A MODEL FOR SELECTING SECURITY PROTOCOLS FOR WIRELESS SENSOR NETWORKS

By

JOHN GICHUKI NDIA

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

NOVEMBER, 2013

DECLARATION

I declare that this Research project is my original work and has not been previously published or submitted elsewhere for award of a degree. I also declare that this Research project contains no material written or published by other people except where due reference is made and author duly acknowledged.

Student Name: John Gichuki Ndia Reg, No. 11/01715

Sign: _____Date: _____

I do hereby confirm that I have examined the master's Research project of

John Gichuki Ndia

AND have certified that all revisions that the Research project panel and examiners recommended have been adequately addressed.

Sign:_____ Date: _____

Supervisor's Name: Dr. Cyrus Wekesa

A MODEL FOR SELECTING SECURITY PROTOCOLS FOR WIRELESS SENSOR NETWORKS

ABSTRACT

The process of mapping security requirements to the most appropriate security protocol has over the time proved a great challenge. Though there are various security mechanisms designed to curb security threats, they come with various properties and therefore the choice of the best security protocol for a given application becomes quite complex. To ease the process of mapping security requirement of sensor applications to security protocol, security environments for WSNs have been defined formally.

There are numerous WSNs applications being developed day to day, ranging from simple environmental monitoring e.g. collecting of temperatures in an agricultural farm to complex applications like for monitoring battle field. Therefore this research dissertation objective was to enable selection of best security protocol that falls under a certain security class for the various existing WSNs applications and applications to be developed in the future. The research endeavored to identify and evaluate the security protocols that are practically used in WSNs and to identify the best tool to be used in simulation process, and finally to validate selection of security protocols.

WSNs have become a hot area of research especially in issues of security, routing, systems hardware design and data management. This research contributes to the area of security by focusing more on selection of security protocols so as to meet the security requirements of WSNs applications at the same time ensuring the sensor networks lifetime is extended as long as possible. There are different methods for evaluation of WSNs security, in this research simulation methodology was chosen based on the fact that you can generate reliable results in less time, in addition the simulation tool used is open source, thus cutting down simulation costs.

This research project introduced energy metric within a security class and compared protocols in the same security class to check which protocol utilizes lesser energy. The simulation tool used was network simulator version 2.35 which at the time of research was the latest release. The ns2.35 was installed together with mannasim framework, which extends ns2 for WSNs simulation.

Security protocols were classified into four security classes also called security environments namely WSN_Env_A, WSN_Env_B, WSN_Env_C and WSN_Env_D and compared them within their security class using energy metrics; The results showed that in class WSN_Env_A LiSP is the best protocol while in WSN_Env_B class LLSP is the best protocol based on energy efficiency.

Keywords: Wireless sensor Networks, Security Class, Security Protocols, Simulation, Sensor nodes, Security Services

TABLE OF CONTENTS

CONTENT **PAGE**

DEDICATION

I dedicate this work to my lovely wife Velma Auma and my parents Danson N. Gichuki and Esther T. Ndia. Your continued encouragement has always been a blessing to me.

ACKNOWLEDGEMENT

I take this first opportunity to thank God, by whose grace and endowment of health and focus I began and ended this research work.

It is with distinct pleasure I express my deep sense of gratitude to my supervisor Dr. Cyrus Wekesa for his extraordinary and invaluable guidance .He walked with me in this journey of research and never got tired in reviewing this work, and did it with great and positive attitude such that I accomplished this work on time.

I also sincerely thank Prof. Ddembe who provided me with great insight on writing of this research project. I wish to thank all lecturers who took me through units that opened my eyes to the hot areas of research in data communication, this includes: Prof Ogao, Dr. Kanyi and Dr. Musyoki Stephen.

Last but not the least am greatly indebted to everyone who assisted me even with the slightest idea on my research project and so I express my heartfelt gratitude to you.

LIST OF FIGURES AND TABLES

ACRONYMS AND ABBREVIATIONS

WSNs	Wireless Sensor Networks
LLSP	Link Layer Security Protocol
SPINS	Sensor Protocols for Information Via Negotiation
LiSP	Lightweight security protocol
LEDS	Location aware end-to-end security
LEAP	Localized encryption and authentication protocol
MAC	Message authentication code
DSS	Digital signature scheme
KS	Key-Server
CSMA	Carrier-sense Multiple Access
TK	Temporary Key
CPU	Central Processing Unit
OTcl	Object-oriented Tool command language
NS-2	Network simulator version two
GH	Group Head

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND TO THE STUDY

A wireless sensor network (WSNs) comprises of many identical nodes with limited resources. Sensor nodes communicate wirelessly and they intelligently process signals and transmit data over the networks. These nodes are normally spread over the whole network area for monitoring, data collection, processing, and forwarding to a base station to process further (Sharma, Chaba & Singh, 2010).

The Sensors are small in size, limited in terms of power and their cost is normally low. Sensors have the following capabilities: communication is over short distances, they can sense or read data from the environment, and their data processing capability is limited. Normally sensor operates at 2.4 GHz frequency, 250Kbps data rate, flash memory is 128KB, memory of 512KB for purpose of recording measurements, they transmit powers ranging from 100uW and 1mW, and communication range is between 30m to 100m. Therefore, the greatest design consideration should be energy efficiency of WSN protocols (Uluagac et al., 2008).

The greatest challenge for WSNs are security issues, and for certain sensor networks applications, like health care applications and military applications security becomes even more crucial. These challenges are as follows;

i. It's difficult to protect wireless communication since it is done by broadcasting. Packets can be injected, eavesdropping is a possibility, interception of moving data, and data transmitted can be altered easily by adversaries.
ii. The WSNs may be installed in environments that are potentially insecure; where there is a possibility for adversaries to masquerade as authorized nodes in the network, and nodes stealing can occur.

iii. The WSNs are susceptible to attacks of consumption of resources. Attackers can waste network bandwidth and frequently send packets to exhaust a node battery.

Due to these factors, it's essential for the sensitive digital information to be securely transmitted over the sensor networks.

To provide security in wireless sensor networks, security schemes or mechanisms can be useful, but due to limited and starved nature of the resources it's quite complex to do so. A significant number of WSNs security protocols utilize symmetric key cryptography.

WSNs security protocols like TinySec, Localized Encryption and Authentication Protocol (LEAP) and SPINS have been developed for provision of security services. According to Sharma, Chaba & Singh, 2010 the security requirements or services are such as; availability, authorization, authentication, confidentiality, integrity, non-repudiation, data freshness, robustness, self-organization and time synchronization.

1.2 STATEMENT OF THE PROBLEM

Several security protocols have been introduced for sensor networks although they are designed with different assumptions on them. In addition it's a great challenge to identify the security protocol most suitable for different kinds of applications used with sensors (Noman, 2008).

The process of identifying appropriate security requirements and be able to select the security protocols to fit into the requirements identified is quite a difficult task (Shohel, 2009). Security classes or security environments for sensor networks have been defined formally to ease the process of identifying the security protocols that sensor applications may require.

This research introduces energy metric within a security class, which will help in determining best security protocol based on energy consumption.

1.3 RESEARCH OBJECTIVES

1.3.1 MAIN OBJECTIVE

The core research objective was to determine the best security protocol in a given security class for wireless sensor networks.

1.3.2 SPECIFIC OBJECTIVES

i. To identify and evaluate wireless sensor networks security protocols and security classes.

ii. To determine tools to facilitate best security protocol selection

iii. To implement and validate best security protocol selection for a given security class.

1.4 SIGNIFICANCE OF THE STUDY

Wireless Sensor Network (WSNs) is a promising technology for diverse applications for the future. WSNs are up-and-coming as a hot area for active research involving security, data management, systems hardware design, programming models, distributed algorithms, networking, and social factors. The sensing technology which has wireless communication and processing power capability makes it worthwhile for its great exploitation in future (Sushma, Deepak & Vikas, 2011).

The past few years, Wireless Sensor Networks (WSNs) have demonstrated great potential, as integral components of solutions, applicable in various domains, such as observation of environmental conditions, military monitoring, surveillance, smart (home) environments and ambient assisting living(Zahariadis et al.,2009). Technically the major challenge for WSNs is basically the limited energy with battery powered nodes, and this poses a fundamental challenge in terms of limitation of network lifetime.

WSNs are used in lots of applications with different security requirements. E.g., an application for environmental monitoring demands less security whereas; battlefield monitoring applications demands high security levels. For environmental monitoring applications in-network processing is vital to reduce the network contention (Shohel, 2009).

Wireless sensor network applications are increasing day by day and it's important before their deployment we can tell the security protocols to implement and much more the best of all the security protocol that fall within a given security class. This will go a long way in furtherance of lifetime of the sensor networks.

1.5 SCOPE OF THE STUDY

This research used a simulation model to determine selection of best security protocol. The selection is basically based on energy consumption of security protocols.Mannasim framework was used to extend NS-2.35 for WSN functionalities. Protocols which fall under the same security class were implemented in NS-2.35 and simulation was used to compare protocols in terms of energy efficiency.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

This chapter focuses on presenting literature for security protocols used in practice by WSNs, security services, attacks in WSNs and security classes.

2.2 SECURITY PROTOCOLS.

Security protocol is defined as a set of rules that determine how the interaction between peer processes to make available a given security service (Aseri & Singla, 2011).

