FACULTY OF COMPUTING & INFORMATION MANAGEMENT

AN APPROACH FOR DETERMINING ENERGY EFFICIENT SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS.

MWANGI PETER MAINA

REG NO KCA 13/01715

A RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF MASTER OF SCIENCE IN DATA COMMUNICATION IN THE FACULTY OF COMPUTING AND INFORMATION MANAGEMENT AT KCA UNIVERSITY

KCA -UNIVERSITY

DECLARATION

DECLARATION BY THE CANDIDATE

I declare that this research is my original work and has not been previously published or submitted in any university for the award of degree. I also declare that this Research contain no material written or published by other expect where references have been made and author acknowledged.

Mwangi Peter Maina Signature……………………. Date……………………

REG/NO: KCA 13/01715

DECLARATION BY THE SUPERVISORS

This research has been submitted for examination with our approval as University Supervisors.

Supervisor

Dr Simon Mwendia Signature……………………… Date…………….………

Faculty of Computing & Information Management

KCA University, Nairobi, Kenya.

DEDICATION

I wish to dedicate this research to my lovely family for their ever-growing support and inspiration in my life especially during the duration of my studies at KCA University.

Abstract

Sensor networks are one of the dominant technology trends in the coming decades and the use of wireless Sensor Networks (WSNs) is bringing huge changes in data gathering, processing and propagation of different environments and applications. These sensor networks are composed of hundreds, and potentially thousands of tiny sensor nodes, functioning autonomously, and in many cases, with limited resources (i.e. computation, storage, and battery) this shorten the life span of sensor nodes. Cost constraints and the need for ubiquitous, invisible distributions will result in small sized, resource-constrained sensor nodes. With this constraint, it is very hard to implement security protocols. Cost constraints and the need for ubiquitous, invisible distributions will result in small sized, resource-constrained sensor nodes. With this constraint, it is very hard to implement security protocols.

The research was aimed at evaluating wireless senor network security protocol in terms of their energy efficient and determine on that has better energy consumption.

The research simulates using NS-2 because it is an open source, discrete-event network simulator that provides support for a simulation of main protocols, routing, multicast protocols for wired and wireless networks.

Acknowledgements

My sincere gratitude goes to my Supervisor Dr Simon Mwendia who took his precious time in leading me through the research process. I am deeply indebted to you for all the insightful ideas, suggestions, feedback I have received throughout the period of writing this report. You kept me on the right track with your intellectual guidance, advices and communications.

I am also grateful to my friends and colleagues for their support, company, sharing of ideas and resources. Finally, I would like to thank all those who facilitated me in coming up with this research.

May God Bless You All.

Contents

TABLE OF FIGURES

Definition of terms

NS2: Network Simulator

WSN: Wireless Sensor Network

LEAP: Localized Encryption and Authentication Protocol

SPINS: Security Protocol for Sensor Network.

SNEP: Sensor Network Encryption Protocol

GIU: Graphical User interface

LISP: Lightweight Security Protocols

DD: Directed Diffusion

GAF: Geographic Adaptive Fidelity

GEAR: Geographic and Energy-Aware Routing

LEACH: Low-energy adaptive clustering hierarchy

CHAPTER 1 INTRODUCTION

1.1 Background to the Study

Wireless Sensor Networks (WSNs) are composed of a large set of nodes with resource constraints. Each sensor nodes has a wireless communication capability plus some level of intelligence for signal processing and data networking. These nodes are usually scattered over the area to be monitored and to collect data, process it, and forward it to a central node for further processing. Wireless Sensor Networks (WSN) are usually characterized by unattended operational environments, ad-hoc style wireless communications and resource-constrained sensor nodes in terms of power, memory and computational capabilities and communication bandwidth (Ruiping Ma, 2012). A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions such a temperature, sound, vibration, pressure and humidity.

Sensor networks are small, low cost, low power devices with the following functionalities; they communicate for short distances, sense environmental data and perform limited data processing e.g. a typical node can have 4MHz of processing power, 4KB of RAM, and short transmission distance of less than 30meters. They communicate using radio frequency, so broadcast is the fundamental communication basic.

The main characteristics of a WSN include power consumption constrains for nodes, node failures, mobility of nodes, communication failures, heterogeneity of nodes, and ability to withstand harsh environmental conditions and ease of use. Sensor nodes can be imagined as small computers. They usually consist of a processing unit with limited computational power and limited memory, a communication device and a power source (Singh, Dua, & Mathur, 2012).

Security is one of the most difficult problem facing these wireless sensor networks, and certain applications of sensor networks, like military application, health care applications, energy monitoring applications, distributed temperature monitoring applications etc. thus security becomes a major concern.

This is because, first Wireless communication is difficult to protect since it is realized over a broadcast medium. In a broadcast medium, enemies can easily eavesdrop on, intercept and alter transmitted data.

Also, since sensor networks may be deployed in a variety of physically insecure environments, an attacker can steal nodes, recover their cryptographic materials, and pose as authorized node in the network.

Lastly, sensor networks are vulnerable to resource consumption attacks. An attacker can repeatedly send packets to drain a node battery and waste network bandwidth. In these and other vital or security sensitive deployment, secure transmission of sensitive digital information over the sensor network is essential.

Security models can be applied to provide security in wireless sensor network s, but keeping in view their resource constraints nature it is very difficult to do so. Researchers are working to develop improved WSN security protocols. Most of the security protocols of sensor networks are symmetric key cryptography based and the protocols are not many.

Security protocols such as SPINS, LEAP and TinySec etc. have been built to provide security requirements such as:

Availability- Ensures that the desired network services are available even in the presence of denial of service attacks.

Authorization-Ensures that only authorized sensors can be involved in providing information to network services.

Authentication-Ensures that the communication from one node to another node is genuine. That is, a malicious node cannot masquerade as a trusted network node.

Confidentiality- Ensures that a given message cannot be understood by anyone other than the desired recipients.

Integrity -Ensures that a message sent from one node to another is not modified by malicious intermediate nodes.

Robustness-When some nodes are compromised the entire network should not be compromised.

Data Freshness-Implies that the data is recent and ensures that no adversary can replay old messages.

Non-repudiation-Denotes that a node cannot deny sending a message it has previously sent.

Self-organization-Nodes should be flexible enough to be self-organizing (autonomous) and self-healing (failure tolerant).

Time Synchronization- These protocols should not be manipulated to produce incorrect data.

To achieve the security requirements, several researchers have focused on evaluating cryptographic algorithms in WSNs and proposing energy efficient ciphers (Xueying Zhang, 2013). Examples of network simulation software are ns2/ns3, OPNET, NetSim. Network simulators are particularly used to design various kinds of networks, simulate and then analyze the effect of various parameters on the network performance

1.2 Statement of the Problem

Wireless Sensor Network (WSN) is an emerging technology that shows great promise for various applications both for mass public and military (Vishal Rathod, 2011). Today, WSN applications can be used in detecting environmental condition, system monitoring, medical system, military and industrial monitoring for ability to transform human life in various aspects. The sensing technology combined with processing power and wireless communication makes it lucrative for being exploited in future. All these applications require a certain level of reliability and security during data transmission. Thus, securing data transmission in WSN is a must.

Depending on applications used for WSNs, security is the biggest challenges in WSNs and security aspect is essential for WSNs before designing WSNs (Princy & Sasikumar, 2015).

Some initiatives have been taken to introduce security mechanism for WSNs but there are not very secure because they are based on different assumptions and according to (Noman, 2008) it is very difficult to establish which security mechanism is suitable for which kind of sensor application.

Identification of most appropriate security protocol and selection of security is a major (Ahmed, 2009). That is most security protocols are not desirable and sometimes infeasible in sensor network (Hassan & Bach, 2014).

Energy is the biggest constraint for a WSN security protocol (Sen, 2013). That is energy consumption in sensor nodes can be categorized in three parts: energy for the sensor transducer, energy for communication among sensor nodes and energy for microprocessor computation. That is sensors are equipped with batteries, but these batteries do have a limited life time, e.g. in underwater scenario, there are no plug-in sockets to provide the power as per the requirement. To save on energy sensor networks should be operated for a longer time without battery replacement. The problem is the lifespan of sensors and maintaining the operation of WSN at a satisfactory level for application. Hence avoiding frequent battery replacement in Wireless Sensor Network which are expensive the researcher evaluated existing security protocol in WSN in terms of their energy consumption

The research have identify the most suitable wireless sensor network protocol based on energy consumption.

1.3 Research Objectives

1.3.1 General objective

The main research objective is to evaluate security protocol in wireless sensor networks based on energy consumption.

1.3.2 Specific Objectives

The main purpose of this research is to

1) To investigate existing WSN security protocols.
2) To establish an energy based approach for evaluating identified WSN security protocols.
3) To evaluate the established approach.

1.4 Significance of the Study

Wireless Sensor Network (WSN) is an emerging technology that shows great promise for various applications both for mass public and military. WSNs are emerging as an area of active research involving hardware and system design, networking, distributed algorithms, programming models, data management, security and social factors. The sensing technology combined with processing power and wireless communication makes it productive for being exploited in in future. All these applications require a certain level of reliability and security during data transmission. Data transmission in wireless Sensor network is important. Since sensors have a limited power, memory and computational resources, any security mechanism for sensor network must be energy efficient and use less memory.

The past few years, Wireless Sensor Networks (WSNs) has demonstrated a greater potential, as integral component of solution, applicable in a variety of domain such as environmental observations, surveillance, military and ambient assisting living. A major technical challenge in wireless sensor network lies in the energy constraints at battery powered node which poses a fundamental limiting factor on the network lifespan.

WSNs applications that are used in hostile environments are being needed for demand of today's world because of many natural disasters like earthquakes, flooding, Tsunamis and forest firing, etc. Currently, WSNs have provided usefulness to several important field areas such as environmental monitoring like flood and forest firing detection, industrial monitoring like status monitoring, medical like Body Sensor Network (BSN), military like reconnaissance of opposing forces and other monitoring systems like air, water and animals. Thus, wireless applications are increasing day by day and it is important to know the best suitable security mechanism to use before implementing the application to achieve the best results in terms of security.

1.5 Scope of the Study.

Wireless Sensor Networks can be used to monitor various hostile environments, and therefore have wide range of applications that uses it. This applications that use WSN can be of sensitive nature and therefore require a boosted secured environment. Since sensors are used to observe some areas that are a bit sensitive therefore energy utilization and security should be a consideration when designing wireless sensor networks. The Sensor nodes get their power from batteries. Since the sensor nodes are sometimes deployed in hostile environment they cannot be recharged. Due to unattended deployment and inability of recharging, the power consumption of the nodes should be ideal.

This research was a simulation model that has analyzed security protocol for wireless sensor networks in term of energy utilization implemented, validated and tested the result.

CHAPTER 2: LITERATURE REVIEW.

This chapter focuses on presenting information on the protocol that are used to secure WSNs

2.1 Introduction

Wireless Sensor Networks (WSNs) are consisting of a large number of low cost, low power, and multifunctional sensor nodes that are deployed in a region of interest (Nour El Din M. Khalifa, 2013). These sensors may have wireless communications and computing capabilities. They are small in size, but are equipped with sensors, embedded microprocessors and radio transceivers. Sensor nodes are scattered in an unattended environment (i.e. sensing field) to sense the physical world. They communicate over a short distance via a wireless medium and collaborate to accomplish a common task, for example, environment monitoring, battlefield surveillance, and industrial process control. Sensed data can be collected by small number sink nodes which have accesses to infrastructure networks like the internet.

The deployment nature of sensor networks made it prone to physical interaction with environment and resource limitations raises some serious questions to secure these nodes against adversaries (K. Kifayat, 2010).

Basically, (Padmavathi & Shanmugapriya, 2009) sensor networks are application dependent. Sensor networks are primarily designed for real-time collection and analysis of low level data in hostile environments

2.1.1 WSN Architecture

Wireless Sensor Network has the following components (Vikash, Anshu & Barwal, 2014):

Gateway: A gateway is an interface between the application platform and the wireless nodes on the wireless sensor network. All information received from the wireless nodes is aggregated/manipulated (e.g. translation between network packet formats) by the gateway and forwarded to the application. That application may run on a local computer or a networked computer. In the reverse direction, when a command is issued by the application program to a wireless node, the gateway relays the information to the wireless sensor network.

All gateways can perform protocol conversion to enable the wireless network to work with other industry or non-standard network protocols.

