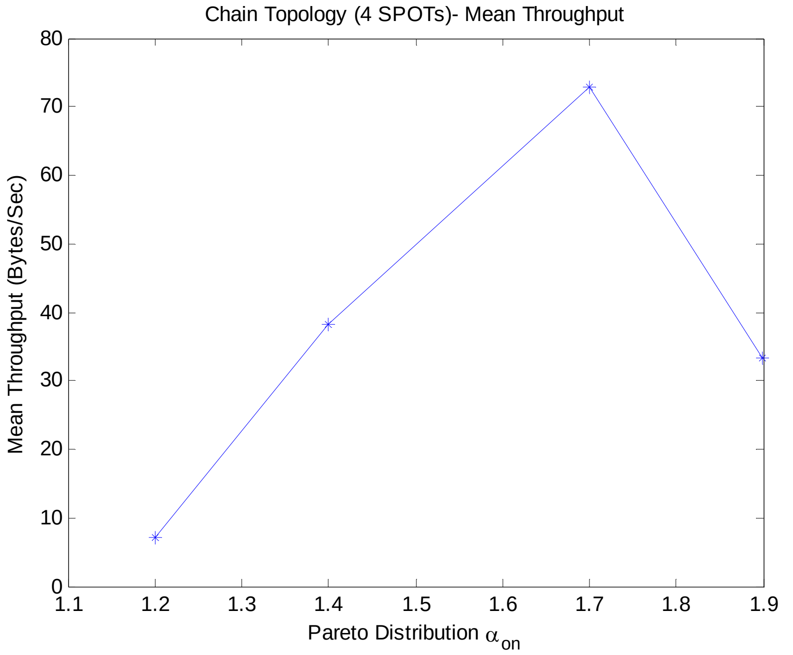


Figure 42. Chain Topology (4SPOTs) - Mean Throughput (bytes/second) Plot.

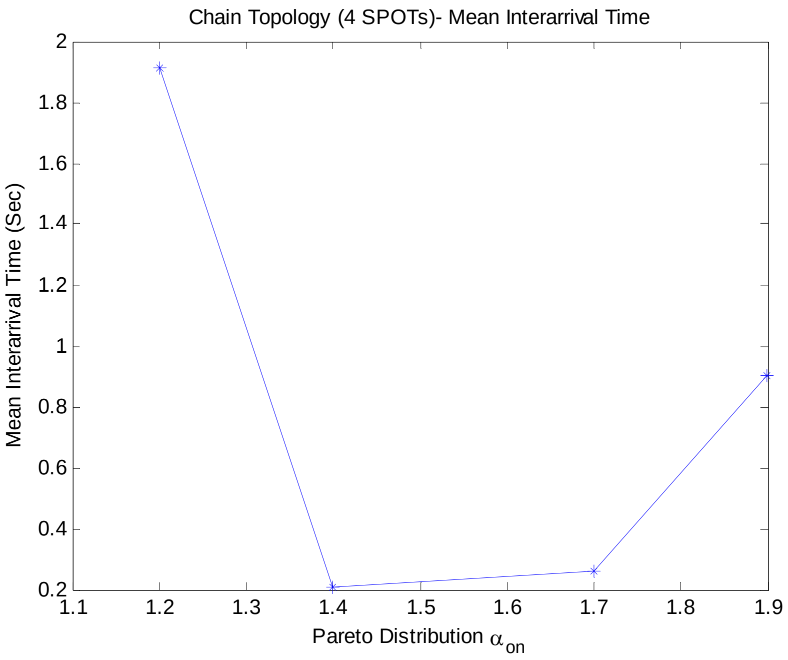


Figure 43. Chain Topology (4SPOTs) - Mean Interarrival Time (sec) Plot.