2.2.1 TINYSEC

TinySec is a link layer security protocols for Wireless sensor networks (WSNs). The provision of passive communication (in-network processing) is done by Link layer security among local nodes to eliminate communications that are overlapping with the sink (base station) (Shohel, 2009).

Karlof et al., 2004 designed TinySec to replace the incomplete Sensor Network Encryption Protocol (SNEP), called TinySec. TinySec is link layer security architecture for WSNs and it offers security services such as access control, confidentiality, and message integrity.

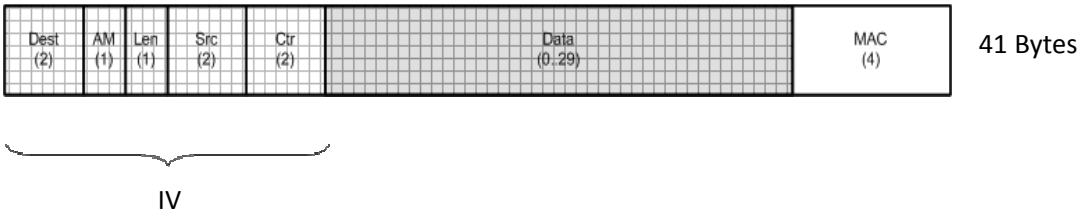
Integrity and access control are ensured through authentication method referred to as MAC (Message Authentication Code), and confidentiality through encryption method referred to as CBC (Cipher Block Chaining) mode. A unique initialization vector (IV) provides semantic security for each invocation of the encryption algorithm. This means there should be no guessing of any no or yes question as regards a given message by adversaries for no more than 50% probability. Initialization vectors (IVs) provides variation to encryption and this is necessary when variation of messages to be encrypted are few (Karlof et al., 2004).

TinySec is a lightweight security implementation which creates additional overhead to the TinyOS system to the minimal. TinySec is also highly (fully) compatible with security protocols at higher levels and is fully transparent to upper layer security protocols. In addition, TinySec is a software based cryptographic method. TinySec provides two (2) options for security; (Tinysec-Auth) authentication only and (TinySec-AE) authenticated encryption (Shohel, 2009).

Tinysec-AE uses MAC to authenticate a packet and also encrypts the data payload. The MAC field is computed over the encrypted data and the packet header. It ensures data is received from a legitimate node; moreover it prevents any attacker from seeing data. On the other hand Tinysec-Auth only authenticates the packet with a MAC without encrypting the data payload. Tinysec-Auth only ensures data is received from a legitimate node and is well implemented where data does not need to be private (Karlof et al., 2004).

(a)TinyOS Packet Format

(b)Tinysec-Auth Packet Format



(c)Tinysec-AE Packet Format

Figure: 2.1 Packet formats (Karlof, Sastry & Wagner, 2004)

KEY

Dest : Destination

AM : Active message handler

Len : Data length

Grp : Group field

Src : Address source

Ctr : Counter

CRC : Cyclic redundancy check

TinySec retains field Dest, AM, Len, and Data, and removes Grp field, and 4-byte MAC replaces CRC field. In addition TinySEC-AE adds two fields; src (source address) of the sender, and two (2)-byte counter of Ctr. The counter begins from 0, and the sender increases it by one (1) for every message sent (Karlof et al., 2004).

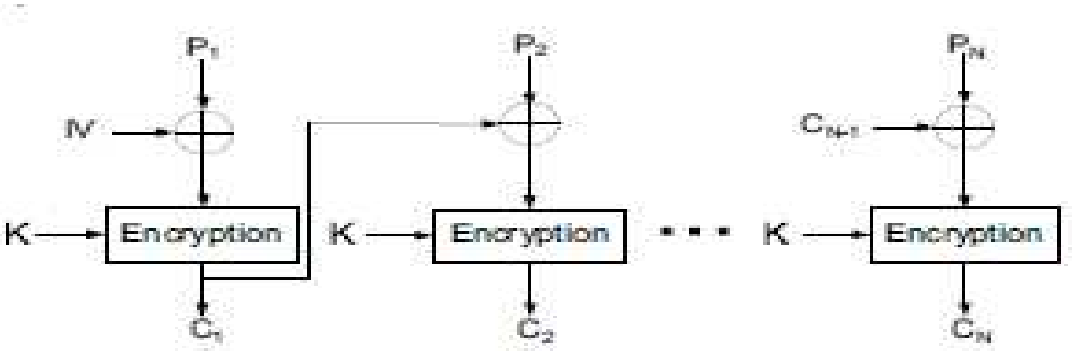


Figure: 2.2 Cipher Block Chaining (CBC) mode encryption (Lightfoot, Jian & Tongtong, 2007)

CBC mode has each of its blocks of plaintext XORed with the previous ciphertext block before encryption. Therefore, cipher text block depends on all plaintext blocks processed up to that point. An initialization vector is used in the first block to make each message unique (Karlof et al., 2004).

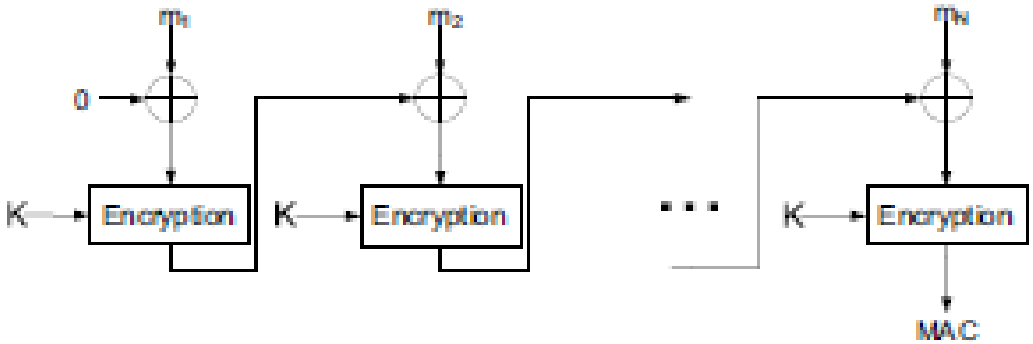


Figure: 2.3 Cipher Block Chaining Message Authentication Code (CBC-MAC)(Lightfoot, Jian & Tongtong, 2007)

The technique of CBC-MAC is used to construct a message authentication code from a block cipher. A chain of blocks is created by the message which is encrypted with some block cipher algorithm in CBC mode so that each block depends on the proper encryption of the previous block. This interdependence ensures that a variation to any of the plaintext bits will cause the last encrypted block to change in a way that cannot be predicted without the knowledge of the block key (Karlof et al., 2004).

2.2.2 LINK-LAYER SECURITY PROTOCOL (LLSP)

Lighfoot et al., 2009; designed a Link-Layer Protocol (LLSP) and the goal was to develop a protocol with low energy requirements as compared to TinySec. LLSP ensures message confidentiality, message authentication, replay protection and access control. LLSP is different from TinySec in that it uses different crypto structure and packet format. LLSP supports early rejection capability in addition It has low performance overhead. However maintaining a large network is difficult with in node counter due to that it has low scalability.

The data encryption scheme implemented in LLSP is Advanced Encryption Standard with cipher block chaining (AES-CBC) mode. The AES-CBC unique design provides semantic security; this means, same plaintext encrypted more than once will generate different cipher texts. To assure replay protection a synchronous four (4) byte counter is proposed between the sending and receiving node. Feedback Shift Register (FSR) is used to update this counter. A synchronous counter is maintained between sender and receiver; therefore, the counter value doesn't have to be transmitted, and from each message packet counter bytes are discarded (Lighfoot et al., 2009).

Figure: 2.4 LLSP Packet format (Krzysztof & Ewa, 2012)

2.2.3 SPINS

SPINS was proposed by Perrig et al., 2002, and it's a collection of security protocols optimized for sensor networks. SPINS has two secure building blocks specifically Secure Network Encryption Protocol (SNEP) and ^{u}TESLA. SNEP provides data authentication for two parties, confidentiality of data, and freshness of data while ^{u}TESLA authenticates broadcasts.

Limited storage hurdle is achieved by protocols through the reuse of code for all crypto primitives such as, message authentication code, encryption, and hash random number generator. In addition, to reducing the communication overhead, it shares the common state between communication parties. Semantic security is achieved through SNEP by incorporating counter in both sender and receiver ends. It's important to note that the counter is not incorporated with the message so as to reduce the data transmission rate (Shohel, 2009).

SNEP supports simply base-to-node communication and vice-versa while ^{u}TESLA provides authenticated broadcast. Traditionally to authenticate broadcasts you require asymmetric keys to authenticate the initial packets, but ^{u}TESLA uses symmetric key to provide security with symmetric keys disclosure delayed. Unfortunately with a network of many nodes synchronization is a challenge (Shohel, 2009).

2.2.4 LIGHT WEIGHT SECURITY PROTOCOL (LISP)

LiSP is a lightweight security mechanism that supports key renewability and puts into balance the need for security and consumption of resources. LiSP from time to time renews the shared key to solve the problem of reuse of key stream-reuse and maximize energy efficiency and scalability. LiSP also supports distribution of keys which is reliable (Taejoon & Kang, 2004).

LiSP is efficient in terms of energy and is robust to denial of service (DoS) attacks, since it doesn't require retransmitting or any control packets. LiSP has a joint authentication and recovery algorithm for rekeying, where Key -Server (KS) from time to time a new key is

broadcast before it's used for encryption and decryption. The key received is authenticated by client node and then recovers all keys that have been missing (Taejoon & Kang, 2004).

