

**SECURE, ENERGY EFFICIENT
ROUTING ALGORITHM FOR
SENSOR NETWORKS**

By

MALLESWARI AKKINENI

Bachelor of Technology

Jawaharlal Nehru Technological University

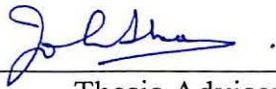
Hyderabad, India

2001

**Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2004**

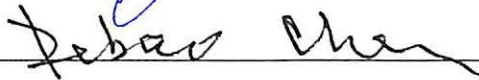
SECURE, ENERGY EFFICIENT
ROUTING ALGORITHM FOR
SENSOR NETWORKS

Thesis Approved:



Thesis Adviser





Dean of the Graduate College

ACKNOWLEDGMENTS

I express my extreme gratitude to my graduate advisor and committee chair, Dr. Thomas Johnson for his continuous guidance, academic supervision, research ideas, help and support. I also thank him for giving me a great opportunity to work on a NASA funded research project, which has been a milestone in my career. I also thank Dr. Debao Chen and Dr. Ajith Abraham for serving on my thesis committee and providing valuable suggestions and support.

I convey my special appreciation to my parents, my brother and my friends who have always been great moral support.

My appreciation extends to the United States, the Oklahoma State University and the Department of Computer Science for providing outstanding environment, resources and research facilities for my education at Masters level.

TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION.....	1
1.1 Sensor Networks.....	1
1.2 Energy Limitations.....	2
1.2 Security Issue.....	2
2. LITERATURE REVIEW.....	4
2.1 Background.....	4
2.2 Power-Aware Routing.....	6
2.2.1 Flooding.....	6
2.2.2 Gossiping.....	7
2.2.3 Energy-Efficient Routes.....	7
2.2.4 Power Aware Routing Protocol.....	10
2.2.5 Alternate Path Routing Protocol (APR).....	11
2.2.6 Localized Energy Aware Routing Protocol.....	11
2.2.7 Flow Augmentation Routing Protocol.....	11
2.2.8 Online Max-Min Routing Protocol (OOM).....	12
2.2.9 Power-Aware Routing Protocol.....	12
2.2.10 Span Protocol.....	12
2.2.11 Geographic Adaptive Fidelity Protocol (GAF).....	13
2.2.12 Data-Centric Routing.....	13
2.3 Secure Routing.....	15
2.3.1 Multi-Tiered Security Architecture.....	15
2.3.2 SPINS (SNEP AND μ TESLA).....	16
2.3.3 Secure Routing – Attacks And Counter Measures.....	19
3. THESIS STATEMENT.....	30
4. PROPOSED SOLUTION.....	31
5. SIMULATION AND RESULTS.....	40
5.1 BRITE.....	40
5.2 Format of BRITE output.....	41

5.3 System attributes.....	43
5.4 Graphs.....	44
5.5 Results.....	55
6. CONCLUSIONS.....	57
6.1 Conclusions.....	57
6.2 Future work.....	57
REFERENCES.....	58

LIST OF FIGURES

Figure	Page
1. Sensor Network.....	4
2. Energy Efficient Routes.....	8
3. Data Aggregation.....	14
4. Flat topology parameters for BRITE.....	41
5. Sample BRITE output.....	42
6. Energy Consumption - Flooding versus proposed algorithm – Level 1.....	44
7. Simulation Output Data of Power consumption – Level 1.....	45
8. Energy Consumption - Flooding versus proposed algorithm – Level 2.....	45
9. Simulation Output Data of Power consumption – Level 2.....	46
10. Number of hops on a route comparison – Level 1.....	49
11. Number of hops on a route comparison – Level 2.....	49
12. Number of routes used – A comparison	50
13. Number of routes used by the proposed algorithm	51
14. Adversary trying to create a faulty link between two nodes.....	53
15. Visibility of sensor message packets to the adversary – 1 adversary.....	54
16. Visibility of sensor message packets to the adversary – 2 adversaries.....	54

NOMENCLATURE

GPS	Global Positioning System
MAC	Medium Access Control
SPINS	Security Protocols for Sensor Networks
SNEP	Sensor Network Encryption Protocol
MAC	Message Authentication Code
K_{mac}	Key for message authentication code
K_{encr}	Key for message encryption

Chapter 1

INTRODUCTION

1.1 Sensor Networks

A *sensor* is a small measuring device that can monitor some physical phenomenon (like atmospheric gas content) and send the information to a destination that collects the data. A *Sensor Network* is a wireless system containing tiny sensors and actuators and may have thousands of low-power, low-cost nodes comprised of these sensors/actuators. These networks can be heterogeneous in the sense that different sensors may be capable of observing and reporting on various dynamic properties of their terrain. Nodes may also be mobile or stable.