Relay Node: Each relay node is considered a full-function device (FFD). They are usually called "routers," and they are used to extend network coverage area, route around obstacles and provide back-up routes in case of network congestion or device failure. In some cases, relay nodes may also be connected via analog and digital interfaces to sensors and actuators, providing the same I/O functionality of a leaf node.

Leaf Node: A leaf node is considered as a reduced-function device (RFD). It is sometimes called endpoint. It is designed to provide the physical interface between the wireless sensor network and the sensor or actuator that it is wired to. Leaf nodes are usually equipped with one or more I/O connections for connecting to and communicating with analog or digital sensors or actuator devices.

Sensor/Actuator: This is the device use for interaction with the physical system that you ultimately wish to monitor and/or control. An example is a sensor monitoring the temperature in a room and controlling the air-conditioned equipment.

2.1.2 Security related issues and challenges in wireless sensor networks

Sensor networks pose unique security challenges because of their inherent limitations in communication and computing (Zia & Zomaya, 2013). The deployment nature of sensor networks makes them more vulnerable to various attacks. Sensor networks are deployed in applications where they have physical interactions with the environment, people and other objects making them more vulnerable to security threats. Inherent limitations of sensor networks can be categorized as

Node limitations: A typical sensor node processor is of 4-8 MHz, having 4KBof RAM, 128 KB flash and ideally 916 MHz of radio frequency. Heterogeneous nature of sensor nodes is an additional limitation which prevents one security solution. Due to the deployment nature, sensor nodes would be deployed in environments where they would be highly prone to physical vandalism. **Network limitations:** Beside node limitations, sensor networks bring all the limitations of a mobile ad hoc network where they lack physical infrastructure, and they rely on insecure wireless

Media.

Physical limitations: Sensor networks deployment nature in public and hostile environments in many applications makes them highly vulnerable to capture and vandalism. Physically security of sensor nodes with tamper proof material increases the node cost.

2.1.3 Design challenges in wireless sensor networks and security requirements of WSN

WSNs have many constraints from which new challenges stand out (Chelli, 2015). The extreme resource limitations of sensor nodes and unreliable communication medium in unattended environments make it very difficult to directly employ the existing security approaches on a sensor platform due to the complexity of the algorithms. He listed some of major design challenges as follows:

Very Limited Resources: WSNs pose unique challenges because of the strict resource constraints on each individual sensor. Embedded devices with very limited resource must implement complex, distributed, ad-hoc networking protocols. Size reduction of sensor nodes is essential to cut costs and create more applications. As physical size decreases, so does energy capacity. The underlying energy constraints end up creating computational and storage limitations that lead to a new set of design issues. For example, Zigbex sensor type HBE has an 8-bit, 7.372 MHz ATmega128L RISC MCU with only 4 Kb SRAM, 128 Kb flash memories and 512 Kb flash storage. With such limitations, the software built for the sensor must also be quite small.

Unreliable Communication: Due to the wireless medium that is inherently broadcast in nature, packets may get damaged due to channel errors and conflict will occur, or dropped at highly congested nodes in the network. As well, an attacker can launch Denial-of Service (DoS) attacks without much effort, etc. Furthermore, the multi-hop routing, network congestion and node processing can lead to greater latency in the network, thus making it difficult to achieve synchronization among sensor nodes. The synchronization issues can be critical to sensor security where the security mechanism relies on critical event reports and cryptographic key distribution.

Energy constraints: Energy is one of the major constraint for a WSN. According to Sen, (2013) energy consumption in sensor nodes can be categorized in three parts energy for the sensor transducer, energy for communication among sensor nodes, and energy for microprocessor computation. A study done found that each bit transmitted in WSNs consumes about as much power as executing 800 to 1000 instructions. Thus, communication is costlier than computation in

WSNs. Any message expansion caused by security mechanisms comes at a significant cost. Further, higher security levels in WSNs usually correspond to more energy consumption for cryptographic functions. Thus, WSNs could be divided into different security levels depending on energy cost.

Higher latency in communication: In multi-hop routing, network congestion and processing in the intermediate nodes may lead to higher latency in packet transmission. This makes synchronization very difficult to achieve (Sen, 2013). The synchronization issues may sometimes be very critical in security as some security mechanisms may rely on critical event reports and cryptographic key distribution (Stankovic, Abdelzaher, LU, Sha, & Hou, 2013).

Memory Limitation: A sensor is a tiny device with only a small amount of memory and storage space. According to (Sen, 2013) memory is a sensor node usually includes flash memory and RAM. Flash memory is used for storing downloaded application code and RAM is used for storing application programs, sensor data, and intermediate results of computations. There is usually not enough space to run complicated algorithms after loading the OS and application code. In a SmartDust project, for example, TinyOS consumes about 4K bytes of instructions, leaving only 4500 bytes for running security algorithms and applications.

Unattended Operations: Sensors nodes interact closely with their physical environments, process and fuse data, and eventually create new knowledge that must be presented to an end-user. These tiny nodes are often deployed in open, large-scale and even hostile areas. Potential issues range from accidental node failure to physical capture. Getting secure data in harsh **e**nvironment from physical wireless sensors to an end-user is not a simple task due to these severe constraints.

WSNs have provided usefulness to several important field areas such as because of the useful characteristics such as Power consumption constraints for nodes using batteries, ability to cope node failures, mobility of nodes, dynamic network topology, heterogeneity of nodes, scalability to large scale of deployment, ability to withstand harsh environmental condition and ease of use (Maw, 2014).

According to Maw, (2014) because of the above characteristics, they have limitations and constraints. In WSN, sensor nodes have tiny device, small in volume, limited storage capacity, limited resources, limited processing power consumption and radio ranges, communication bandwidth and storage space

This gives rise to new and unique challenges in data management and information processing such as energy and security. Therefore, in order to develop useful resources efficient mechanisms for WSN, it is necessary to know and understand these constraints first in (Ace Dimitrievski, 2012) and (Muazzam A. Khan, 2013). According to Maw, (2014) energy and security are biggest challenges in WSN.

2.1.3.1 Energy efficiency design challenge

Sensor nodes are likely to be battery powered, and it is often very difficult to change or recharge batteries for these nodes. In fact, someday we expect some nodes to be cheap enough that they are discarded rather than recharged. Prolonging network lifetime for these nodes is a critical issue. Reducing power consumption is clearly an important goal because battery life is not expected to increase significantly in the coming years. In terms of energy consumption, the wireless exchange of data between nodes strongly dominates other node functions such as sensing and processing (Sendra, Lloret, García, & Toledo, 2011)

There are some major sources of energy waste for such communication.

The first one is collision. Usually data in sensor network is transferred by radio therefore two nodes may transfer data to each other at the same time or several nodes transfer data to the same node at the same time. When a transmitted packet is corrupted, it has to be discarded, and the follow-on retransmissions increase energy consumption. Collision increases latency as well.

The second source is overhearing, which occurs when a node picks up packets that are destined to other nodes. In an ad hoc fashion, a transmission from one node to another is potentially overheard by all the neighbors of the transmitting node thus all of these nodes consume power even though the packet transmission was not directed to them.

The third source is control packet overhead. Sending and receiving control packets such as routing update and synchronization consumes energy and effectively reduces the network bandwidth for data packets. The last major source of inefficiency is idle listening, i.e., listening to receive possible traffic that is not sent. This is especially true in many sensor network applications since if nothing is sensed, nodes are in idle mode for most of the time. However, nodes must listen to the channel to receive possible traffic. Many measurements (Savvides, Han, & Srivastava, 2013) have shown that in such networks idle listening consumes 50–100% of the energy required for receiving.

2.1.4 Security requirements of WSN

According to Chawla, (2014) it is necessary to know and understand these security requirements first before implementing security scheme for WSN. WSN should take the following major security requirements which are basic requirements for any network into consideration of secure mechanism.

According to Maw, (2014) important security requirements in WSN are:

2.1.4.1 Data integrity

According to Chawla, (2014) data integrity in sensor networks is needed to ensure the reliability of the data. It ensures that data packets received by destination is exactly the same with transferred by the sender and any one in the middle cannot alter that packet (maw & jaw, 2013). Data integrity is a service that prevents or identifies unauthorized alteration of data during transmission (maw & jaw, 2013). The techniques like message digest and MAC are applied to maintain integrity of the data. By providing data integrity we are able to solve the Data integrity attacks. Data integrity is achieved by means of authentication the data content.

2.1.4.2 Data confidentiality.

Confidentiality is to protect data during communication in a network to be understood other than intended recipient (Chawla, 2014). (Maw, 2014) Defined data confidentiality of the network to mean that data transfer between sender and receiver will be totally secure and no third person can access it (neither read nor write). Cryptography techniques are used to provide confidentiality i.e. symmetric or asymmetric key can be used to protect the data (Ace Dimitrievski, 2006).

2.1.4.3 Data Authentication

According to Padmavathi & Shanmugapriya, (2009) due to the wireless nature of the media and the unattended nature of sensor networks, it is extremely challenging to ensure authentication. Data authentication of a sensor node ensures the receiver that the data has not been modified during the transmission (Prajeet, Niresh, & Rajdeep, March 2012). Asymmetric cryptographic communication digital signatures are used to check the authentication of any message or user while in symmetric key, MAC (Message Authentication Code) are used for authentication purpose (M. A. Khan: G. A. Shah, 2011). Data authentication is achieved through symmetric or asymmetric mechanisms where sending and receiving nodes share secret keys.

2.1.4.4 Data Availability

Data Availability determines whether a node has the ability to use the resources and whether the network is available for the messages to communicate. However, failure of the base station or cluster leader's availability will eventually threaten the entire sensor network (Maw, 2014). Availability ensures that sensor nodes are active in the network to fulfill the functionality of the network. It should be ensured that security mechanisms imposed for data confidentiality and authentication are allowing the authorized nodes to participate in the processing of data or communication when their services are needed (M. A. Khan: G. A. Shah, 2011).

2.1.4.5 Self-Organization

WSN is typically an ad-hoc network, which requires every sensor node be independent and flexible enough to be self-organizing and self-healing according to different situations (Chelli, 2015). There is no fixed infrastructure available for the network management, so nodes must their selves adapt the topology and deployment strategy.

2.1.4.6 Data Freshness

According to Chawla, (2014) data freshness is very important in wireless sensor networks. Because an attacker can send an expire packet to waste the network resources and decrease in network lifetime. Freshness ensures that the data received by the receiver is the recent and fresh data and no adversary can replay the old data. The freshness is achieved by using mechanisms like nonce or timestamp should add to each data packet.

2.1.4.7 Time Synchronization

Many WSN applications demand some form of time synchronization for execution (Chelli, 2015). A more collaborative sensor network may require group synchronization for tracking applications. In order to conserve power, an individual sensor's radio may be turned off for periods of time (Chelli, 2015). Furthermore, sensors may wish to compute the end-to end delay of a packet as it travels between two pairwise sensors. A more collaborative sensor network may require group synchronization for tracking applications, etc.

2.1.4.8 Secure Localization

Sensors may get displaced while deploying them or after a time interval or even after some critical displacement incident. The utility of a WSN will rely on its ability to accurately and automatically locate each sensor in the network (Chelli, 2015).

2.1.5 Factors to consider when dealing with security protocol for WSN application.

Security protocol require a certain amount of resources for the implementation including memory, latency and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor. To be able to select a suitable security mechanism for wireless sensor networks application there are some factors that one need to consider. These factors include:

Energy consumption: The study found that each bit transmitted in WSNs consumes about as much power as executing 800 to 1000 instructions. Thus, communication is expensive in WSNs. Any message expansion caused by security mechanisms comes at a substantial cost. Further, higher security levels in WSNs usually relates to more energy usage for cryptographic functions. Thus, WSNs could be separated into different security levels depending on energy cost. Energy is the biggest challenge to wireless sensor application. The assumptions are that once sensor nodes are deployed in a sensor network, they cannot be easily replaced or recharged. Thus, battery charge taken with them to the field must be well-preserved to extend the life of the sensor node and the entire sensor network. When implementing a cryptographic function or protocol within a sensor node, the effect of energy of the added security code must be carefully considered. When adding security to a sensor node, the issue is the influence that security has on the lifespan of a sensor i.e. its battery life. Thus extra power used up by sensor nodes due to security is related to the processing required for security functions such as during encryption process, decryption process, signing data, verifying signatures, the energy required to transfer the security related data or overhead and the energy required to store security parameters in a secure manner such as cryptographic key storage (Hill, et al., 2000).

Here are some major sources of energy wastage in wireless sensor communication.