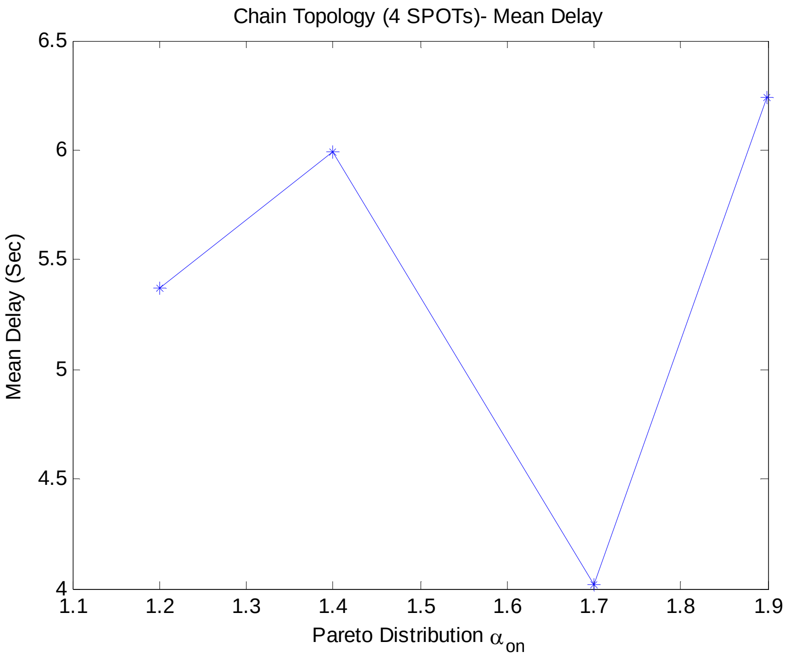


Figure 44. Chain Topology (4SPOTs) - Mean Delay (sec) Plot.

In summary, for the chain topology shaping parameters with higher self-similarity suffered low mean throughput while those with lower self-similarity had higher mean throughput. The performance of the three SPOTs chain was similar to that of the binary tree topology. Nonetheless, it provided very good depth coverage as compared to the binary tree topology. The only drawback with this topology was that it does not provide sufficient breadth of coverage.

### 4. Summary

In this chapter, the approximation method for implementing the Pareto Distribution on the Sun SPOT Java VM and the simulation results of the topologies were presented. The star topology provided good breadth of coverage; however, the simulation results were inconclusive and thus the best performing shaping parameter could not be ascertained. The two-level binary tree topology provided good breadth of coverage and some depth coverage. The simulation results reveal that the shaping parameter in the mid self-similarity range provided good mean throughput performance with low mean interarrival time. The chain topology had an inverse performance as compared to the binary tree topology. The simulation results showed that the shaping parameters at the lower end of the self-similar range provide better performance. It also showed that the performance of the three SPOTs chain topology was very similar to that of the binary tree topology. However, the advantage of the chain topology was that it provides better depth coverage.

Having seen the performance of all three topologies, Table 3 provides a recommendation of the topologies and their shaping parameter that will provide the desired breadth and depth network coverage.

| Topology | Recommended | | | Mean throughput (bytes/sec) | Mean interarrival time (sec) | Mean delay (sec) |
|---|---|---|---|---|---|---|
| | $\alpha_{On}$ | Max breadth of coverage | Max depth coverage | | | |
| Star | 1.4 | 2 | 1 | 917 | 0.049 | 0.26 |
| Binary Tree | 1.4 | 2 | 2 | 124 | 0.194 | 4.76 |
| Chain | 1.7 | 1 | 3 | 142 | 0.116 | 3.33 |

Table 3. Architecture Recommendation for IEEE 802.15.4 and the Zigbee Standard.

The decision to select the type of topology is based on the trade-off between breadth and depth of coverage. If it is desired to have depth of coverage then the chain topology should be adopted. If the decision is to have good breadth and some depth of coverage then a binary tree topology will be good. Otherwise, if only breadth of coverage is desired, a star topology is excellent.

Each shaping parameter performs differently when applied in different network topology. Table 4 provides a list of recommended topologies that provide the best performance with respect to a given shaping parameter. For example, $\alpha_{On} = 1.4$ has two best performance topologies, namely star and binary tree topology. One could implement the star topology combined with the binary tree topology to expend on the breadth and depth of coverage.

| $\alpha_{On}$ | Recommended Topologies | Min achievable Throughput (bytes/sec) | Max Mean Interarrival Time (sec) | Max Mean Delay Time (sec) |
|---|---|---|---|---|
| 1.2 | Star | 163 | 0.14 | 9 |
| 1.4 | Star | 917 | 0.049 | 0.26 |
|  | Binary Tree | 124 | 0.194 | 4.76 |
| 1.7 | Chain | 142 | 0.116 | 3.33 |
| 1.9 | Star | 103 | 0.127 | 5.54 |

Table 4. Topology Recommendation with Respect to Shaping Parameter.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CONCLUSION AND RECOMMENDATION

## A. CONCLUSIONS

This thesis provided a performance analysis of the three WSN topologies, namely star topology, binary tree topology and chain topology by means of simulation. The simulation used a Pareto distribution to model video traffic over the WSN. The shaping parameters under investigation in the study were $\alpha_{On} = 1.2, 1.4, 1.7, 1.9$. The shaping parameter $\alpha_{On} = 1.2$ is recommended for modeling Ethernet traffic while $\alpha_{On} = 1.9$ is recommended for modeling MPEG-2 video traffic. These four shaping parameters fall in the range of self-similarity. The performance parameters used in the study were mean throughput, mean interarrival time, mean packet drop, and mean delay.