The goal of LiSP is to offer a lightweight security solution for a large-scale network of resource-limited sensor devices. LiSP divides the whole network into clusters and selects a Group-head (GH) for each of them to offer scalability for a large number of sensors (Taejoon & Kang, 2004).

LiSP uses KS to control the group security and for a sensor network that has multiple groups; LiSP assigns group head (GH) as Key server. In general, we can say there is existence of one (1) KS for each group. In addition LiSP has a Key Server for the network (KSN) that coordinates KS's in re-keying for intergroup communications (Taejoon & Kang, 2004).

LiSP uses a stream cipher which is cheaper and processes fast, and supports periodical renewal of keys with cryptographic hash algorithms which are low-priced. LiSP is reliable, and it works well with Carrier Sense with Multiple Access (CSMA) protocol that do not support reliable broadcast. In addition, it requires loose time synchronization among group members (Taejoon & Kang, 2004).

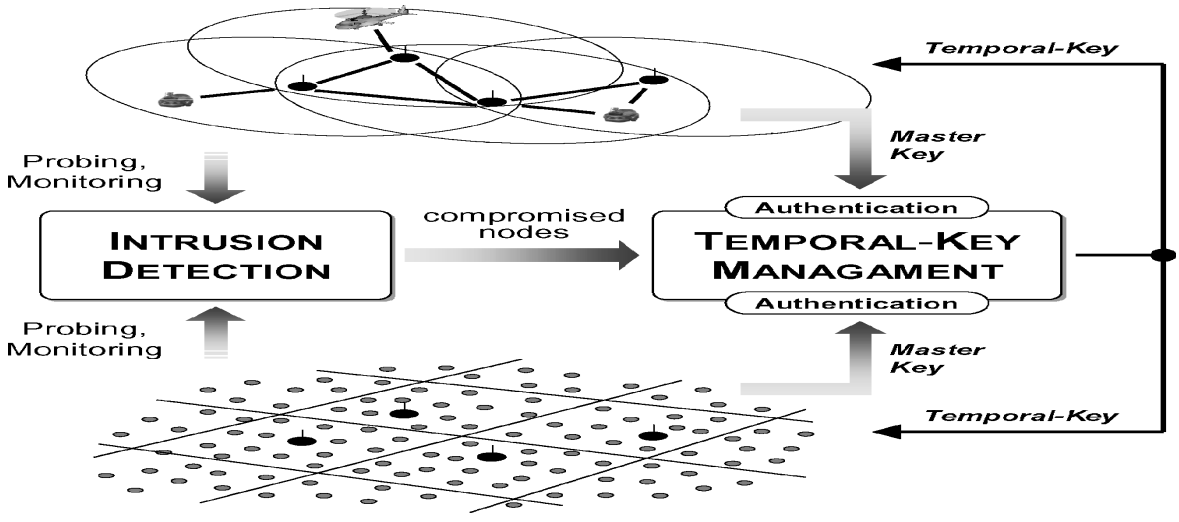


Figure: 2.5 LiSP Architecture (Taejoon & Kang, 2004*)*

The Figures 2.5 above shows the two main components for key hierarchy;

i. Intrusion detection- it checks activities of the network to unearth nodes that are compromised.
ii. Temporary Key (TK) management-it protects or shields network traffic from adversaries by periodical rekeying of TK.

2.2.5 LOCATION AWARE END-TO –END SECURITY (LEDS)

LEDS offers location aware end-to-end security. Several sensing nodes endorse genuine event reports in LEDS and are encrypted with a unique secret key which is shared between the sink and event sensing nodes. LEDS provides end-to-end authentication and en-route filtering capability to deal with the recognized attacks for injection of data. If there are no more than a given stated number of compromised nodes in each single area of interest, LEDS assures that a fake or false data report from a given cell can be filtered by genuine in-between sink or the nodes (Kui, Wenjing, & Yanchao, 2007).

LEDS provides location aware key management. LEDS can be used in both small and large networks and the key numbers increases with size of the cell. In addition, LEDS doesn't support dynamic topology. LEDS puts the network into several cell regions and when an event occurs in a given region, the event should be sensed by several nodes (Shohel, 2009).

Data availability is assured by LEDS because it deals with both report disrupting attack and selective forwarding attack at the same time. Wireless links are broadcast in nature and so LEDS adopts one node to many nodes data forwarding approach, this ensures LEDS reports are authenticated by several next-hop nodes separately. This means that no reports disappear due to being dropped by a single node. (Anjali et al., 2011)

LEDS ensures a very high level of security without considering the costs for communication and computing in addition LEDS provides data confidentiality and node capture attacks to a reasonable level (Shohel, 2009).

2.2.6 LOCALIZED ENCRYPTION AND AUTHENTICATION PROTOCOL (LEAP)

Zhu et al., 2003 proposed LEAP as a key management protocol for WSNs, after observing that different kinds of messages transmitted in wireless sensor networks (WSNs) demands different security requirements. The key design aims of this protocol are; robustness, lightweight, survivability and energy efficient operation.

LEAP has four (4) different keying mechanisms which are;

i. Individual keys-This key are shared between every node and sink (base station). This provides confidentiality in communication between individual nodes and base station.
ii. Group key- Encrypted messages from the base station are sent using this key to the whole wireless sensor network. They are used to send queries to the network nodes.
iii. Cluster key-It's like group key but it's shared between a node and its neighbor. It's used to broadcast messages locally in a secure manner.
iv. A pair wise shared key- This key is shared by all nodes with their closest (immediate) neighbors.

2.3 SECURITY IN WSNs

The universal approach for defense against cyber attacks is cryptography, but there exists challenges in keeping required level of security and safety of critical data transmitted over wireless sensor network.

The conventional computer networks don't have the challenges inherent in WSNs.

Table 2.1: Comparison of WSN and Traditional networks

Wireless sensor networks (WSNs)	**Conventional(Traditional) networks**
Bandwidth is less	More bandwidth
Devices have very little computational power	Comparatively devices have more computational power
Energy is less with wireless sensor devices	Energy for devices is comparatively high
Information is mostly transmitted in hop-by-hop	Information is mostly transmitted using end to end
Vulnerable to resource consumption	Not vulnerable to resource consumption
Quite difficult to protect	Comparatively much easier to protect

The primary objective of WSNs is to extend the life of sensor nodes. To realize this, it is important to minimize energy used by reducing the energy amount of node for node transmission and using energy aware protocols and algorithms. The balance between security and lifetime of the WSNs has to be achieved in consideration of the limited resources of WSNs nodes.

It's difficult to implement strong security protocols which based on asymmetric cryptography because, asymmetrical signatures are lengthy and need high communication overhead, hence they are not practical for WSNs applications. On the other hand, weak security protocols based on symmetric cryptography may be easily broken. It is crucial to design WSNs putting into consideration security right from the start.

2.4 SUMMARY OF ATTACK PROTECTION

Wireless Sensor Networks (WSNs) are open to to diverse attacks. The attacks are categorized into; (Rajkumar, Sunith, & Chandrakanth, 2012)

i. Secrecy and authentication attacks –These attacks are such as spoofing, eavesdropping, and packet replay attacks.
ii. Attacks on network availability-These attacks are also known as denial-of-service (DoS) attacks.
iii. Stealthy attack against service integrity-The attacker makes the WSN acknowledge a false data value. E.g. through injection of false data value.

The table below shows the levels of protection against attacks. The levels can be weak (no protection at all), partial (very low protection), medium (average protection) or strong (protection assured).

Table 2.2: Attacks Protection

Attack	**Tinysec**	**LLSP**	**SPINS**	**LiSP**	**LEDS**	**LEAP**
Replay	Weak	Strong	strong	strong	Strong	strong
Injection	Partial	Partial	partial	medium	Strong	partial
Alternation	Partial	partial	partial	medium	Strong	medium
Node capture	Weak	Weak	weak	partial	Medium	partial
DOS	Weak	Weak	weak	partial	Medium	partial

2.5 SECURITY SERVICES

According to Uluagac et al., 2008, security requirements can be looked at in these two dimensions;

i. The meaning of a specific security service in Wireless Sensor Networks (WSNs) field.

ii. The state of art of the security service.

2.5.1 CONFIDENTIALITY

Confidentiality is referred to as the capability to hide messages from any given adversary (attacker) to ensure any message transmitted through the WSN is confidential (Padmavathi & Shanmugapriya, 2009). Incase a rival, accesses the content, he should not be able to decode the messages exchanged in the network.

To provide a confidential security service to WSNs applications you require the use of cryptographic mechanisms such as encryption techniques. Generally, two kinds of encryption approaches are used;

i. Symmetric encryption

ii. Asymmetric encryption.

Symmetric encryption uses the identical key at both the sender and receiver nodes to encrypt and decrypt the information from plain text to cipher text and vice versa. While asymmetric key based encryption, uses dissimilar keys, one public and the other private which are used to convert and recover the information (Uluagac et al., 2008).

There is no single encryption mechanism that one can claim is better than another as it is basically a matter to do with size of the key and the computational effort that can be used to break the encryption algorithm.

Another facet to confidentiality research in WSNs is on issue of designing efficient key management schemes. The keys must always be available to all the nodes communicating and this ensures privacy of channels is maintained (Uluagac et al., 2008).

The process of managing keys involves two basic steps;

i. Key generation
ii. Keys distribution

This process is triggered by keying events like network attack. However, it's not a simple task and in a number of applications it may be overwhelming operation to go to each and every sensor considering their numerous numbers and updating of their keys, for-example underwater sensor applications. Therefore, management of keys intelligently is essential for WSNs (Uluagac et al., 2008).