Collision is a major source of energy loss. Usually data in sensor network is transferred by radio therefore two nodes may transfer data to each other at the same time or several nodes transfer data to the same node at the same time. When a transmitted packet is corrupted, it has to be rejected, and the follow on retransmissions increase energy utilization. Collision also increases latency.

The second source of energy wastage is overhearing, which occurs when a node picks up packets that are destined to other nodes. In an ad hoc fashion, a transmission from one node to another is potentially overheard by all the neighbors of the transmitting node thus all of these nodes consume power even though the packet transmission was not directed to them.

Control packet overhead also an issue when it comes to energy wastage. Sending and receiving control packets such as routing update and synchronization consumes energy and effectively reduces the network bandwidth for data packets.

The last major source of energy loss is idle listening, that I listening to receive possible traffic that is not sent. This is true in many sensor network applications since if nothing is sensed, nodes are in idle mode for most of the time. However, nodes must listen to the channel to receive possible traffic. Many measurements have shown that in such networks idle listening consumes 50–100% of the energy required for receiving.

Memory and Storage Space. A sensor is a very small device with only a limited amount of memory and storage space for the code. In order to build an efficient security mechanism, it is necessary to limit the code size of the security algorithm e.g. one common sensor type (TelosB) has a 16-bit, 8 MHz RISC CPU with only 10KRAM, 48K program memory, and 1024K flash storage. With such a restraint, the software built for the sensor are quite small. The total code space of TinyOS, the de-facto standard operating system for wireless sensors, is approximately 4K, and the core scheduler occupies only 178 bytes. Thus, code size for the all security related code must also be small.

Latency. Multi-hop routing, network congestion, and node processing can lead to bigger latency in the network, this making it very difficult to achieve synchronization among sensor nodes. The synchronization issues are critical to sensor security and security mechanism relies on critical event reports and cryptographic key distribution.

2.1.5.1 Existing energy reduction methods

Energy aware routing method: This is a reactive method that aims to increase the lifetime of the network. This method seeks to maintain a set of paths instead of maintaining or enforcing one optimal path at higher rates, although the behavior of this protocol is similar to directed diffusion protocols (Sendra, Lloret, García, & Toledo, 2011). These routes are selected and maintained by a probability factor. The value of this probability depends on the lowest level of energy achieved in each path. The method assumes that each node is addressable through a class-based addressing scheme which includes the location and the type of nodes.

Hybrid Energy-Efficient Distributed clustering (HEED): According Sendra, Loret, García, & Toledo, (2011) they proposed a method of saving energy for clusters of nodes in WSNs. HEED (Hybrid Energy Efficient Distributed clustering) periodically selects the main nodes in the cluster according to a set of parameters such as residual energy and a secondary endpoint. It also seeks to extend the network lifetime by distributing energy consumption. In HEED the system does not take care of the type of technology used.

2.2 Protocol Class

In WSN, the main task of a sensor node is to sense data and sends it to the base station in multi hop environment for which routing path is essential (Matin & Islam, 2012). For computing the routing path from the source node to the base station there is huge numbers of proposed routing protocols exist (Sharma et al., 2011). The design of routing protocols for WSNs must consider the power and resource limitations of the network nodes, the time-varying quality of the wireless channel, and the possibility for packet loss and delay.

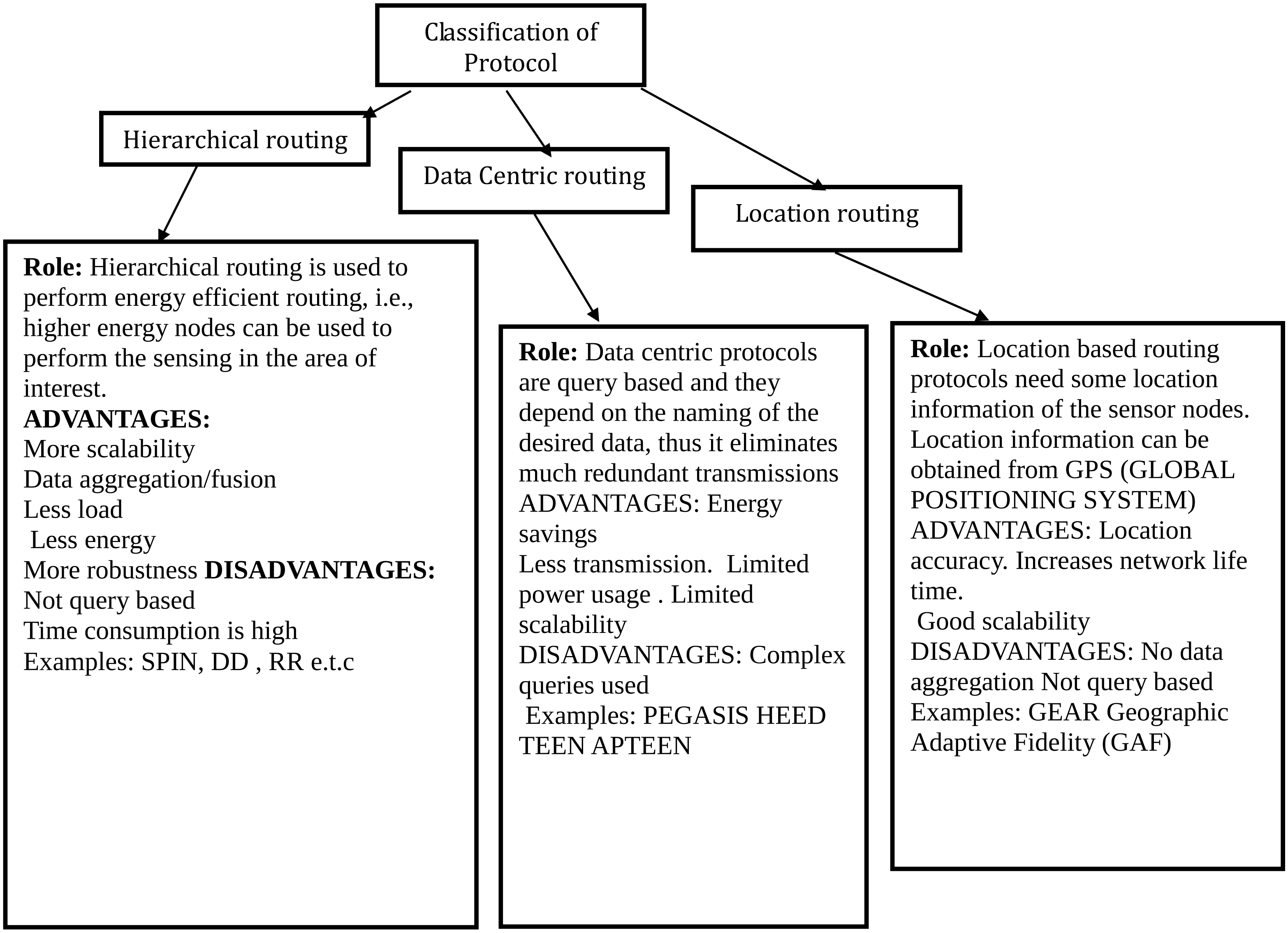
To address these design requirements, several routing strategies for WSNs have been proposed in (Akkaya et al., 2005), and (Waharte et al., 2006)and (Labrador et al., 2009).

The f**irst class of routing protocols adopts a flat network architecture** in which all nodes are considered peers. Flat network architecture has several advantages, including minimal overhead to maintain the infrastructure and the potential for the discovery of multiple routes between communicating nodes for fault tolerance.

A second class of routing protocols imposes a structure on the network to achieve energy efficiency, stability, and scalability. In this class of protocols, network nodes are organized in clusters in which a node with higher residual energy, for example, assumes the role of a cluster head. The cluster head is responsible for coordinating activities within the cluster and forwarding information between clusters. Clustering has potential to reduce energy consumption and extend the lifetime of the network. Example of protocols in this class are LISP and LEAP.

A third class of routing protocols uses a data-centric approach to disseminate interest within the network. The approach uses attribute-based naming, whereby a source node queries an attribute for the phenomenon rather than an individual sensor node. The interest Wireless Sensor Networks Technology and Protocols dissemination is achieved by assigning tasks to sensor nodes and expressing queries to relative to specific attributes. Different strategies can be used to communicate interests to the sensor nodes, including broadcasting, attribute-based multicasting, geo-casting, and any casting.

A fourth class of routing protocols uses location to address a sensor node. Location-based routing is useful in applications where the position of the node within the geographical coverage of the network is relevant to the query issued by the source node. Such a query may specify a specific area where a phenomenon of interest may occur or the vicinity to a specific point in the network environment.



Role: Hierarchical routing is used to perform energy efficient routing, i.e., higher energy nodes can be used to perform the sensing in the area of interest.
ADVANTAGES:
More scalability
Data aggregation/fusion
Less load
Less energy
More robustness **DISADVANTAGES:**
Not query based
Time consumption is high
Examples: SPIN, DD , RR e.t.c

Role: Data centric protocols are query based and they depend on the naming of the desired data, thus it eliminates much redundant transmissions
ADVANTAGES: Energy savings
Less transmission. Limited power usage . Limited scalability
DISADVANTAGES: Complex queries used
Examples: PEGASIS HEED TEEN APTEEN

Role: Location based routing protocols need some location information of the sensor nodes. Location information can be obtained from GPS (GLOBAL POSITIONING SYSTEM)
ADVANTAGES: Location accuracy. Increases network life time.
Good scalability
DISADVANTAGES: No data aggregation Not query based
Examples: GEAR Geographic Adaptive Fidelity (GAF)

Figure 1: Classification and Comparison of protocols in WSN

2.2.1 Security protocol in WSN

There are a number of wireless Sensor Network Protocols today, (Sharma, Chaba, & Singh, 2010) most of them require intensive computation and memory which is the limiting factor in wireless sensor networks. Some these protocols are discussed below:

2.2.1.1 SPIN (Security Protocol for Sensor Network.)

Adrian Perrig et al proposed “SPINS” a suite of security protocols optimized for sensor networks. SPINS has two secure building blocks: SNEP and µTESLA. SNEP includes: data confidentiality, two-party data authentication, and evidence of data freshness. µTESLA provides authenticated broadcast for severely resource-constrained environments (Perrig A. , Szewczyk, Wen, Culler, & Tygar, 2001).

According to Sharma, Chaba, & Singh, (2010) SNEP provides low communication overhead as it only adds 8 bytes per message, achieves semantic security, which prevents eavesdroppers from inferring the message content from the encrypted message and also offers data authentication, replay protection, and weak message freshness.

µTesla is asymmetric digital signatures are impractical for sensor networks for the authentication, as they require long signatures with high communication overhead of 50-1000.

2.1.1.2 TINYSEC

It a Link ayer security protocol for WSN (Karlof, Sastry, & Wagner, 2004) Tinysec provides authentication, message integrity, and confidentiality and replay protection. A major difference between TinySec and SNEP is that there are no counters used in TinySec. TinySec supports two different security options: authenticated encryption (TinySec-AE) and authentication only (TinySec-Auth). With authenticated encryption, TinySec encrypts the data payload and authenticates the packet with a MAC. The MAC is computed over the encrypted data and the packet header. In authentication only mode, TinySec authenticates the entire packet with a MAC, but the data payload is not encrypted.

2.2.1.3 LEAP (Localized Encryption and Authentication Protocol)

LEAP provides key management protocol for sensor networks (Zhu, Setia, & Jajodia, 2003). LEAP is designed to support secure communications in sensor networks; therefore, it provides the basic security services such as confidentiality and authentication.

2.2.1.4 LEACH (Low-Energy Adaptive. Clustering Hierarchy)

It is basically cluster based protocol. It is based on two phases: a setup phase and a steady phase (Gill1, Chawla, & Sachdeva, 2014). A setup phase is responsible for cluster creation in the network and chooses the cluster in the network. Each node decides to become a cluster heads randomly. Cluster head chooses the data to be used in its cluster. In the steady phase the node in the cluster sense and forward data to its cluster head. Cluster head gather all the data send by the node, start compress and aggregate it and send back to sink. LEACH assumes that all cluster head can directly communicate with the sink of the network. Therefore in the network having large regions it is not applicable. Nodes can sleep when there is not their turn to transmit. Cluster heads are rotated randomly. It transmits only new data to the sink.

Its advantages include that it is distributed with no global knowledge is required and we can save energy due to aggregation in the cluster head.

Its disadvantages includes that this protocol assumes that each node have enough power to transmit it to the cluster head and cluster head have enough power to transmit it to sink.