The sensor used in the study was the Sun Small Programmable Object Technology (Sun SPOT) kit. The kit consists of a Base Station (BS) and two free-range SPOTs. Each sensor has a processor that runs on the Jave VM "Squawk" and which serves as an IEEE 802.15.4 wireless network node.

In general, the star topology provided good breadth of coverage. In this study, the simulation results for each shaping parameter were very close and as such the results were inconclusive. Thus, the best-performing shaping parameter could not be clearly ascertained. The two-level binary tree topology provided good breadth of coverage and some depth coverage. The simulation results revealed that a shaping parameter in the mid self-similarity range (i.e., $\alpha_{On} = 1.4, 1.7$) provided good mean throughput of 124 bytes/sec and 106 bytes/sec, respectively. This was accompanied with low mean interarrival time of 0.19 sec and 0.17 sec, respectively. On the other hand, the other two shaping parameters (i.e. $\alpha_{On} = 1.2, 1.9$) which are at the opposite edges of the self-similarity range, faired poorer. Their mean throughputs were 92 bytes/sec and 35 bytes/sec, respectively, and their mean interarrival times were 0.22 sec and 0.23 sec, respectively. The best performing shaping parameter observed in this study was $\alpha_{On} = 1.4$. An inverse mean throughput performance was observed in the chain topology as compared to the binary tree topology. The simulation results showed that the shaping parameters at the lower end (i.e., $\alpha_{On} = 1.7, 1.9$) of the self-similar range provided better performance. In the three SPOT chain topology, the mean throughput values with respect to $\alpha_{On} = 1.7, 1.9$ were 142 bytes/sec and 107 bytes/sec, respectively, while the mean interarrival times were 0.12 sec and 0.23 sec, respectively. The results also showed that the performance of the three SPOT chain topology was very similar to that of the binary tree topology. In the four SPOT chain topology, the mean throughput was reduced by approximately one-half. Thus, the mean throughputs were 73 bytes/sec and 33 byte/sec, respectively, and the mean interarrival times were 0.26 sec and 0.9 sec, respectively. In this topology, the observed best performing shaping parameter was $\alpha_{On} = 1.7$. In terms of range coverage, the chain topology was advantageous as it provided better depth coverage. The only drawback with this topology was that it did not provide sufficient breadth of coverage.

With these simulation results, a table (see Table 3) of recommendations for IEEE 802.15.4 and the Zigbee type of architecture was presented in Chapter IV. The table recommended the type of topology to adopt dependent upon the desired breadth and depth of coverage of the network. It also presented the best performing shaping parameter of the topologies.

## B. RECOMMENDATIONS

From the simulation results presented in Chapter IV, the deepest coverage provided by the chain topology with the best mean throughput was only three hops away from the BS. For IEEE 802.15.4 and the Zigbee standard, it would mean that the maximum range of the network was approximately 30 meters. To expand the operating range of the network, there is a need to explore the topic of topology creation. In a mesh network environment, a mesh point (MP) is activated to discover existing mesh networks so that it can associate with them [27]. If there is no existing mesh network then it must be able to initiate a new network without user intervention. There are two approaches to discover a network, namely passive and active. The passive approach discovers a network based on the reception of beacon messages. It involves listening to all possible communication channels. The disadvantage of this approach is that it takes a longer discovery process. The active approach discovers a network by means of sending probing messages. It has a shorter response time and thus allows a fast discovery of topology. The two approaches results in basic connectivity between nodes in the network after which the MP will have to form an Extended Service Set (ESS) network by associating with the neighboring node. Within the ESSs, gateway nodes maybe assigned to connect to other ESS thus forming a larger ESS. The end result is expanding the operating range of the network.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES

[1] Dr. James A. Freebersyser and Joseph P. Macker, “Realizing the Network-Centric Warfare vision: Network Technology Challenges and Guidelines”. http://cs.itd.nrl.navy.mil/pubs/docs/netcentric_milcom01.pdf. Last accessed 30 September 2007.