2.5.2 AUTHENTICATION

This security requirement ensures that there is valid communication from a given node to another node; this means an untrusted node cannot pretend as a trusted node (Rajkumar.et al., 2012.).

The X.800 specification recommends two kinds of authentication.

i. Peer entity Authentication
ii. Origin of data authentication

Peer entity authentication means that all participating nodes in a communication are authenticated. Two nodes or one node can be authenticated, for-example cluster-head and numerous other nodes around a given node. Origin of data authentication can be implemented at the base station or at any other sensor node where aggregation of data takes place (Uluagac et al., 2008).

The conventional authentication methods discussed in the literature are.

i. Password based method - A password is sent by a node with its login information and the password is verified by the receiving node to ensure it's associated with the sending node.
ii. Cryptographic-based method/Challenge response-Message Authentication Codes (MAC) is utilized to provide authentication.
iii. Address-based also called identity-based method. - The location of the sending node is checked. This method is quite practical for WSNs when you compare with other mechanisms, unfortunately it doesn't provide a strong authentication mechanism since its inconsequential to spoof a sensor ID.

2.5.3 INTEGRITY

Integrity is basically confirmation of a message not being changed, tampered with or altered (Padmavathi & Shanmugapriya, 2009). On the message content a content digest is appended to provide integrity of content exchanged. On receipt of message by the receiving node content digest is checked to confirm that content digest computed and received digest are equal. Once confirmed to be equal or same then it's treated as a legitimate message. Hashing algorithms are used to create content digests (Uluagac et al., 2008).

There are several algorithms for hashing available and these algorithms do not usually require the keys presence unless designed specifically to work with keyed-hashing for-example Keyed-Hashing for message Authentication Code (HMAC) and Cipher-based Message Authentication Code (CMAC) (Uluagac et al., 2008).

Integrity service checks data staleness since some decisions for some applications depends on whether the data is recent or it's not. For- example, waters of a given territory can be protected

with sinks detonated mines. Message freshness and its accurate timing from the sensor nodes in this kind of application is critical (Uluagac et al., 2008).

Integrity service also is meant to provide a mechanism for recovery from any content that has been altered (Uluagac et al., 2008).

2.5.4 ACCESS CONTROL

This security service ensures that there is no unauthorized usage of any given resource in WSNs. For- example, the rights or privileges of sinks should not be permitted for other sensor nodes such as varying parameters for the network. Therefore each participant should play its role as deemed (Uluagac et al., 2008).

Access control is a security service which is quite challenging for WSNs. Practically there is normally one end point, that is, the base station (sink) where all network data is collected. Therefore, no other sensors should access any resource that may be resident on other nodes. However, in certain circumstances sensor nodes which act as source may be queried by other sensor nodes. In such special circumstances, access control policies should be implemented (Uluagac et al., 2008).

Access control policy is meant to prevent illegitimate nodes from accessing critical data. The implementation of access control policy is quite practical for hierarchical and cluster based WSNs (Uluagac et al., 2008).

2.5.5 NON-REPUDIATION

Non-repudiation security service ensures that a node cannot refute the messages it has sent (Rajkumar.et al., 2012). To offer non-repudiation service digital signature scheme (DSS), which utilizes encryption methods, can be used. DSS can use either symmetric or asymmetric encryptions (Uluagac et al., 2008).

When you use symmetric encryption the WSN may be in danger of another sensor masquerading as the sensor's original signature. On the other hand, using asymmetric encryption may be expensive. Basically non-repudiation service facilitates the approval by another entity for message sent or received in WSNs. Therefore, a legitimate node, such as the base station (sink) can offer the service (Uluagac et al., 2008).

2.5.6 AVAILABILITY

This is a security service that checks to see if a given node can utilize the resources and also if the network is available to communicate messages. The WSN can be endangered if the sink (base station) or cluster head fails. Therefore availability is crucial for a network to be operational (Padmavathi & Shanmugapriya, 2009).

The availability security service for WSNs has been looked at in-depth from the Denial-of-Service (DoS) type attacks dimension in addition, properties for connecting WSNs as concerns availability has also been studied in great length (Uluagac et al., 2008).

2.6 SECURITY CLASSES

(Noman, 2008) came up with practical security classes for WSNs. The security classes of sensor networks are defined through arrangement of all the groups in their possible orders. These groups are:

i. Level of security demand.

   Demand for security can be categorized as Medium or as High

ii. Structure for network and routing

The following are the possible categories;

Table 2.3 Network and routing categories

1.	Hierarchical and multi- hop
2.	Undefined and multi-hop
3.	Star and multi-hop
4.	Undefined and multi-path
5.	Hierarchical and multi-path

iii. Nodes mobility

Nodes Mobility can take either of these two values, mobile or immobile.

The three groups discussed assists in deriving the various security classes for WSNs applications. Moreover, their outputs indicate the different uniqueness of the security classes of WSNs and the way you arrange their different outputs shows various security classes for WSNs. When you combine all the groups' possible values, we derive all possible security classes of WSNs.

Fig 2.6 shows the security classes' definition framework;

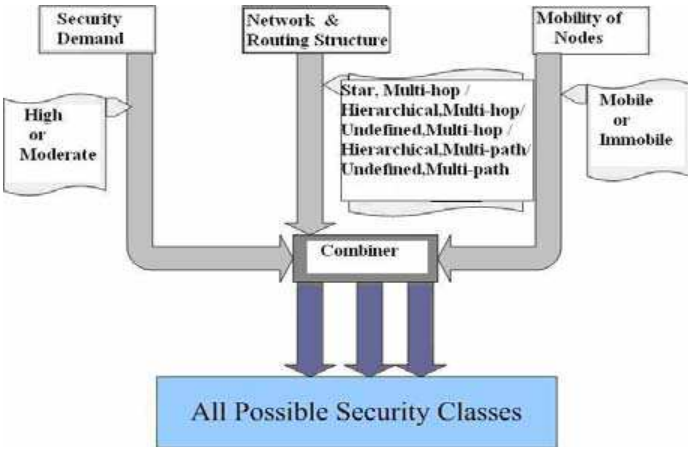


Figure: 2.6 security classes (Noman, 2008)

From Fig 2.6 above, we note that the possible security environments (or security classes) is derived from summing all combinations of these three groups. Therefore, when you multiply 2*5*2 you get twenty (20) security classes. It is imperative to note that, only five (5) possibilities of network and routing structures have been considered. Practically, we can have more than five (5) possibilities.

Fig. 2.6 above shows that all possible security classes can be derived when all the groups possible values are all identified. In addition, it shows the different security classes practically used with existing WSNs applications. It's important to note that a sensor application can identify all possible security protocols for its security class.

2.6.1 SECURITY CLASSES

For the purpose of getting just a few typical security classes, a few security classes from all the possible security classes have been taken. In addition, undefined network structure has been assumed. Considering the assumption the possible network and routing structure would be;

(undefined and multi-hop) and (undefined and multi-path). Table 2.3 below shows four (4) distinctive security environments (security classes) of WSNs.

Table 2.4: security classes

Environment name	**Security demands**	**Network and routing structure**	**Mobility**
WSN_Env_A	High	Multi-hop	Mobile
WSN_Env_B	Moderate	Multi-hop	Mobile
WSN_Env_C	High	Multi-path	Immobile
WSN_Env_D	Moderate	Multi-path	Immobile

2.6.2 SENSOR APPLICATION SECURITY CLASS

A WSN application that exists can locate the security class it belongs to. E.g., a WSN application that requires high security, (undefined and multi-path) network and routing structure and no support for nodes mobility, will belong to WSN_Env_C security class.

2.6.3 SECURITY CLASS FOR A SECURITY PROTOCOL

The security classes as shown in (fig 2.6) clearly express the security protocols requirements. E.g. WSN_Env_B security class requirements are moderate security demand, network and routing structure is (undefined and Multi-hop) and the support of nodes mobility from its security protocol. Looking at WSN_Env_B security class further its demand for security which is moderate, it can identify requirements such as; scalability robustness and availability.

SPINS, TinySec and LLSP are used for resource constrained environment and they provide confidentiality alongside authentication. However the protection against DoS attack is low,

furthermore they don't assure of protection against injection, alternation and node capture as shown in Table 2.2. Therefore we can conclude that SPINS, TinySec and LLSP offer moderate security. In addition they support mobility of nodes and they all support multi-hop structure. This means that these protocols fall under security class WSN_Env_B.

2.7 SIMULATION TOOLS

There are a number of simulators that are used to study WSNs, and they are often targeted to some area of WSN networks. The credibility of simulations performed in a chosen simulator is an important factor. Simulators are commonly used to develop and test WSNs protocols. Emulator is a tool which uses firmware and hardware to perform simulation (Fei, 2011).

This section will discuss different simulators used in WSN, showing their advantages and limitations. In addition the simulator chosen will be justified.

2.7.1 Network Simulator-2(NS-2)

NS-2 is an object oriented discrete-event and a general purpose simulator written in C++ with a TCL front-end for networking research. It's a free and open source and NS-2 is the simulator most commonly used (Kurkowski et al., 2005). It's not a specific network simulator, and it supports many protocols in all layers.NS-2 can be run on various operating systems such as Linux and windows over Cygwin.