2.2.1.5 GEAR (Geographic and Energy-Aware Routing)

It is a Location based routing protocols for sensor network need location information of all the sensor nodes to calculate the distance between any two nodes (Gill1, Chawla, & Sachdeva, 2014). GEAR is a location based routing protocol which uses GIS (Geographical Information System) to find the location of sensor nodes in the network. According to this protocol, each node stores two types of cost of reaching the destination: estimated cost and learning cost. The estimated cost is a combination of residual energy and distance to destination. The learned cost is a modified estimated cost and it accounts the routing around holes in the network. When a node does not have any closure neighbours towards the target region, a hole occurs. In case where no holes exit, the estimated cost is equal to the learned cost. The GEAR protocol only considers a certain region rather than sending the interests to the whole network as happens in Directed Diffusion and thus restricting the number of interests (Gill1, Chawla, & Sachdeva, 2014).

2.2.1.6 GAF :(Geographic Adaptive Fidelity)

According to Gill1, Chawla, & Sachdeva, (2014) GAF is an energy efficient location-based routing protocol. This protocol was initially conceived for mobile ad hoc networks, but it can also be applied to sensor networks. GAF can be implemented both for non-mobile and mobile nodes. Although GAF is a location based protocol, it may also be implemented as a hierarchical protocol where the clusters are based on geographic location. Initially the area of interest is split into some fixed zones forming a virtual grid for the covered area. Nodes in each zone have different functionalities and each node uses its GPS indicated location to associate itself with a point in the grid. Nodes which are positioned at the same point on the grid are considered equivalent in terms of the cost of packet routing. Such equivalence is exploited in keeping some nodes located in a particular grid area in a sleeping state in order to save energy. Thus GAF can increase the network lifetime as the number of nodes increases. GAF conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity. GAF defines three states:

discovery, active, sleep. The ‘discovery’ state is used for determining the neighbors in the grid; the ‘active’ state participates in routing process and at the time of ‘sleep’ state, the radio is turned off. In order to handle the mobility, each node in the grid estimates it’s leaving time of grid and sends this to its neighbors. The sleeping neighbors adjust their sleeping time accordingly in order to keep the routing fidelity. Before the leaving time of the active node expires, sleeping nodes wake up and one of them becomes active.

2.2.1.7 Limitation of security protocols

The existing security protocols have the following limitations in the security aspects like overload in communication, low computational power, high Resource consumption, lack of integrity and lack of confidentiality (Ramesh, Priya, & B.Santh, 2012).

2.3 Existing comparison methods of WSN. Security protocols

According to various researchers there are several methods that are used to compare and evaluate various security protocols. According to Wang, Attebury, & Ramamurthy, (2012) the came up with methods of evaluating security protocols in WSN. They compared SNEP, LEAP, and µTESLA interms of their confidentiality, authentication, integrity and scalability the methods are illustrated in the below table:

Protocols	**Categories**	**Confidentiality**	**P2P authentication**	**Broadcast Authentication**	**integrity**	**scalability**
SNEP	Flat	yes	yes	no	yes	Good
LEAP	Hierarchy	Yes	yes	yes	yes	medium
µTESLA	Flat/Hierarchy	no	no	yes	yes	good

Figure 2: comparison of WSN security protocols according to (Wang, Attebury, & Ramamurthy, 2012)

They evaluated Sleach, Leach AND Dsdv security protocols in terms of performance using three metrics packet delay, end to end delay and packet loss. From their study they discovered that Sleach is better in terms of performance than the other two protocols (Ouafaa, Mustapha, & Krit Salah-Ddine, 2016).

According to Meena & Talwa, (2015) they compared Flat based routing protocols that is SPIN and DD based on throughput, packet loss and end to end delay. Their study found out that SPIN cannot grantee data delivery and DD overcomes the problem of SPIN. That is SPIN protocol is not scalable because if the sink is not interested in too many events, this could make the sensor nodes around it reduce their energy. But DD is more scalable than SPIN

According to sha, Gehlot, Robert, (2012) single routing is simple and scalable. It is simple because the route between the source and destination node can be established in a short period of time, and scalable because even if the network changes the complexity and the approach to discover the path remains the same. In single routing failures are common because of insufficient power, storage space e.t.c.

2.4 Simulation tools

Network simulators are used to simulate the behavior of networks. Simulation is a process in which an entire system is made functional in a hyper theoretical manner with the help of simulation tool.

Simulation tools for wireless sensor networks are increasingly being used to study sensor webs and to test new applications and protocols in this evolving research field (Mishra, Mishra, Kayal, & Chudi, 2012).

Simulation result depends upon on the environment and physical layer assumption which may not be accurate to predict the real behavior of wireless sensor network. Simulation is necessary to test the application and protocols in this field (Chhimwal, Rai, & Rawat, 2013). The correctness of the model and Suitability of model for the implementation are necessary factors of WSN simulations

The key properties of good Simulator reusability and availability, performance and scalability, support for rich-semantics scripting languages to define experiments and process results and Graphical, debug and trace support (Chhimwal, Rai, & Rawat, 2013).

There are several tools used for simulating wireless networks they include:

2.4.1 NS-2

NS-2 is the abbreviation of Network simulator version two, which first been developed by 1989 using as the REAL network simulator (Sinha, Chaczko, & Klempous, 2009). Now, NS-2 is supported by Defense Advanced Research Projects Agency and National Science Foundation. NS-

2 is a discrete event network simulator built in Object- Oriented extension of Tool Command Language and C++. People can run NS-2 simulator on Linux Operating Systems or on Cygwin, which is a Unix-like environment and command-line interface running on Windows. NS-2 is a popular non-specific network simulator can used in both wire and wireless area. This simulator is open source and provides online document.

Language:-

Object- Oriented extension of Tool Command Language and C++

Key feature:-

NS-2 extensibility features.

Object oriented design allow creating and using of new protocol.

It provide visualization tool-NAM (Network Animator)

Limitation:-

However, this simulator has some limitations (Sinha, Chaczko, & Klempous, 2009). People who want to use this simulator needs to be familiar with writing scripting language and modeling technique; the Tool Command Language is somewhat difficult to understand and write.

Sometimes using NS-2 is more complex and time-consuming than other simulators to model a desired job.

NS-2 provides a poor graphical support, no Graphical User Interface (GUI); the users have to directly face to text commands of the electronic devices.

Due to the continuing changing the code base, the result may not be consistent, or contains bugs

2.4.2 TOSSIM

It is discrete event simulator for TinyOS Wireless Sensor Network, which is open source operating system targeting embedded operating system (Sinha, Chaczko, & Klempous, 2009). It was first developed at UC Berkeley. TOSSIM is a bit-level discrete event network emulator built in Python, a high-level programming language emphasizing code readability, and

C++. People can run TOSSIM on Linux Operating Systems or on Cygwin on Windows. TOSSIM also provides open sources and online documents.

Environment:

It runs on custom mote hardware. It chooses the accuracy and complexity of model necessary for their simulation.

Language:- Python, NesC, C++

Key Features:

It provides interaction with the networks due to its graphical support. Packet can be dynamically injected into the network. Packet traffic can be easily monitored in his way.

Advantages:

Open Source and online documentation

Graphical User Support (Tiny ViZ).

Simple and powerful emulator for Wireless sensor Network.

Support thousands of Nodes.

Limitation:

It is specially designed for tinyOS, not designed for simulation performance metrics of other new protocol (Sinha, Chaczko, & Klempous, 2009).Therefore, TOSSIM cannot correctly simulate issues of the energy consumption in WSN; people can use Power TOSSIM, another TinyOS simulator extending the power model to TOSSIM, to estimate the power consumption of each node. Secondly, every node has to run on NesC code, a programming language that is event-driven, component-based and implemented on TinyOS, thus TOSSIM can only emulate the type of homogeneous applications. Thirdly, because TOSSIM is specifically designed for WSN simulation, motes-like nodes are the only thing that TOSSIM can simulate. In sum, TOSSIM as an emulator of WSN contains both advantages and disadvantages.

2.4.3 OMNeT++

Modular discrete event simulator implemented in C++. Getting started with it is quite simple, due to its clean design. OMNET++ also provides a powerful GUI library for animation and tracing and debugging support. Its major drawback is the lack of available protocols in its library, compared to other simulators (Sinha, Chaczko, & Klempous, 2009).

Advantages:

Powerful Graphical User Interface (making tracing and bugging easier)

Simulate power Consumption problem

Limitation:

Number of protocol is not large enough. Compatibility problem (not portable)

2.4.4 J-Sim

A component-based simulation environment developed entirely in Java. It provides real-time process based simulation. The main benefit of J-Sim is its considerable list of supported protocols, including a WSN simulation framework with a very detailed model of WSNs, and a implementation of localization, routing and data diffusion WSN algorithms (Sinha, Chaczko, & Klempous, 2009). J-Sim models are easily reusable and interchangeable offering the maximum flexibility. Additionally, it provides a GUI library for animation, tracing and debugging support.

Advantages: Models in J-Sim have good reusability and interchangeability, which facilities easily simulation (Sinha, Chaczko, & Klempous, 2009).

J-Sim contains large number of protocols; this simulator can also support data diffusions, routings and localization simulations in WSNs by detail models in the protocols of J-Sim. J-Sim can simulate radio channels and power consumptions in WSNs.

J-Sim provides a GUI library, which can help users to trace and debug programs. The independent platform is easy for users to choose specific components to solve the individual problem. Fourth, comparing with NS-2, J-Sim can simulate larger number of sensor nodes, around 500, and J-Sim can save lots of memory sizes.

Limitation:

J-Sim is relatively complicated to use.

The execution time is much longer than that of NS-2.

Other simulation tool includes Castalia, QualNet and NS-3

2.5 Attacks on Sensor Networks

Wireless Sensor networks are vulnerable to security attacks due to the broadcast nature of the transmission medium (Vikash, Anshu , & Barwal, 2014). Furthermore, wireless sensor networks have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where they are not physically protected (Padmavathi & Shanmugapriya, 2009). According to Chelli, (2015) attackers in WSNs can be categorized as illustrate in Fig 1, based on the following characteristics: goals, performer, and layer wise.

2.5.1 Goal oriented attacks

(CHELLI, 2015) There are two categories of goal oriented attacks they include passive and active attacks

2.5.1.1 Passive attacks

Passive attacks is the monitoring and listening of the communication channel by unauthorized attackers. These attacks are mainly against data confidentiality. An attacker monitors unencrypted traffic and looks for sensitive information that can be used in other types of attacks (Padmavathi & Shanmugapriya, 2009).

According to Chelli, (2015) passive attacks include traffic analysis, monitoring communications, decrypting weakly encrypted traffic, and capturing authentication information. Passive interception of network operations enables adversaries to see upcoming actions. Such attacks result in the disclosure of information or data files to an attacker without the consent or knowledge of the use. Some of passive attacks in WSN are attacks against privacy e.g. monitoring and eavesdropping, traffic analysis and camouflage adversaries.

2.5.1.2 Active Attacks

According to Chelli, (2015) in active attacks, the attacker is no longer passive but takes active measures to achieve control over the network. That is an unauthorized attackers monitors, listens to and modifies the data stream in the communication channel are known as active attack

(Padmavathi & Shanmugapriya, 2009).Some examples of active attacks are DoS, modification of data, black hole, replay, sinkhole, spoofing, flooding, jamming, overwhelm, wormhole, fabrication, Hello flood, node subversion, lack of cooperation, modification, node subversion, man-in-middle attack, selective forwarding and false node.

2.5.1.2 Performer-Oriented Attacks

According to Chelli, (2015) another category in attacks on WSNs can be either outside or inside attack.

Outside attacks: Outside attacks may cause passive eavesdropping on data transmissions, as well as can extend to injecting bogus data into the network to consume network resources and raise Denial of Service attacks.

Inside attacks: This can damage the network stealthily since they can avoid our authentication and authorization because they are legitimate nodes of the native network and have access to the network information, and it is not easy to expect their attack patterns. Inside attackers can launch various types of attacks, such as modification, misrouting, eavesdropping or packet drop. This last attack is tricky to counter, because for a particular packet drop, we cannot distinguish whether it is dropped by an attacker or a result from collision or noise. This attack suppresses the important information reaching the base station which significantly degrades network performance, such as packet delivery rate due to their repeated packet drops.

There are several types of packet drop attacks such as blackhole, grayhole and on-off attacks. This is a serious threat for many applications, such as military surveillance system that monitors the battlefield and other critical infrastructures.

2.5.1.3 Layer-Oriented Attacks

According to Chelli, (2015) WSNs are organized in layered form. This layered architecture makes these networks vulnerable to various kinds of attacks.