[2] G. Varatkar and R. Marculescu, “Traffic Analysis for On-Chip Networks Design of Multimedia Applications,” *Proc. Design Automation Conf. (DAC)*, pp. 10-517, June 2002.

[3] Partha Pratim Pande, Grecu, C., Jones, M., Ivanov, A., and Saleh, R., “Performance Evaluation and Design Trade-Offs for Network_on_Chip Interconnect Architectures,” IEEE Trans. on Communications, Vol. 54, Issue 8. pp. 1025-1040, August 2005.

[4] E. Culurciello,; Joon Hyuk Park; A. Savvides,; “Address-Event Video Streaming over Wireless Sensor Networks,” Circuit and System, 2007. ISCAS 2007. IEEE International Symposium on 27-30 May. Pp. 849-852. 2007.

[5] Ian. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam and Erdal Cayirci, “A Survey on Sensor Networks” Communication Magazine, IEEE, Vol. 40 Issue 8, pp. 102-114, August 2002.

[6] Holger Karl, 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.

[7] Sun Microsystems, Inc., “Sun$^{TM}$ Small Programmable Object Technology (Sun SPOT) Theory of Operation,” November 15, 2006.

[8] J. M. Rabaey, M. J. Ammer, J. L. Da Silva Jr., D. Patel, S Roundy, “Picoradio Supports Ad-hoc Ultra-Low Power Wireless Networking,” IEEE Computer Magazine, Vol. 33, Issue 7, pp. 42-48, July 2000.

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

[10] E. M., Royer, Chai-Keong Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Network” Personal Communications, IEEE, Vol. 6 Issue 2, pp. 46-55, April 1999.

[11] C. E. Perkins and P. Bhagwat "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, "Computer Communications Review, October 1994, pp. 234-244.

[12] C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla. "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," Proceedings of IEEE SICON'97, April 1997, pp. 197-211.

[13] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Network," ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communications Network, Oct. 1996. pp. 183-97.

[14] A. S. Tanenbaum, Computer Networks 3rd ed. Ch. 5, Englewood Cliffs, NJ: Prentice Hall, 1996, pp. 357-58.

[15] C. E. Perkings and E. Royer. "Ad-Hoc On-Demand Distance Vector Routing," Proceeding of 2nd IEEE workshop on Mobile computing & Systems and Applications, February 1999.

[16] David B. Johnson, David A. Maltz, and Josh Broch. "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," Ad Hoc Networking, edited by Charles E. Perkins, chapter 5, pp. 139-172, Addison-Wesley 2001.

[17] J. Broch, D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet draft, draft-ietf-manet-dsr-01.txt, December 1998 (work in progress).

[18] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proc. INFOCOM '97, April. 1997.

[19] William Stallings. Data and Computer Communications, Eighth Edition, Chapter 15.2 pp. 451-455, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2007.

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

[21] Wikipedia. http://en.wikipedia.org/wiki/Network_topology. Accessed on October 10, 2007.

[22] F. L. Lewis, Smart Environments: Technologies, Protocols, and Applications, chapter 2, Edited By Diane Cook and Sajal Das, John Wiley, New York, 2004.

[23] Leonel Tedesco, Aline Mello, Leonardo Giacomet, Ney Calazans, Fernando Moraes, “Application Driven Traffic Modeling for NoCs,” Preceeding of ACM SBCCI’06, August 28 – September 1, 2006, pp.62-67.

[24] William Stallings. High-Speed Networks And Internets Performance and Quality of Service, Second Edition, Chapter 9.3 pp. 232-237, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2002.

[25] mindprob.com. http://mindprob.com/jgloss/power.html. Accessed on November 7, 2007

[26] Sunspotworld.com. www.sunspotworld.com/forums/viewtopic.php?p=2862#2862). Accessed on November 8, 2007.

[27] S.M., Faccin, C., Wijting, J., Kenckt, A., Damle, “Mesh WLAN Networks: Concept and System Design” Wireless Communications, IEEE, Vol 13, Issue 2. pp 10-17, April 2006.

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 Weilian Su
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

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

6. MAJ Mak Wai Yen
   Singapore Armed Forces (SAF)
   Singapore