Advantages

- Different protocols from all layers can be supported by NS-2.
- Its open source nature saves simulation costs.
- There are documents available online that user can improve their codes.

Disadvantages

- It uses both scripting language and technique for modeling which you have to be familiar with
- It has no graphical user interface (GUI)
- NS-2 is not specifically meant for WSNs and so it's limited in WSNs simulation.
- Simulating more than 100 nodes is a challenge (scalability issue).

2.7.2 TOSSIM

TOSSIM is an emulator that is designed specifically for use by WSN applications that run on Tiny Operating System (TinyOS). TinyOS is open source software that targets embedded operating system. TOSSIM was developed by UC Berkeley TinyOS project team and it's a bit-level discrete event network emulator built in python programming language. TOSSIM can be installed and run on operating systems such as Linux and Windows (Fei, 2011).

Advantages

- The nature of TOSSIM is open source software.
- It's built with a Graphical User Interface (GUI).
- Scalability-it offers support for many nodes.
- The documents are freely available online.

Disadvantages

- It can't simulate performance of new protocols.
- It emulates applications that are only homogeneous
- It can only simulate motes.

2.7.3 EmStar

EmStar is designed specifically WSNs and are developed in C programming language. University of California, in Los Angeles was involved in its initial development. It runs in real time and is also trace-driven (Fei, 2011).

Advantages

- Documents are available online.
- It‘s robust i.e. All faults with the sensors can be mitigated
- Each EmStar module can be run separately and still reuse the software.
- It has a Graphical User Interface (GUI).

Disadvantages

- It offers no support for simulation of very many sensors.
- The scalability factor is limited.

2.7.4Objective Modular Network Testbed in C++ (OMNeT++)

It’s an open source, component-based network simulator with Graphical user interface support (Jianlin, 2008). It’s developed with C++ and can installed and run on different operating systems such as windows and Linux.

Advantages

- It has Graphical User Interface (GUI) - it therefore becomes easy to Trace and debug.
- It supports Medium Access Control (MAC) protocols and other WSN localized protocols.
- consumption of power challenges can be simulated for WSNs

Disadvantages

- It doesn't have many protocols.
- Integration of separate models is a big challenge.

2.7.5 Java Simulator (J-Sim)

Java Simulator is a discrete event network simulator developed using Java. The major domains of use are in physiology and biomedicine, but also used for WSNs simulation. J-Sim has the ability to simulate real-time processes (Fei, 2011).

Advantages

- It has many protocols
- Reusable and interchangeable models, which ease simulation process.
- It simulates up to a maximum of 500 nodes
- It facilitates simulation of radio channels and power consumptions in WSNs.
- It has Graphical user interface (GUI) library-It assists in tracing and debugging programs.

Disadvantages

- In comparison to NS-2, time for execution is longer.
- It's a great challenge to add new protocols.

2.7.6 ATEMU

This is an emulator of a processor called AVR(microcontroller) and it's developed using C programming language. It can be installed and run on the following operating system; Solaris and Linux (Fei, 2011).

Advantages

- Multiple sensor nodes can be simulated simultaneously, with each sensor node running different programs.
- It has a Graphical User interface (GUI) that assists in debugging programs, and check executions of a program.
- It's open source software hence saving the simulation costs.
- The documents are available online

Disadvantages

- It takes longer to simulate in comparison to other simulators.
- It has fewer functions for simulation of routing and clustering problems.

2.7.7 Avrora

This simulator is designed specifically for WSNs and was developed in Java programming language. It can simulate AVR (microcontroller) MICA2 sensor nodes just like ATEMU. University of California in Los Angeles developed it (Fei, 2011).

Advantages

- The speed of simulation is fast and offers better scalability.
- It supports numerous nodes for simulation
- Diverse programming code projects can be simulated unlike TOSSIM which supports only TinyOS simulation.

Disadvantages

- It has no Graphical User Interface (GUI).

- The algorithms for network management can't be simulated since it doesn't provide tools for network communication.

2.7.8 Global Mobile system Simulator (GloMoSim)

It's a library-based sequential and parallel simulator for wireless networks. It is designed as a set of library modules, each of which simulates a specific wireless communication protocol. The library has been developed using PARSEC, a C-based parallel simulation language (Xiang, Rajive & Mario, 1997)

Advantages

- It can simulate parallel format.
- It was built specifically for wireless networks
- The visualization tool is available

Disadvantages

- It only deals with networks that are IP based.
- There are no new protocols included in it.

2.7.9 SENSE

SENSE is a simulator that is specific to sensor network, in addition it is a component based simulator and it was developed in 2004 using C++ programming language Chen et al. (2006).

Advantages

- It considers modeling methodology and it's efficient in simulation process.
- It utilizes memory efficiently and it's fast, extensible and reusable.

Disadvantages

- WSN research evaluation is not accurate.
- The set of models available are not comprehensive.
- It lacks a visualization tool.

2.7.10 Optimized Network Engineering Tools (OPNET Modeler)

OPNET Modeler is a discrete event, object oriented, and general purpose network simulator (Korkalainen M. et al., 2009). OPNET was first proposed in 1986 and was developed using C++ programming language by Massachusetts Institute of Technology (MIT) in 1987.

Advantages

- It's a fast discrete event simulator.
- It has an inbuilt Graphical User Interface (GUI) which assists in debugging and analysis process.
- It supports scalability in its simulations.
- It has plenty of component libraries with source code.
- It has a comprehensive manual.

Disadvantages

- It's commercial software.
- Developing a certain component is quite a challenge.

2.8 CONCLUSION

The comparison of the various simulation tools was to provide my research with a specific tool that would benefit this research. The different simulators discussed have their own advantages and disadvantages and this shows that no single tool has all the features needed. It's good to note that NS-2, OPNET Modeler and OMNeT++ are the tools majorly used and the results they generate are universally acceptable by academia world.

When you compare NS-2 with OMNeT++ then we can derive a conclusion that NS-2 has numerous base in terms of users. OPNET Modeler is as good as NS-2 since it's the results generated are quite similar. However OPNET Modeler presents lots of features as compared to NS-2. The focus of OPNET Modeler is mainly to researchers in the industries and to those who require extensive set of built-in reliable models for constructing credible simulations in a quick way, rather than academic researchers.

NS-2 is used by most of the researchers and most of the problems are solved by the research community through the use of forums and mailing lists. However it's criticized for its complicated architecture.

This research used NS-2 due to its popularity in scientific and in the world of academia and also due to the fact that it's open source.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 INTRODUCTION

A set of tools, techniques, methods, and documentation which assists a researcher to realize the objectives set for research is referred to as research methodology. This chapter focuses on methods used for evaluation of wireless sensor networks, proposed methodology and tool used.

3.2 EVALUATION OF CURRENT METHODOLOGIES

There are three methods used in integration of WSNs security protocols for evaluation namely;

- Analytical method
- Direct Application or Physical measurement
- Simulations

Analytical Method- WSNs security protocols can be mathematically modeled and parameterized. The security protocols have parameters that considerably affect WSNs consumption of resources. In comparison to simulation method analytical solutions offer lesser accuracy, but are less costly and consume lesser time (Sklenar, 2000).

Direct Application-This means the actual construction of WSN and implementing a given security protocol. It uses experimental results that have the highest probability to achieve a replica of what is expected in a real world situation. However security protocols implementation for large scale networks is costly and consumes a lot of time (Sklenar, 2000)

Simulation- Using this method you can implement the various security protocols available. This is mainly done with software's. There is motivation for simulation which includes: it's more economically advantageous in the sense that it doesn't have to use expensive equipment,

complex scenarios can be easily tested, results can be easily obtained and it works under controlled experimental conditions.

Simulations can perform many experiments and analyze results using a model created at design stage; however a disadvantage in simulating is that real systems are complex to model. In addition in simulation there could be possibility of simulation errors and may not guarantee accurate representation of WSNs operations (Sklenar, 2000).

3.3 PROPOSED METHODOLOGY

This dissertation used simulation methodology. The techniques used in this study include literature review and simulation experiments. The use of a given method of evaluation can significantly determine the results. These methodologies are different in terms of accuracy, cost, and required time. In consideration of these parameters, simulation was the best suited method for this research. Direct application was eliminated as a choice in this research since it's very expensive and takes a lot of time to implement. Analytical solutions was also ruled out due to the fact that it's less accurate in comparison to simulation methodology, although they cost less and consume less time to implement. Simulation costs are negligible since the NS-2 software required is open source.

3.4 SIMULATION TOOL

Network simulators are basically used in simulation of network behaviors using a specific tool of simulation known as a simulator. The process of simulation makes the whole system functional in a hypothetical manner using a simulation tool.

This research dissertation used the latest release of NS-2 which is ns2.35. Ns2.35 installation was done on windows7 over cygwin. Object oriented tool command language (Otcl) scripts were

written to setup and run a simulation, OTcl scripts are programs for simulation that perform the following; initiation of an event scheduler, setting up topologies for the network and via event scheduler it informs sources of traffic when to begin and end packets transmission.

NS-2 is an OTcl interpreter that receives OTcl script to set the parameters for environment in accordance to the received script. Results for simulation are generated in a single or multiple text based output files (trace files) as soon as the process for simulation is completed. The trace files have the full details of simulation data, which is normally used for direct analysis or with Network Animator (NAM) which is a graphical user interface.