Physical Layer Attacks: Physical attacks on WSNs range from node capturing to the jamming of the radio channel. Physical attacks on WSNs availability are even more difficult to prevent than software attacks, because of the lack of physical control over the individual nodes. Jamming is one of the most important attacks at physical layer, aiming at interfering with normal operations. An attacker may continuously transmit radio signals on a wireless channel.

An attacker can send high-energy signals in order to effectively block wireless medium and to prevent sensor nodes from communicating. This can lead to Denial-of-Service attacks at this layer.

Data Link Layer Attacks: The functionality of link layer protocols is to coordinate neighboring nodes to access shared wireless channels and to provide link abstraction to upper layers. Attackers can deliberately violate predefined protocol behaviors at link layer. For example, attackers may induce collisions by disrupting a packet, cause drain of sensor node energy by repeated retransmissions, or intercepting and examining messages in order to deduce information from patterns in communication. This can be performed even when the messages are encrypted and cannot be decrypted, or even cause unfairness by abusing a cooperative MAC layer priority scheme.

Network Layer Attacks: The network layer of WSNs is vulnerable to the different types of attacks, such as DoS attacks that are aimed at complete disruption of routing information, and therefore the whole operation of ad-hoc network. A sinkhole attack tries to lure almost all the traffic toward the compromised node, creating a metaphorical sinkhole with the adversary at the centre. Also if an attacker captures a single node, it is sufficient for him to get hold of the entire network.

Malicious or attacking nodes can however refuse to route certain messages and drop them. Spoofed, Altered, or Replayed Routing Information are the most direct attacks against a routing protocol in any network, are to target the routing information itself while it is being exchanged between nodes. An attacker may spoof, alter, or replay routing information in order to disrupt traffic in the network.

Transport Layer Attacks: An attacker may repeatedly make new connection request until the resources required by each connection are exhausted, or reach a maximum limit. It produces severe resource constraints for legitimate nodes.

Application Layer Attacks: Different type of attacks can be carried out in this layer, such as overwhelm, repudiation, data corruption and malicious code. In overwhelm attack, an attacker might overwhelm network nodes, causing network to forward large volumes of traffic to a base station. This attack consumes network bandwidth and drains nodes energy.

2.6 Conceptual model

The sensor nodes main objective is to make measurements about an occurrence surrounding the sensors, and form a wireless network by communicating over a wireless medium and collect and route data to the Base station (sink). Sensor nodes form a WSN due to the fact that they are scattered over a given area and security protocols are deployed to the sensor nodes. Figure 3 below describes the conceptual framework that is used to explain how the model implemented for simulation of security mechanisms. The function of sensor protocol stack is to detect and process data detected and forward it to application layer. The function of the application layer is to process and transmit data to the user node in the form of sensor reports through the wireless channel. Power mode consisting of battery, CPU and Radio is key part of the sensor node.

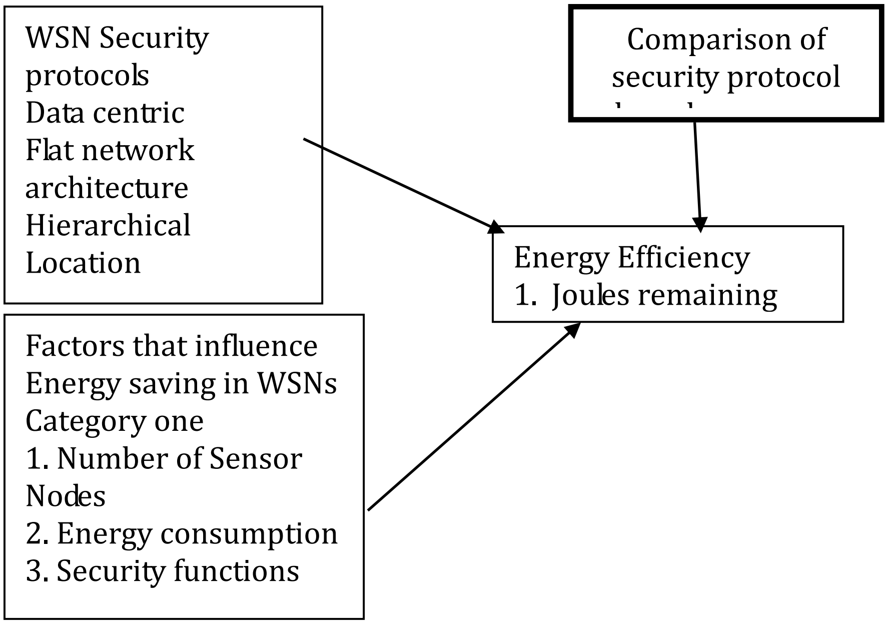


Figure 3: conceptual Model

2.6.1 Operationalization of variables

Variable	**subvariables**	**Indicators/properties**	**Values**
WSN Security protocols	Data centric protocol	Uses attribute-based naming	SPIN ,LISP,DD,TINYSEC,LEACH
	Flat Network architecture	nodes are considered peers	SPIN, DD
	Hierarchical	network nodes are organized in clusters	LEACH, GAP
	Location	position of the node within the geographical coverage of the network	GEAR, GAF SPAN
Factors that affect Energy efficiency	All classes	Number of Sensor Nodes Security functions Sources of energy wastage	Node count No of security functions Collisions Overhearing Idle listening
Energy Efficiency		rate of energy consumption	Remaining Amount of energy (Jouls)

Table 1: Operationalization of variables

2.7 Conclusion

Security is becoming a major concern for energy constrained wireless sensor network because of the broad security-critical applications of WSNs. Thus, security in WSNs has attracted a lot of attention in the recent years. The salient features of WSNs make it very challenging to design strong security protocols while still maintaining low overheads. From the literature review it is noted that energy is a major concern in WSN. Some initiatives have been taken to introduce security protocol for WSNs but there are not very secure because they are based on different assumptions and according to (Noman, 2014) it is very difficult to establish which security mechanism is suitable for which kind of sensor application.

CHAPTER 3 RESEARCH METHODOLOGY

3.1 Introduction

Methodology is a collection of methods, techniques, tools and documentation which help the researcher to realize his research objective. This chapter covers the methods which will be employed to structure the research process in gathering and analyzing information to address the research objectives.

3.2 Evaluation of the current methods

There are three methods mostly used in evaluation of wireless sensor networks (WSN), namely

i. Analytical methods
ii. Direct Application/physical measurement
iii. Simulations

Analytical/Theoretical methods-A security protocol can be mathematically modeled and parameterized. Performance or energy cost can then be expressed in terms of security parameter and wireless Sensor Network (WSNs) characteristics. Each security mechanism may have a parameter that can significantly affect WSNSs resource consumption. Analytical solutions typically offer less accuracy than simulation, but are also less costly and time consuming.

Direct Application/ Physical measurement –this is the actual construction of WSNs and implementing a given security mechanism. With this approach, the experimental results would be more likely to closely match real world situations. Unfortunately, implementing security protocols for large scale networks maybe expensive and time consuming.

Simulation- Performing a simulation is more economically advantageous than actually implementing a design and testing it. The iterative process of designing, implementing, and analyzing can increase expenses for a project. Simulations can use the model created in the design phase for multiple experiments and analysis. In this method you can implement different security protocol, and can be done using software. A simulation can also provide results that are not experimentally measurable or would require many actual experiments

Simulation can use models created in the design phase for multiple experiments and analysis. However, a disadvantage in simulating is that real systems are complex to model, and in addition there could be a possibility of simulation errors and programming the simulations using theories and algorithms. Not everything may be accounted for in terms of an actual deployment.

3.3 ADOPTED METHODOLOGY

The research adopted design science methodology. This methodology has various stages that were followed by the researcher during his study. This activities are illustrated in the table 3 below:

Research activities	**Research techniques**	**Research objectives**
Problem awareness	Online desk research Purposive sampling	To investigate existing WSN security protocols.
Suggestion	Making contrast/comparisons	To establish the suitable method for evaluating identified WSN security protocols.
Evaluation	-Simulation -Experimental design	To evaluate the established method.
Conclusion	-Inductive approach	All objectives

Table 2Adopted Research techniques

3.3.1 Problem awareness

To investigate and identify the existing protocols in WSN the researcher used online desk research and purposive sampling. Online Desk research deals with gathering and analyzing information on print media and internet. According to Noreh, (2009) E-resources are convenient to use and make research a lot easier in that, they enable one to search for information at a faster rote because search engines ore utilized as opposed to manual searches. Academic journal on the internet are credible, clear and are broader in perspective.

Purposive sampling relies on the judgment of the researcher when it comes to selecting the units (e.g., people, cases/organizations, events, pieces of data) that are to be studied. According to Palys, (2008) the researcher sample must be tied to his objectives. Several research journals, papers from the internet that related to the research that deal with WSN security protocols were sampled. When sampling the researcher used journals that are recently published.

3.3.2 Suggestion

To a suitable energy efficiency security protocol in WSN the researcher uses comparing and contrast concept. The researcher identified various method that can be used to compare several security protocol in WSN. That is the researcher identified several journals using online desk research that explained several methods of comparing and evaluating security protocols. The researcher identified several ways of comparing WSN protocols from the journal he sampled and identified that some methods have not been evaluated.

3.3.Evaluation

The research uses simulation methodology. This method has been used to simulate various experiments. Network simulation methodology is often used to verify analytical models, generalize the measurement results, and evaluate the performance of protocols that have being developed (Sarkar, Syafnidar, & I., 2011). This technique has become popular among computer and telecommunication network researchers and developers worldwide (Sarkar, Syafnidar, & I , 2011). This popularity is due to the availability of various sophisticated and powerful simulation packages, and also because of the flexibility in model construction and validation offered by simulation. For selecting an appropriate network simulator for a simulation task, it is important to have good knowledge of the simulation tools available, along with their strengths and weaknesses. It is also important to ensure that the results generated by the simulators are valid and credible.

The selection of a particular evaluation technique can significantly impact the outcome. This methods differs in terms of cost and required time. In consideration of these factors, simulation was most appropriate technique. I ruled out direct application because the technique is based upon both cost and required time. Analytical solutions typically offers less accuracy than simulation but are less costly and time consuming. The cost in simulation is a bit cheaper because simulation software’s are less costly.

3.4 Conclusion

The research has investigate and identify existing WSN security protocols establish the suitable method for evaluating identified WSN security protocols demonstrated the implementation of the established WSN security protocol, validate and test various protocols based on energy consumption.

The research was simulated using NS-2 because it is an open source, discrete-event network simulator that provides support for a simulation of main protocols, routing, multicast protocols for wired and wireless networks. The simulation environment of NS-2 can run on a number of operating systems, i.e., Linux, Windows, OSX, Solaris, etc. The standard ns-2 distribution runs on Linux. However, a package for running ns-2 on cygwin (Linux emulation for windows) is available. In this mode, ns-2 runs in the windows environment on top of Cygwin.

CHAPTER 4: PRESENTATION OF FINDING

4.1 Introduction

This chapter describes the process of defining the architecture module, interface and data for the system to satisfy specified requirements. It also describes the conceptual model that was implemented for simulation and hardware and software environment used for simulation purposes.

4.2 Existing protocols in WSN

Research objective one was to investigate and identify existing WSN security protocols. The researcher found that there are several existing security protocols in WSN. The table 3 below show various protocols that the researcher identified and investigated.

Classes	**Existing protocols**	**Challenges**	**Sources**
Data centric	SPIN	• The dissemination of data in the network through SPIN protocol takes long time. • A node with much more computation consumes more energy • Few sensor nodes may be used several times and those nodes may lose energy early then other nodes in the network. • It is not sure about the data will certainly reach the target or not and it is also not good for high-density distribution of nodes.	(Mohammed & R, 2013) (Rathi, Saraswat, & Bhattacharya, 2012)
	LLSP	• Maintaining a large network is difficult with in node counter due to that it has low scalability. • Can't assure data availability	(Ndia, 2017)

	DD	• It not a good choice for the application such as environmental monitoring because it require continuous data delivery to the sink will not work efficiently with a query driven on demand data model.	(Rathi, Saraswat, & Bhattacharya, 2012)
	TINYSEC	• Cannot guard against resource consumption on attacks, node capture and replay attacks	(Ndia, 2017)
	LEACH	• It does not give any idea about the number of cluster heads in the network. • When Cluster head dies, the cluster will become useless because the data gathered by the cluster nodes would never reach its destination i.e. Base • Clusters are divided randomly, which results in uneven distribution of Clusters. LEACH does not give any idea about the number of cluster heads in the network. • When Cluster head dies, the cluster will become useless because the data gathered by the cluster nodes would never reach its destination i.e. Base Station. • Clusters are divided randomly, which results in uneven distribution of Clusters. This phenomenon can cause an increase in energy consumption and have great impact on the performance of the entire network.	(Gill1, Chawla, & Sachdeva, 2014)
	GAF	• It conserves energy by turning off unnecessary nodes in the network	(Dr.R.KalaiMagal & Revathy, 2014)

Flat Network Architecture	SPIN	• The dissemination of data in the network through SPIN protocol takes long time. • A node with much more computation consumes more energy • Few sensor nodes may be used several times and those nodes may lose energy early then other nodes in the network. • It is not sure about the data will certainly reach the target or not and it is also not good for high-density distribution of nodes.	(Mohammed & R, 2013) (Rathi, Saraswat, & Bhattacharya, 2012)
	DD	• It not a good choice for the application such as environmental monitoring because it require continuous data delivery to the sink will not work efficiently with a query driven on demand data model.	(Rathi, Saraswat, & Bhattacharya, 2012)
Hierarchical architecture	LEACH	• It does not give any idea about the number of cluster heads in the network. • When Cluster head dies, the cluster will become useless because the data gathered by the cluster nodes would never reach its destination i.e. Base Clusters are divided randomly, which results in uneven distribution of Clusters. LEACH does not give any idea about the number of cluster heads in the network. • When Cluster head dies, the cluster will become useless because the data gathered by the cluster	(Gill1, Chawla, & Sachdeva, 2014)
	GAF	• It conserves energy by turning off unnecessary nodes in the network	(Dr.R.KalaiMagal & Revathy, 2014)
	LISP	• It requires an Intrusion Detection System (IDS) for better security	(Ndia, 2017) (Dr.R.KalaiMagal & Revathy, 2014)

	LEAP	• Assumes that sink node is never compromised	(Ndia, 2017) (Dr.R.KalaiMagal & Revathy, 2014)
Location	SPAN GEAR	• It is not scalable and does not support data diffusion.	(Dr.R.KalaiMagal & Revathy, 2014)
	GAF	• It conserves energy by turning off unnecessary nodes in the network	(Dr.R.KalaiMagal & Revathy, 2014)

Table 3: Existing protocols in WSN

The research found out that there are several categories of WSN protocols e.g. Location, Hierarchical architecture, Flat Network Architecture, and data centric. We sampled several protocols from different categories e.g LISP, LEAP, Tinysec, SPIN, and LLSP and looked at their challenges in terms of how the consume energy. For example in SPIN, node with much more computation consumes more energy and if few sensor nodes are used several times and those nodes may lose energy early then other nodes in the network. In LEACH When Cluster head dies, the cluster will become useless because the data gathered by the cluster nodes would never reach its destination i.e. Base Clusters are divided randomly, which results in uneven distribution of Clusters. LEACH does not give any idea about the number of cluster heads in the network.

4.3 Method of comparing security protocols in WSN

Research objective two was to establish the suitable method for evaluating identified WSN security protocols. The researcher found that there are several methods that other researcher have used to compare various security protocols. These methods are discussed below:

Protocols are compared in terms of performance using three metrics which are packet delivery function, packet loss, and throughput. Packet delivery function is the function of all the packets that have been deliver successfully while end to end delay are the possible delays caused buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times and throughput measures how well the network can constantly provide data to the sink, that is through put is how many data packets received by receiver with in data transmission time or successful data transmission performed within a time period.

The research also identified that protocols can also be compared based on data confidentiality, integrity and authenticity as in the table below:

Protocols	**Categories**	**Confidentiality**	**P2P authentication**	**Broadcast Authentication**	**integrity**	**scalability**
SNEP	Flat	yes	yes	No	yes	Good
LEAP	Hierarchy	Yes	yes	Yes	yes	medium
µTESLA	Flat/Hierarchy	no	no	Yes	yes	good

Table 4: comparison of WSN security protocols according to (Wang, Attebury, & Ramamurthy, 2012)

The researcher found other ways of evaluating security protocols. These method are Energy consumption, memory and storage space and latency.

From the above the researcher found out that energy is an essential factor in WSN hence the researcher decided to compare security protocols based on the energy consumption to identify the most suitable protocol in terms of energy efficiency.

4.3.1 Energy efficient pseudo code

From the study it was found out that there are techniques that are simple scalable and reduce the cost of operation in WSN. According to sha, Gehlot & Robert, (2012) one of these technique is were packets are forwarded using a single path (single routing). This method improves simple, scalable but it does not efficiently satisfy the requirement of resource constraint in the network e.g. energy. The issues with this technique, nodes on the network that are not sending or receiving packets will be in idle state and they will be consuming energy.

From the above shortcoming the researcher developed a pseudo code that will make node to sleep when in idle state this will minimize energy wastage hence improving the live span of nodes.

The aim of this pseudo code is to make the node to sleep when they are in idle mode and wake up when the node are sensing and transmitting in order to save energy. The sleep and wake up method of saving energy looks at how to adjust the ratio between sleeping time and waking time of the sensor in each period as shown in the figure 4 below:

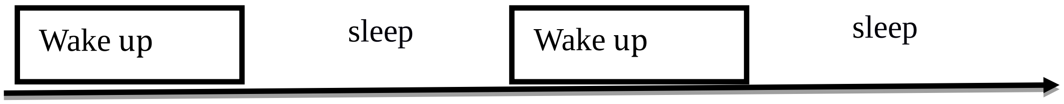


Figure 4: Problem formulation

In sleeping mode the sensor cannot receive or transmit any data (sleep state). In this state the sensors consume little amount of energy.

In wake up mode the sensor can receive and transmit data (wake up state). A node in this state consumes more energy compared to sleep state.

The nodes should adjust their sleeping time and wakeup time in order to save energy and guarantee efficient transmission of packets.

Sensor nodes have three modes of operation. These modes are transmit, listen, and sleep. In transmit mode the node can transmit and receive data. In listening mode the transmitter circuitry is turned off so the node can only receive data. While in sleep mode both transmitter and receiver are turned off. The pseudo code below will be used in this research to reduce energy consumption.

```
start
sensor modes transmit, listen and sleep
if(mode=transmit)
{
the node will transit and receive
}
else if(mode=listen)
{
the node will receive
}
else
{
the node will turn transmitter and receiver and sleep
}
```

Stop

Pseudo code for nodes to sleep and wake up to save energy

4.4 Hardware and Software Environment

Table 4.1: Hardware and software Specifications

Item Description	**Specification**
Desktop 2.00GB,Intel(R) Core i3	Hardware
Operating System	Windows 7 Linux Emulator -cygwin
Network Simulator	Ns-2 Verson 2.35

Table 5: Hardware and software specification

4.5 Implementing security protocols

NS-2 is not exclusively meant to support simulations in wireless sensor networks, however in practice it has been widely been used by researchers worldwide to evaluate sensor networks. To ensure NS-2 had the capability for WSN functionalities, NS-2 with mannasm framework was installed.

4.5.1 Simulation experiments

The following assumptions were made for the purpose of simulation.

The nodes have uniform energy of 10 Joules initially

The nodes are distributed randomly.

The nodes are immobile

The same Packet size for the node

The figure below show example of screen shots of the graphical user interface of my simulation tool it shows a medium 100 nodes of WSN. The upper most panel shows the input parameters, control button and the output of results from simulation and the bottom most panel shows WSN layout. There are control tabs of the GUI window: simulation and theory the theory allows the user to study the analytical results of graph theory and different parameters for comparison with the simulation result. Under simulation these are the sub tabs that lets the user enter simulation parameters, customization and view statistics and graphs based on the simulation results.

Figure 4 shows the simulation setup

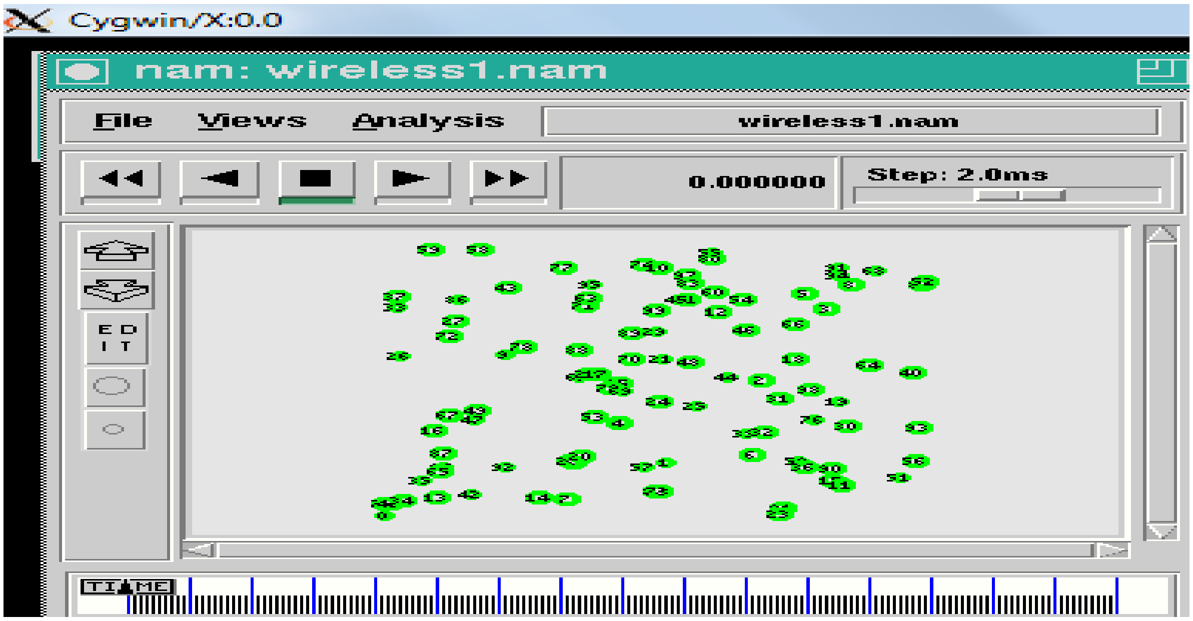


Figure 5: Field topography

On the parameter the user need to specify the number of simulation replications. This parameter can be changed in the data field. Once the run simulation button is hit, the WSN is simulated a number of times depending on the user requests. When the simulation is completed the results will appear. The results summarizes the simulation parameters and provides simulation results.

Using the GUI to study WSNs the user needs to specify the parameters for the simulation. Once the network has been simulated the network virtualization can be customized to show different information. Further we can examine the detailed report from simulation or view the graphs generated from simulation data.

Table 6 below shows the simulation parameters:

ITEM DESCRIPTION	**SPECIFICATION**
Simulation Field	1000m X 1000m
Channel type	Channel/wireless channel
Radio propagation model	Two ray ground
Number of nodes	100
Antennae model	Antenna

Energy model	Battery
Type of network interface	Wirelessphy
MAC protocol	Mac/802_11
Type of Interface queue	Queue/Drop tail/priqueue

Table 6: Simulation parameters

The simulation parameters shown in table 4.2 are explained as follows;

Simulation Field - It determines the area (dimensions) of the sensor field.

Channel type –It specifies the kind of channel being used.

Radio propagation model – It predicts packets received signal power.

NS-2 defines three propagation models namely;

Free space model. It has a direct line of sight between the transmitter and the receiver. The devices with direct line of sight can receive packets.

Two ray ground reflection models. It looks both at line of sight and ground reflection path between the transmitting node and receiving node. It gives accurate results even when the distance between the transmitter and the receiver is lengthy as compared to free space model. Shadowing model - This model plays a great role where the space (distance) between the transmitter and the receiver is long like in mobile communications.

Number of nodes. This refers to how many nodes have been deployed. Antennae model. The antenna type chosen is Omni directional Antenna since it has ability to transmit with equal power in all directions.

Energy model -It represents the amount of energy in the node.

Link layer type - Link Layer (LL) object simulates data link protocols.

Network interface type - It sets the power for transmitting based on distance approximated between the sender and receiver.