Figure 3.1 depicts NS-2 structure

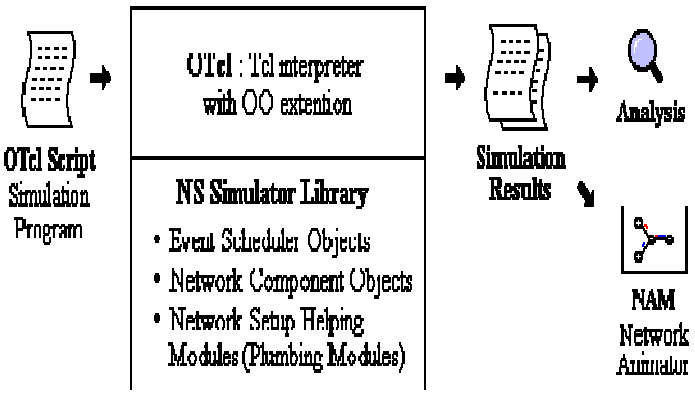


Figure 3.1: NS-2 Logical design

CHAPTER 4: CONCEPTUAL DESIGN

4.1 INTRODUCTION

This chapter describes the conceptual model that was implemented for simulation and hardware and software environment used for simulation purposes.

4.2 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 4.1 below shows the conceptual model that was used to implement security protocols.

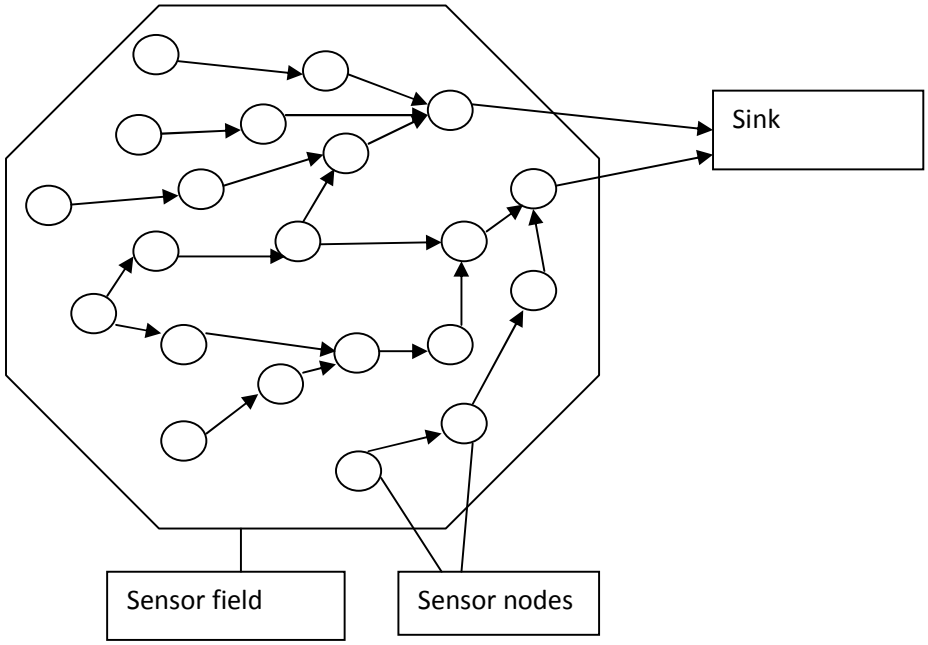


Figure: 4.1 Conceptual model

4.3 SENSOR NODE ARCHITECTURE

Figure 4.2 below describes the sensor node architecture. It has been used to explain the model (Figure 4.1) that was implemented for simulation of security protocols.

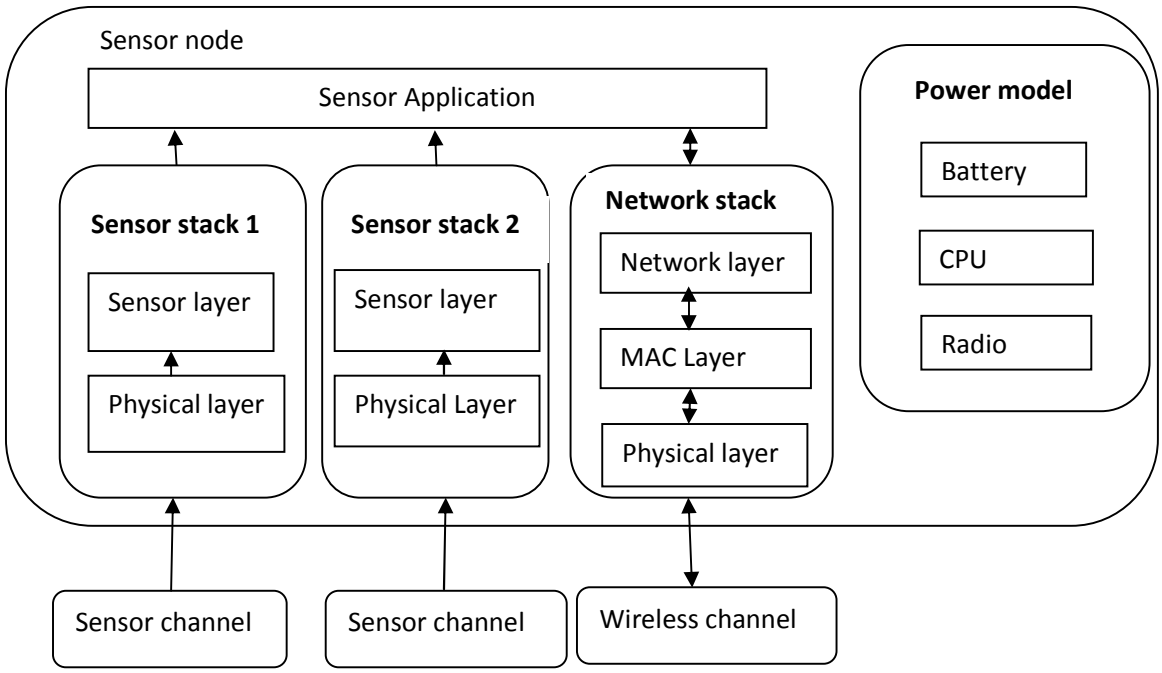


Figure: 4.2 Sensor node architecture (Sriporamanont,T. & Liming,G.,2006)

The function of sensor protocol stack is to detect and process the data detected and forward it to the application layer.

The function of application layer is to process and transmit data to the user node in the form of sensor reports through the wireless channel.

Power model consisting of battery (energy provider) and CPU and Radio (energy consumers) is key part of the sensor node.

4.4 HARDWARE AND SOFTWARE ENVIRONMENT

Table 4.1: Hardware and software Specifications

	Item Description	**Specification**
1	**Hardware**	Laptop,2.00GB,Intel(R) Core i3 2.30GHz,64-bit
2	**Operating system**	Windows 7 Home Basic Linux Emulator-Cygwin
3	**Network Simulator**	Ns-2 version 2.35

CHAPTER 5: PRESENTATION AND DISCUSSION OF FINDING

5.1 INTRODUCTION

This chapter describes the implementation of security protocols, simulation experiments and findings on energy consumption of security protocols.

5.2 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 mannasim framework was installed.

5.3 SIMULATION EXPERIMENTS

The following assumptions were made for the purpose of simulation

i. The nodes are homogenous
ii. The nodes have uniform energy of 10 Joules initially
iii. The nodes are distributed randomly.
iv. The nodes are immobile
v. The same Packet size for the nodes.

Figure 5.1 shows the simulation setup

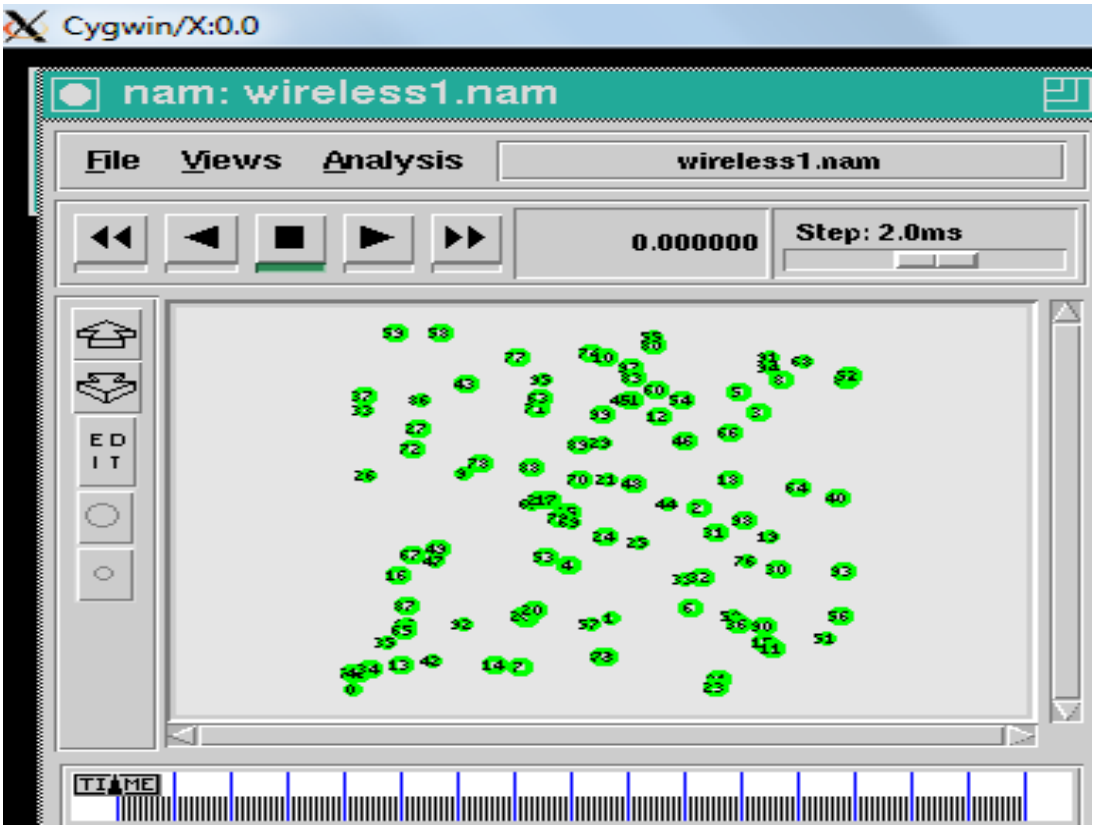


Figure: 5.1 Field topography

Table 5.1 below shows the simulation parameters:

Table 5.1: Simulation parameters

	ITEM DESCRIPTION	**SPECIFICATION**
1	Simulation Field	1000m X 1000m
2	Channel type	Channel/wireless channel
3	Radio propagation model	Two ray ground
4	Number of nodes	100
5	Antennae model	Antenna/Omniantenna
6	Energy model	Battery
7	Link layer type	LL
8	Type of network interface	Phy/wirelessphy
9	MAC protocol	Mac/802_11
10	Type of Interface queue	Queue/Drop tail/priqueue
11	Simulation period	30 Seconds

The simulation parameters shown in table 5.1 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 model** - 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.

5.4 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 e 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 5.2 and figure 5.3 show the average energy amount in each sensor node at different time intervals.

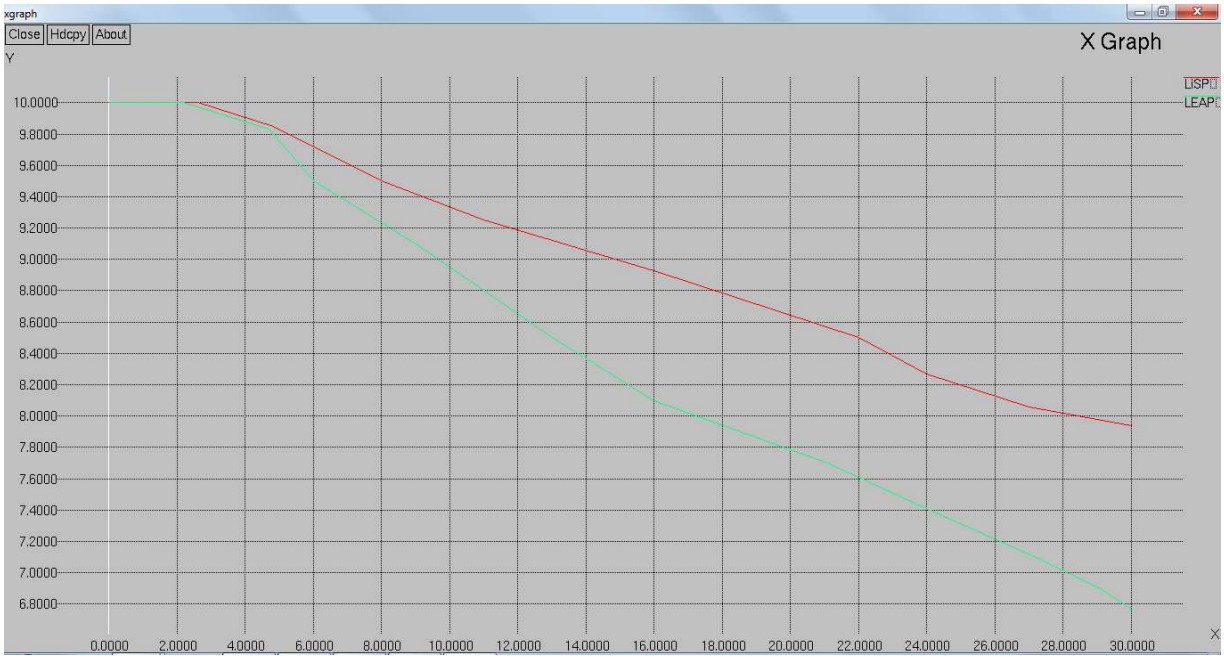


Figure 5.2: Energy consumption in WSN_ENV_A

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 looses 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 5.2: WSN_ENV_A energy remaining

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

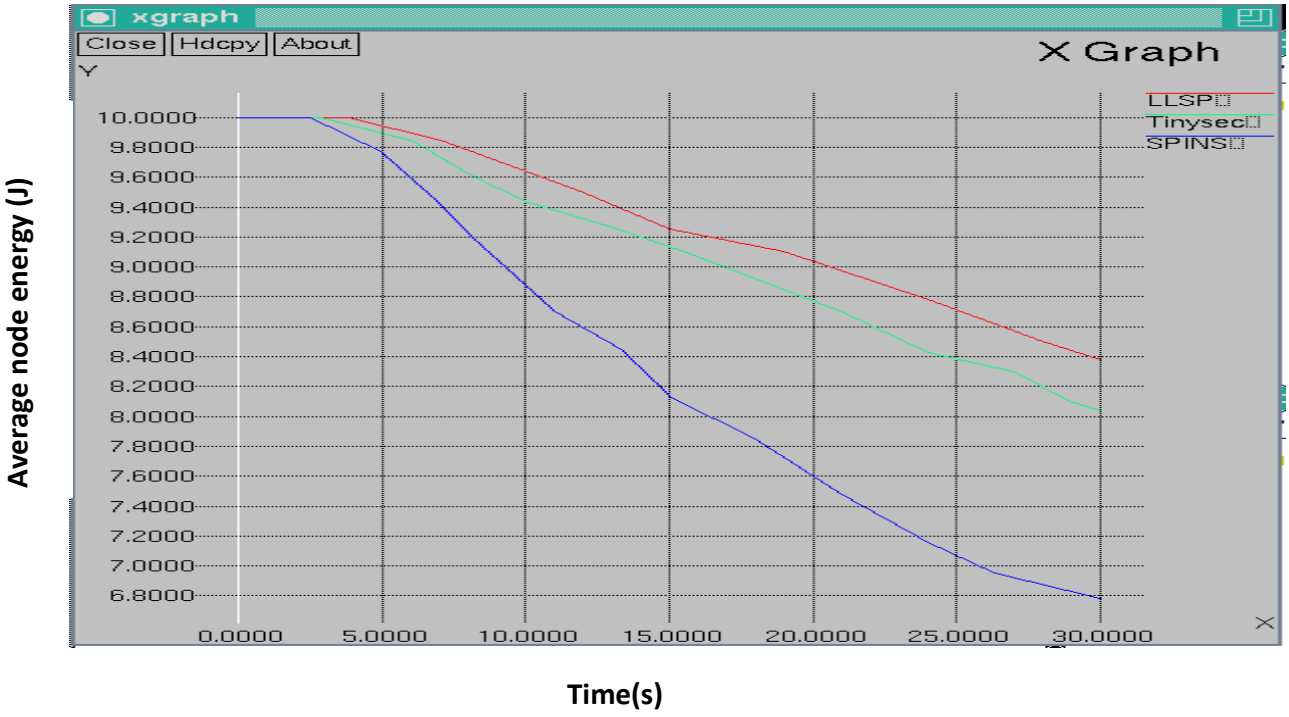


Figure 5.3: Energy consumption WSN_ENV_B

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 looses after approximately 3.7s and LLSP looses 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 5.3:WSN_ENV_B energy remaining

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

5.5 CONCLUSION

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

CHAPTER 6: CONCLUSIONS AND FURTHER WORK

6.1 INTRODUCTION

This research dissertation, introduced energy metric in security classes to determine best security protocol. This chapter shows the accomplishments, challenges and recommendations and suggestions for future work.

6.2 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 of determining selection of best security protocol, by first of all categorizing the security protocol into its security class and then comparing protocols in same class, and finally get the best in a given security class based on energy metric. The result of simulations conducted for protocols considered in security class WSN_ENV_A, showed that LiSP security protocol is the best protocol in terms of energy efficiency, and for security class WSN_ENV_B results showed that LLSP security protocol is the best protocol in terms of energy efficiency

6.3 ACCOMPLISHMENTS

- Implemented SPINS, TinySec, LLSP, LiSP and LEAP protocols in NS-2
- Tool Command Language was used successfully
- Skills on how to simulate protocols using NS-2 was acquired.

6.4 CHALLENGES

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

6.5 FUTURE WORK

NS-2 is the most popular simulator in networks, unfortunately it's not originally designed for wireless sensor networks and so I suggest use of a simulator such as OPNET Modeler, which was initially designed for simulation of WSNs parameters, and provides more features than NS-2.

This research proposes all security protocols falling in same security class to be compared not only for energy efficiency, but also other metrics to be considered such as throughput, latency, and jitter.

REFERENCES

Karlof, C., Sastry, N., and Wagner, D., 2004.TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA*, pp. 162-175.

Perrig, A. et al., 2002. Security Protocols for Sensor Networks, *Wireless Networks*, Vol. 8(No. 5), pp. 521-534.

Lightfoot,L. E.,and Jian R. and Tongtong L. 2007.An Energy Efficient Link-Layer Security Protocol for Wireless Sensor Networks. *IEEE EIT Proceedings.* pp 233-238

Anjali, S. et al. 2011.A new approach for Evolution of end to end Security in Wireless Sensor Network, *International Journal on Computer Science and Engineering (IJCSE)* ,Vol. 3(No. 6), pp 2531-2543

Taejoon P. and Kang G. 2004.LiSP: A Lightweight Security Protocol for Wireless Sensor Networks .*ACM Transactions on Embedded Computing Systems*, Vol. 3, (No. 3).

Jiyong J., Taekyoung K., and Jooseok S. 2007.A Time-Based Key Management Protocol for Wireless Sensor Networks. pp. 314–328,.

Sencun Z, Sanjeev S. and Sushil J. 2004.LEAP: Efficient Security Mechanisms for Large Scale Distributed Sensor Networks Technical report.

L. Gheorghe et al. 2010.Reliable Authentication and Anti-replay Security Protocol for Wireless Sensor Networks.*The Second International Conferences on Advanced Service Computing.*

Anupma S. et al., 2011.A Review of various security protocols in Wireless Sensor Network ,*Int. J. Comp. Tech. Appl*., Vol 2 (4), pp790- 797.

Boyle, D., and Newe, T., 2007.Security protocols for use with wireless sensor networks: A survey of security architecture, *Proceedings of the Third International conference on wireless and mobile communications*, p. 54.

Abu, S., 2009. An Evaluation of Security Protocols on Wireless Sensor Network. *Seminar on Internetworking.*

Taejoon, P. and Kang, G. 2004. LiSP: A Lightweight Security Protocol for Wireless Sensor Networks. *ACM Transaction*.

Gaurav, S., Suman, B., and Anil K. 2012. Security Frameworks for Wireless Sensor Networks-Review.*2nd International Conference on Communication, Computing & Security [ICCCS-2012]*.

Krzysztof, D. and Ewa, N. 2012. A Survey of Energy Efficient Security Architectures and Protocols for Wireless Sensor Networks. *Journal of Telecommunication and information technology*.pp 64-72

Adrian, P., John, S., *and* Wagner, D.2004 .Security in wireless sensor networks.*Communications of the ACM June 2004*,vol. 47(no. 6) pp 53-57.

Shiqun, L.et al.2007. Efficient Link Layer Security Scheme for Wireless Sensor Networks. *Project supported by the National Nature Science Foundation of China* .Shanghai Jiaotong University.

Kui, R.,Wenjing, L.,and Yanchao, Z. 2007. LEDS: Providing Location-aware End-to-end Data Security in Wireless Sensor Networks.

Zeenat, R. et al. 2010.,SPIN Implementation in TinyOS Environment using nesC. *Second International conference on Computing, Communication and Networking Technologies*.

Aseri, T.C., Singla, N..2011.Enhanced Security Protocol in Wireless Sensor Networks *International. Journal of Computers, Communications & Control*. Vol. VI (2011), No. 2 (June), pp. 214-221

Padmavathi, G., and Shanmugapriya, D.2009.A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks.*International Journal of Computer Science and Information Security.(IJCSIS),* Vol. 4(No. 1 & 2).pp 1-9.

Rajkumar, Sunitha,K.R.,and Chandrakanth, H.G.2012.A Survey on Security Attacks in Wireless Sensor Network.*International Journal of Engineering Research and Applications (IJERA),*Vol. 2(Issue 4), pp.1684-1691

Uluagac et al.2008.Designing Secure Protocols for Wireless Sensor Networks. *Third International Conference, Wireless Algorithms, Systems, and Applications (WASA)* Proceedings. pp 503-514

Luis, E., and Antonio, J. 2008. Security in Wireless Sensor Networks.

Werle, C. et al.2008. Decision process for automated selection of security protocols,*IEEE*.

Noman, A.N.M. 2008. A generic framework for defining security environments of Wireless Sensor Networks. *Electrical and Computer Engineering, 2008. ICECE 2008.* doi: 10.1109/ICECE.2008.4769344

David Boyle and Thomas Newe.2008.Securing wireless sensor networks: Security architectures, *Journal of networks*, vol.3(No.1).

Miloš Jevtic and Nikola Zogovic, 2009. Evaluation of Wireless Sensor Network Simulators.*17th Telecommunications forum TELFOR*, pp 1303-1306

Murat, M. K., 2008. A Survey of Network Simulators Supporting Wireless Networks.

Fei Yu,2011. A survey of wireless sensor network simulation tools.URL: http://www1.cse.wustl.edu/~jain/cse567-11/index.html

Egea, E. et al.2005. Simulation Tools for Wireless Sensor Networks.*Summer Simulation Multiconference - SPECTS* 2005

Xiang, Z.,Rajive, B. and Mario G.1997.GloMoSim:A Library for Parallel Simulation of Large-scale Wireless Networks.University of California,Los Angeles.

Cheng, L., Zhang, X. & Bourgeois, A. G. 2006. Ieee 802.15.4 simulation module in network simulator gtnets, Proc. *VTC 2006-Spring Vehicular Technology Conf.* IEEE 63rd, Vol. 3, pp. 1308–1312.

Kurkowski, S., Camp, T. & Colagrosso, M. 2005. MANET simulation studies: the incredible.*Mobile Computing and Communications Review* 9(4): pp 50–61.

Korkalainen M. et al.2009.Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications. *Fifth International Conference on Networking and services,IEEE* pp102-106.

Sharma, R.,Chaba, Y. and Singh, Y. 2010. Analysis of Security Protocols in Wireless Sensor Network.*International Journal Advanced Networking and Applications.* Volume.02, Issue: 03, pp707-713.

Jianlin, p.2008.A Survey of Network Simulation Tools:Current Status and Future Developments.Available online at http://www.cse.wustl.edu/~jain/cse567-08/index.html

Sushma, Deepak Nandal and Vikas Nandal. 2011.Security Threats in Wireless Sensor Networks. *International Journal of Computer Science & Management Studies*, Vol. 11, Issue 01.PP59-63

T. Zahariadis et al.2009.Securing Wireless Sensor Networks Towards a Trusted “Internet of Things”. *Towards the Future Internet.IOS Press*,doi:10.3233/978-1-60750-007-0-47.pp 47-56

Sklenar,J.:Simulation(University of Malta,2000)

Sriporamanont,T. & Liming, G.,2006.Wireless Sensor Network Simulator.Technical report.

NAM: Network Animator,URL: http://www.isi.edu/nsnam/nam/ [accessed April 2013].

Cygwin User’s Guide, URL: http://www.cygwin.com/ [accessed April 2013].

M.Greis, “Tutorial for the Network Simulator NS”, URL: http://www.isi.edu/nsnam/ns/tutorial/index.html [accessed April 2013].

The Network Simulator ns-2 documentation URL: http://www.isi.edu/nsnam/ns/ns documentation.html [accessed April 2013].

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:

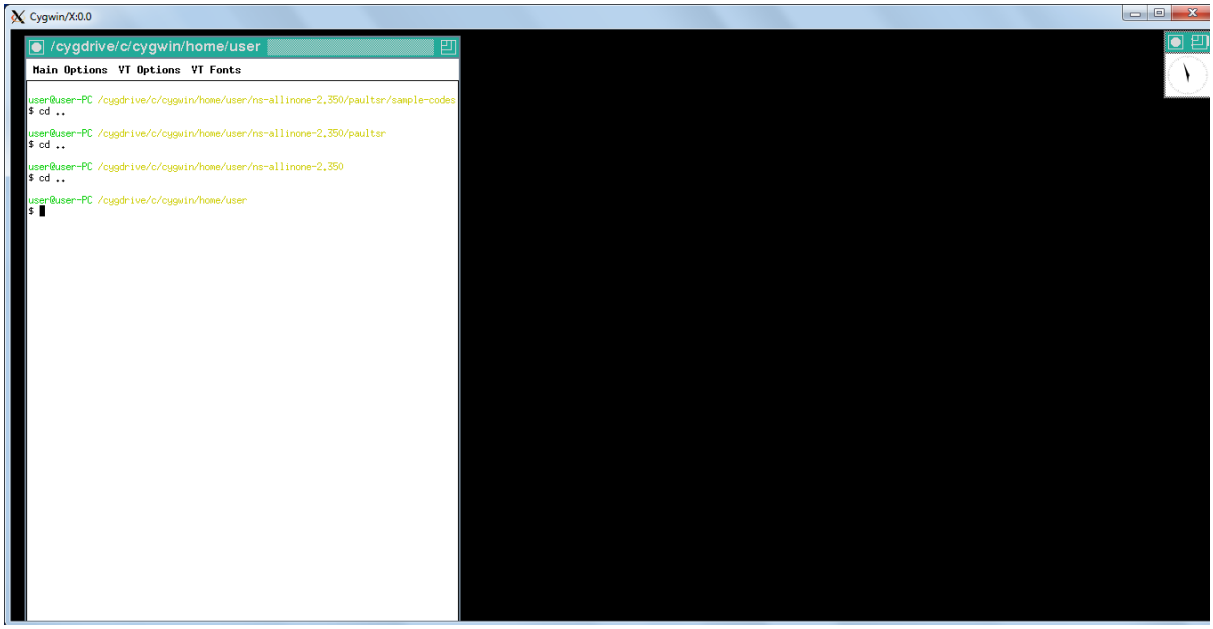
$NS_HOME/tk8.5.10/unix:$NS_HOME/bin:$PATH

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_PATH

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)



Appendix 2

TCL Script code

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)