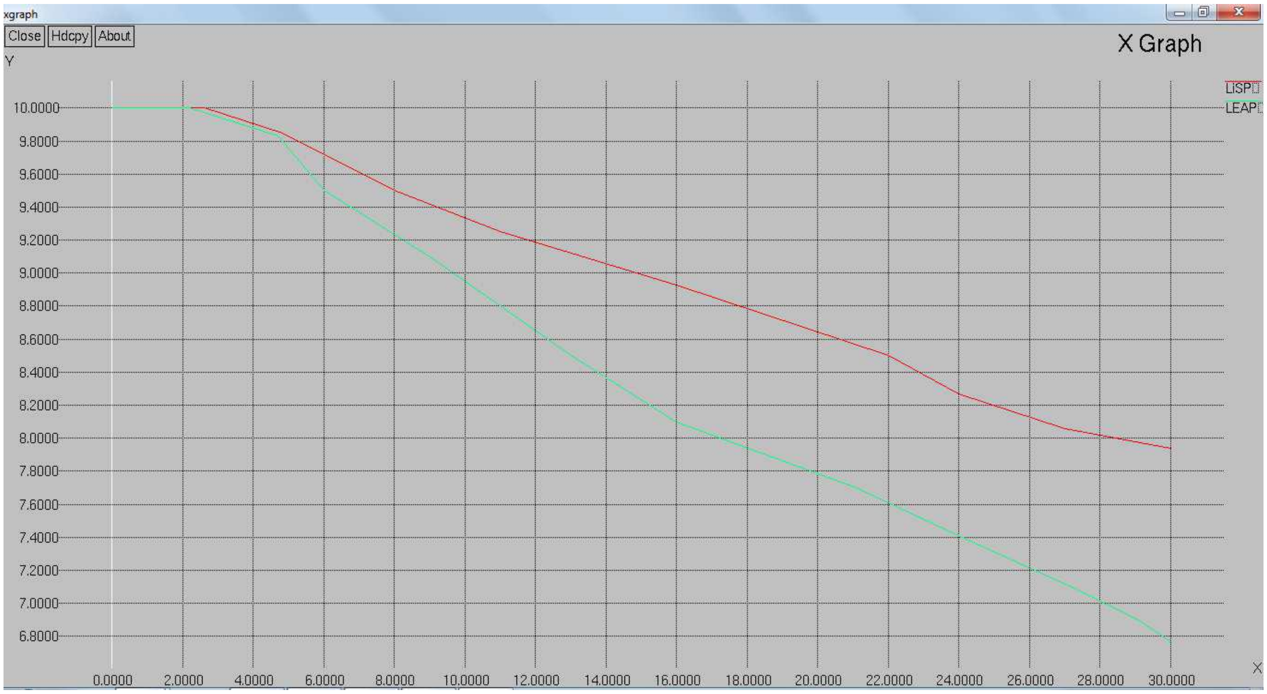
Interface queue type - The queue type used in the simulations is Drop Tail. This is a queue management technique that implements first- in- first- out mechanism.

4.6 Energy consumption

The core objective of the research was to determine selection of best security protocol based on energy utilization. Sensor nodes use battery and so they are limited in terms of energy and this significantly reduces lifetime of the network.

The assumption made of sensor nodes is that they are homogenous and the energy is 10 Joules (J) at the start of the simulation. The sensor node energy is exhausted during sensing and transmission of received signals, and the energy decreases with simulation time.

The graphs in figure 4.2 and figure 4.3 show the average energy amount in each sensor node at different time intervals.



Time(s)

Figure 6: Energy consumption in WSN_SIM 1

The Xgraph results show a slow start in energy consumption of LiSP and LEAP then gradually curves down. LiSP starts loosing energy after approximately 2.5s while LEAP drops after approximately 3.7s. The Xgraph shows that in any given simulation time the average energy amount in the sensor field is greater with LiSP in comparison to LEAP, at the end of simulation, which takes only 30 seconds.

Table 7: WSN_SIM 1 energy remaining

Protocol	**Energy Remaining(J)**
LiSP	7.9
LEAP	**6.79**

Table 7: WSN_SIM 1 energy remaining

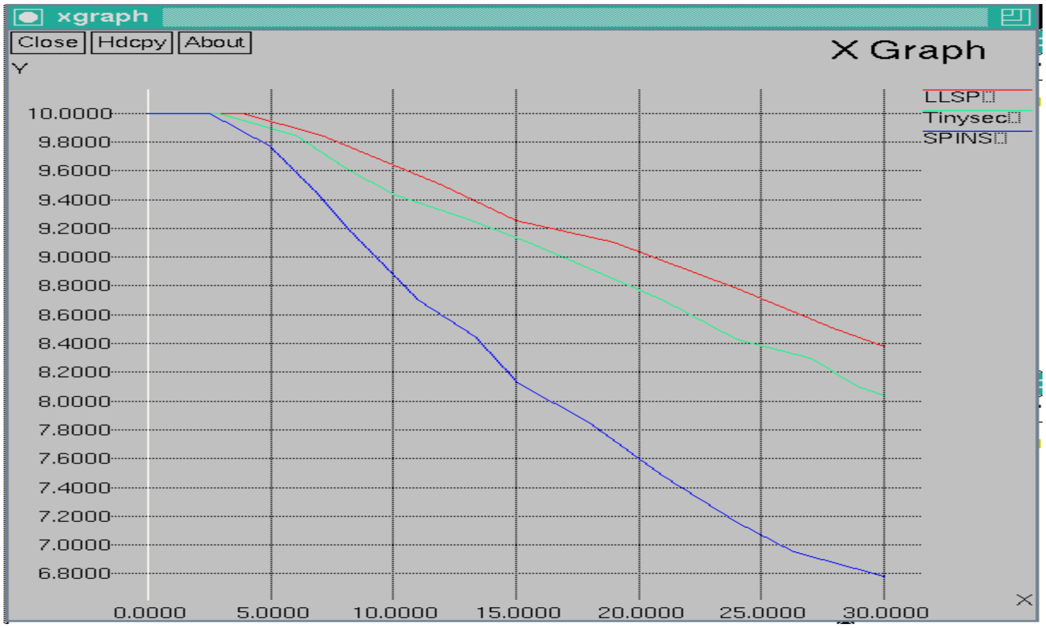


Figure 7: Energy consumption WSN_SIM 2

The X graph results show a slow start in energy consumption of LLSP, TinySec and SPINS then gradually curves down. SPINS starts loosing energy after approximately 2.5s, TinySec loses after approximately 3.7s and LLSP drops energy after approximately 3.9s. It is demonstrated that in any given simulation time the average energy amount is greater with LLSP in comparison to TinySec and SPINS, at the end of simulation, which takes only 30 seconds.

Table 8:WSN_SIM 2 energy remaining

Protocol	**Energy Remaining(J)**
LLSP	8.39
TinySec	8.05
SPINS	6.79

Table 8: WSN_SIM 2 energy remaining

4.7 Conclusion

From the study it was noted that there are several protocols that have been developed to secure WSN. Some of these protocols include SPIN, SNEP, LISP, LEAP and LLSP etc. During development of these protocols some assumptions were made hence making them not very secure.

Researchers have evaluated various protocols based on several matrices e.g. performance, data confidentiality, integrity and authenticity. In terms of performance various researcher evaluated Sleach, Leach and Dsdv and their study discovered that Sleach is better in terms of performance than the other two protocols.

From the above the researcher found out that energy is an essential factor because sensor nodes requires a battery in order to operate, the researcher decided to evaluate security protocols based on the energy consumption to identify the most suitable protocol in terms of energy efficiency.

During simulation it was noted that wireless radio as the most energy consuming unit of sensor node in WSN and operates in various state these are transmit, receive idle and sleep. Reducing energy consumption in WSN is by using only the required nodes as active hence reducing redundant traffic, decreasing packet delay thus avoiding packet collision. The other way is to put few sensor node to sleep and use the necessary in active mode for sensing and communication.

The simulations results shows that by comparing LiSP and LEAP which are in security class herein named as *WSN_SIM 1*, LiSP is better in terms of energy efficiency. LLSP, TinySec and SPINS are in the same security class referred as *WSN_SIM 2*, and results show LLSP is more energy efficient, Therefore it’s better to select LLSP protocol which is a hierarchical architecture security protocol for the applications that fall under this security class.

CHAPTER 5: CONCLUSIONS AND FURTHER WORK

5.1 Introduction

This research, introduced energy mechanism that can assist in reducing energy between various security categories and determined best security protocol. This chapter shows the accomplishments, challenges and recommendations and suggestions for future work.

5.2 Conclusions

A wireless sensor network (WSNs) operates by gathering and conveying the sensed (collected) data to a sink where it's processed further. Due to the limited resources in WSNs including small memory, low battery life, low processing power, and wireless communication channel, security becomes a major concern. The selected WSNs security protocol should therefore take into account the constraints of WSNs in order to prolong its life span. In selection of security protocol, energy efficiency should be a major factor in consideration, at the same time ensuring security is not compromised.

This research has achieved the objective by identifying various security protocols e.g. SPIN LISP, LEAP, LLSP and Tinysec. The research also determined various methods that can be used to compare and evaluate security protocols e.g. performance, energy consumption, memory and latency.

The research also simulated of several protocols and determined the their energy consumption, by first of all categorizing the security protocol into its security categories and then comparing protocols in same category, and finally got the best in a given security class based on energy metric. The result of simulations conducted for protocols considered in security class WSN_SIM_A, showed that LiSP security protocol is the best protocol in terms of energy efficiency, and for security class WSN_SIM_B results showed that LLSP security protocol is the best protocol in terms of energy efficiency.

The study shows that if one uses LLSP security protocol to secure the network. The network will have a longer battery life span compared to other protocols hence improve the lifespan of the WSN and reduces the cost and time of replacing the batteries. These will optimize data gathering and transmission.

5.3Accomplishments

The research was able to Implement SPINS, TinySec, LLSP, LiSP and LEAP security protocols in NS-2.Tool Command Language was used successfully used and also acquired the skills of simulating protocols using NS-2.

5.4Challenges

The installation of NS-2.35 simulator took long time, learning a new programming language known as tool command language took some time and protocols Integration in NS-2 was quite complex.

5.6 Future work

This researcher proposes all WSN security protocols in same class to be compared not only for energy utilization, but also other metrics to be considered such as jitter , latency, and memory usage, fault tolerance, high sensing fidelity, low- cost and rapid deployment, above all the application requirements. This is because a wide range of application will make sensor network an integral of our dairy live in future. Also, researcher must develop technologies needed for different layer of sensor network protocol stack.

Future research in WNS can also be directed toward increasing throughput and coming up with various techniques for conserving energy in clustered sensor network

REFERENCE

Ace Dimitrievski, B. S. (2012). Securing communication in WSN through use of cryptography. *NATO-ARW*.

Ahmed, A. S. (2009). An Evaluation of Security Protocols on Wireless Sensor Network. *Seminar on Internetworking.* Helsinki University of Technology.

Bhanu, P., & Saravanan, J. (2014). Data Security in Wireless Sensor Network. *International Journal of Innovative Research in Computer* , 3202-3203.

Chawla, H. (July 2014). Some issues and challenges of Wireless Sensor Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 237.

CHELLI, K. (July 2015). Security Issues in Wireless Sensor Networks: . *Proceedings of the World Congress on Engineering 2015* (p. 1). London, U.K.: Newswood Limited.

Chhimwal, M. P., Rai, D. S., & Rawat, D. (Mar. - Apr. 2013). Comparison between Different Wireless Sensor Simulation Tools . *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 5, Issue 2*, 55.

Constantinescu, C., K.Kanoun, H.Madeira, Murphy, B., Pramanick, I., & Brown, A. (2015). panel Statement. *presented at the international Confrence on Dependeble System and Networks.*

Delaney, P. M. (2014). An Introduction to NS, NAM and OTcl scripting.

G. Padmavathi, D. S. (2009). A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks. *International Journal of Computer Science and Information, Vol. 4*, 1-2.

G.Murugaboopathi, V.Geta, V.Sujathabai, babu, T., & S.Hariharasitaraman. (2012). An Analysis of Threat’s in Wireless Sensor Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 468.

Huang, B., Bauer, M., & Katchabaw, M. (2015). Hpcbench - A linux-based network benchmark for high performance networks. *Proceedings of the 19th international symposium on high performance* , (pp. 65-71).

Huang, B., Bauer, M., & M. Katchabaw. (2005). Hpcbench - A linux based network benchmark for high perfomance networks . *Proceedings of the 19th International symposium on hi Perfomance computing Systems and Application*, (pp. 65-71).

Jeffery Undercoffer, S. A. (2011). *Wireless Sensor Network* . Kiuwer Publication .

K. Kifayat, M. M. (2010.). Security in Wireless Sensor Network. *Handbook of Information and*, pp. 513-552.

Karlof, C., & D.Wagner. (2013). Secure Routing in Sensor Networks Attack and Countermeasures. *Elsevier's AdHoc Network Journal Special Issues on Sensor Networks (SNPA)*, 293-315.

Karlof, C., Sastry, N., & David Wagner. (2014). TinySec: Alink Layer Security Architecture for Wireless Sensor Network . *ACM SenSys*, 162-175.

KumarSingh, S., Singh, M., & Singh, D. (2010). Energy Efficiency Transmission Error Recovery for Wireless Sensor Network. *International Journal of Grid and Distributed Computing (IJGDC)*, 89-104.

M. A. Khan: G. A. Shah, M. S. (2011). Challenges for Security in Wireless sensor Networks (WSNs). *World Academy of Science, Engineering and Technology*, 80.

Maw, T. W. (May 2014). INTEGRATION OF SECURITY AND AUTHENTICATION. *International Journal of Information Technology, Modeling and Computing (IJITMC)*, 52-53.

maw, T. w., & jaw, M. h. (2013). A secure for mitigation of DoS attack in cluster Based wireless sensor networks. *International Journal of Computer & Communication Research*, 68.

Mishra, S., Mishra, S., Kayal, A., & Chudi, S. R. (July 2012). Simulation in Wireless Sensor Networks . *International Journal of Electronics Communication and Computer Technology (IJECCT) Volume 2 Issue 4*, 176.

Mohanty, P., Panigrahi, S. A., Sarma, N., & S. S. Satapathy. (2010). Security Issues in Wireless Sensor Network Data Gathering Protocols. *A Survey Journal of Theoretical and Applied Information Technology* , 14-27.

Muazzam A. Khan, G. A. (2011). Challenges for Security in Wireless sensor Networks (WSNs. *World Academy of Science, Engineering and Technology*.

Muazzam A. Khan, G. A. (2011). Challenges for Security in Wireless sensor Networks (WSNs). *World Academy of Science, Engineering and Technology*.

Neha Singh, P. R. (May 2012). Network Simulator NS2-2.35. *International Journal of Advanced Research in Computer Science and Software Engineering*, 225.

Neha Singh, P. R. (May 2012). Network Simulator NS2-2.35. *Network Simulator NS2-2.35*.

Neha Singh, P. R. (May 2012). Network Simulator NS2-2. *IJARCSSE*, 35.

Noman, A. (2008). Ageneric framework for defining security enviroments of wireless Sensor Networks. *Electrical and Computer Engineering(ICECE)*.

Nour El Din M. Khalifa, M. H. (December 2013). A Secure Energy Mechanism for WSN and Its Implementation in NS-2. *CiiT International Journal of Wireless Communication*, 984-990.

P. Sung, A. S., & Srivastava, M. (2001). Simulating networks of wireless sensors. *Simulation Conference*, (pp. 1330 – 1338).

Padmavathi, D. G., & Shanmugapriya, D. (2009). A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks. *International Journal of Computer Science and Information Security*, 5.

Padmavathi, G., & Shanmugapriya, D. (2009). A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks. *International Journal of Computer Science and Information Security, Vol. 4*, 1,2.

Paul, W. J., Zhengqiang, L., Weisong, S., & Vipin, C. (2006). Wireless Sensor Network Security. *Security in Distributed, Grid, and Pervasive Computing* (p. 8). Wayne State: Auerbach Publications.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., & Tygar, J. (2011). *Security Protocols for Sensor Network*, 521-534.

Pooja, M., & Singh, D. (2013). Security Issues and Sybil Attack in Wireless Sensor Networks. *International Journal of P2P Network Trends and technology*, 7-13.

Prajeet Sharma, N. S. (2012). A Secure Intrusion detection system against DDOS attack in Wireless Mobile Ad-hoc Network. *International Journal of Computer Applications Volume 41*, 21.

Prajeet Sharma, N. S. (2012). A Secure Intrusion detection system against DDOS attack in Wireless Mobile Ad-hoc Network. *International Journal of Computer Applications*, 21.

Prajeet, S., Niresh, S., & Rajdeep, S. (March 2012). A Secure Intrusion detection System against DDOS attack in Wireless Mobile Ad-hoc Network. *International Journal of Computer Application, Vol. 41*, 21.

Pugliese, M., & Santucci, F. (2008). Pair-wise Network Authenticated Hybrid Cryptographic Keys for Wireless Sensor Networks using Vector Algebra. *4th IEEE International Workshop on Wireless Sensor Networks Security*.

Rajra, B., & Deepa. (February 2015). A Survey on Network Security Attacks and PreventionMechanism. *Journal of Current Computer Science and Technology*, 4.

Rathod, V., & Mehta, M. (Jan 2011). Security in Wireless Sensor Network. *GANPAT UNIVERSITY JOURNAL OF ENGINEERING & TECHNOLOGY, VOL.-1*, 36.

Reddy, Y. (2011). Securурity Issues in Wireless Sensor Network. 14.

Reddy, Y. B. (2011). Security Issues In Wireless Sensor Networks. 14-16.

Ren, X., & Haibin Yu. (2006). Security Mechanism for Wireless Sensor Network. *International Journal of Computer Science and Network Security (Ijcsns)*, 155-161.

Ruiping Ma, L. x. (2012). Linear Crypatanalysis of A Survivable Data Transmission Mechanism. In L. x. Ruiping Ma, *Linear Crypatanalysis of A Survivable Data Transmission Mechanism* (pp. pp-562-567).

S.Princy, & Sasikumar, .. (Nov 2015). Security Challenges and Schemes on Wireless Sensor Network. *IJCSET*, 374.

Sarkar, N. I., Member, S., IEEE, & Halim, S. A. (2011). A Review of Simulation of Telecommunication Networks. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT),* , 10.

Sarkar, N. I., Member, S., IEEE, & Halim, S. A. (2011). A Review of Simulation of Telecommunication Networks. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, (p. 14).

Sharma, G., Bala, S., & Verma, A. K. (2012). Security Frameworks for Wireless Sensor Networks. *2nd International Conference on Communication, Computing & Security [ICCCS-2012]*, (p. 979). Thapar University, Patiala-147004, India.

Sharma, G., Bala, S., & Verma, A. K. (2012). Security Frameworks for Wireless Sensor Networks. *2nd International Conference on Communication, Computing & Security [ICCCS-2012]* (p. 979). Patiala: Elsevier Ltd.

Sharma, R., Chaba, Y., & Singh, Y. (Aug 2010). Analysis of Security Protocols in Wireless Sensor Network . *International Journal of Advanced Networking and Application*, 707-713.

Singh, S. M., & Singhtise, D. (2011). A survey on network security and attack defense mechanism for wireless sensor networks. *International Journal Computer. Trends Technology*, 5-6.

Sinha, S., Chaczko, Z., & Klempous, R. (2009). SNIPER: A Wireless Sensor Network Simulator, Computer Aided Systems . *EUROCAST Vol 5717/2009*, 913-920 .

Stallings, W. (2000). *Cryptography and Network Security Principles and Practice.* Prentice-Hall: Prentice-Hall.

Xueying Zhang, H. M. (2010). Energy Efficiency of Symmetric key Cryptographic Algorithms in Wireless Sensor. 168-172.

Zhu, S., Setia, S., & Jajodia, S. (2003). LEAP: Efficient Security Mechanism for large scale Distributed Sensor \\\\\\network. *ICCS"03"Proceedings of the 10th Confrence on Computer and Communication Security*, (pp. 62-72). New York , USA.

Zia, T., & Zomaya, A. (November 3, 2006). Security Issues in Wireless Sensor Networks. *Proceedings of the International Conference on Systems and Networks Communications*, (p. 1). Tahiti

Appendix 1

Compilation and Installation of NS-2.35

Installation of NS-2.35 was done on Cygwin, a Linux emulator on windows. The following packages were installed.

- Tcl version 8.5.10
- Tk version 8.5.10
- Otcl version 1.14
- Tclcl version 1.20
- Ns2.35 version
- Nam version 1.15
- Xgraph version 12.2

Xgraph produces graphical results under Cygwin platform.

The following steps were followed in the installation process

Ns-allinone was downloaded and extracted to C:\home\user

To install the packages from extracted file,following commands were executed

```
 > cd c:
  > cd cygwin
 > cd home
> cd user
 > cd ns-allinone-2.35
 > ./install  (This command initiates the process of installing NS-2)
```

- In BASHRC File,following paths were set
- NS_HOME=c/cygwin/home/user/ns-allinone-2.350
- Export PATH=$NS_HOME/nam1.15:$NS_HOME/tcl8.5.10/unix:
- $NS_HOME/tk8.5.10/unix:$NS_HOME/bin:$PATH
- export LD_LIBRARY_PATH=$NS_HOME/tcl8.5.10/unix:
- $NS_HOME/tk8.5.10/unix:/$NS_HOME/otcl1.14:$NS_HOME/lib:$LD_LIBRARY_PAT

H

- export TCL_LIBRARY=$NS_HOME/tcl8.5.10/library

Open CYGWIN bash prompt and got to Ns-allinone-2.35

Type startx or startxwin

Xserver window as shown below opens (This indicates NS-2 has been installed successfully)

Appindex 2

These code were used to set simulation parameters

```
## Setting The wireless Channels
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(ifqlen) 40
set val(nn) 100
set val(rp) DSR
set val(x) 1000
set val(y) 1000
set val(stop) 30.0
# Create a simulator object
set ns [new Simulator]
# Create a trace file and nam file
set tracefd [open wireless1.tr w]
set namtrace [open wireless1.nam w]
# Trace the nam and trace details from the main simulation
$ns trace-all $tracefd
```

```
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
set god_ [create-god $val(nn)]
## Color Descriptions
$ns color 1 green
$ns color 2 blue

# Setting node config event with set of inputs..
puts "Node Configuration Started here...\n \
-channel $val(chan) \n \
-adhocRouting $val(rp) \n \
-llType $val(ll) \n \
-macType $val(mac) \n \
-ifqType $val(ifq) \n \
-ifqLen $val(ifqlen) \n \
-antType $val(ant) \n \
-propType $val(prop) \n \
-phyType $val(netif) \n"
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
 -macTrace OFF \
-movementTrace ON
# Energy model
$ns node-config -energyModel EnergyModel \
-initialEnergy 10 \
-txPower 0.9 \
-rxPower 0.8 \
-idlePower 0.0 \
-sensePower 0.0175
## Creating node objects..
for {set i 0} {$i < $val(nn) } { incr i } {
set node_($i) [$ns node]
}
for {set i 0} {$i < $val(nn)} {incr i} {
$node_($i) color green
$ns at 0.0 "$node_($i) color green"
}
## Provide initial location of mobilenodes..

if {$val(nn) >0} {
for {set i 1} {$i < $val(nn) } { incr i } {
set xx [expr rand()*1000]
set yy [expr rand()*1000];
$node_($i) set X_ $xx
$node_($i) set Y_ $yy
}
}

## Define node initial position in nam..
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam..
$ns initial_node_pos $node_($i) 30
}
# informing nodes end of simulation
for {set i 0} {$i < $val(nn) } { incr i } {
$ns at $val(stop) "$node_($i) reset";
}
# End nam and simulation..
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 30.01 "puts \"end simulation\" " ;# $ns halt
## Stop procedure..
proc stop {} {
global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
exec nam wireless1.nam &
exec xgraph wireless1.tr   -geometry 500 x 500 &

exit 0
}
$ns run
```

Appendix 3

Installation of Mannasim in ns-2.35

The following are the steps for installation of Mannasim framework

Step 1: Download Mannasim.tar.gz file for ns2.35 from this site (https://dl.dropboxusercontent.com/u/24623828/mannasim/mannasim.tar.gz)

Step 2: The folder is unpacked inside the ~ns-2.35/ folder and inside the mannasim/ folder

Step 3: Copy the files from the ns-modified-files and substitute with the ones in these locations

- ns-allinone-2.35/ns-2.35/apps/udp.cc
- ns-allinone-2.35/ns-2.35/common/ns-process.h
- ns-allinone-2.35/ns-2.35/common/packet.cc
- ns-allinone-2.35/ns-2.35/common/packet.h
- ns-allinone-2.35/ns-2.35/Makefile.in
- ns-allinone-2.35/ns-2.35/tcl/lib/ns-default.tcl
- ns-allinone-2.35/ns-2.35/tcl/lib/ns-lib.tcl

Step 4: Once everything is done, go to the terminal (ns-2.35 folder) and type the following commands one by one

./configure (This command is for configuring script)

./make (This command is for re-compiling the system)