Sensor networks have been proposed for military surveillance and environmental monitoring applications. They are rapidly emerging as an important new area in the research community. The main characteristic of Sensor Networks is that their nodes are untethered and unattended. These small sensor nodes could be deployed in industries such as transportation, health care, disaster recovery, warfare, security, and even space exploration. By connecting these small sensor nodes by radio links, the sensor nodes could perform tasks which traditional single sensor nodes are hard to match, e.g. detecting danger spots in a disaster area. Sensor networks are similar to Ad-hoc networks

in various aspects though they differ in some. Due to numerous constraints in Sensor Networks, they have various technical issues that are still being researched.

1.2 Energy Limitations

Each node in a sensor network may consist of one or more sensors, a low power radio, a portable power supply and possibly localization hardware or a GPS unit or some ranging device. Consequently, Sensor Networks are severely constrained by computational and energy resources. Also, because they are unattended and untethered in nature, these resources are practically non-replenishable.

A routing protocol does the job of discovering routes between nodes. While establishment of correct and efficient routes is the primary goal, another challenging goal is to provide energy efficient routing protocols. This is important because each node has an operation time depending on its battery capacity and the lifetime of a node directly effects the lifetime of a network.

1.3 Security Issue

Sensor Networks are increasingly important in diverse areas, where the information transmitted is highly sensitive and should be protected from intrusions. The resource limitations of sensor networks, also makes them vulnerable to security attacks. Resource intensive computations such as complex cryptographic functions and resource utilization for lengthy and extensive communications are not feasible for sensor networks. Therefore the potential for intrusion, eavesdropping and other security attacks are greater than for typical wireless networks.

Thus, a need for securing the information is also a vital issue to be considered in Sensor Networks. The key to effectively protect a data transmission network against a wide range of attacks is a suite of mechanisms spanning several layers of the protocol stack. *Routing* is one area at network layer where such additional protection can be achieved by supplementing other security measures.

Given the problems with sensor networks identified above, this thesis focuses on achieving power savings and security at the same time in routing protocols for sensor networks. Recent work has investigated both these issues from various viewpoints. This thesis is limited to the network layer routing solution to the problem.

The organization of this Thesis is as follows: Chapter 2 provides a background on various topics relevant to the Thesis problem and the related work performed in those areas. The thesis statement in Chapter 3 lists the objectives of this work. Chapter 4 proposes a solution/algorithm proposed for the thesis problem and explains it in detail. Chapter 5 shows all the simulations, describes the tools and system attributes used and elaborates the results. Chapter 6 concludes the thesis and provides pointers for future work.

Chapter 2

LITERATURE REVIEW

2.1 Background

Figure 1 depicts a sensor network and its connection to a data processing center:

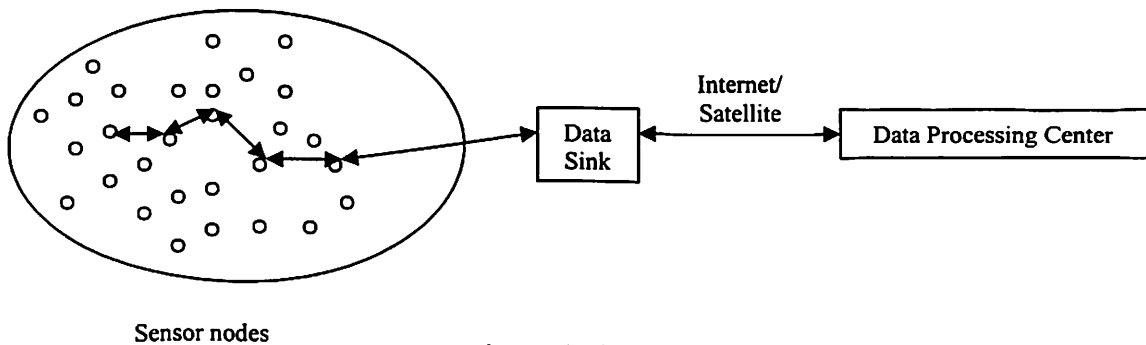


Figure 1: Sensor Network

Each sensor node typically consists of a sensing unit, a processing unit, a transceiver unit and a power unit. It may also have additional application dependent components such as a location finding system, power generator and mobilizer. The protocol stack associated with sensors and sink nodes consists of the physical layer, data link layer, network layer, transport layer and application layer. All these layers span across three major issues: power management, mobility management and task management [12].

Traditionally, effective power management to extend battery life is a critical issue in mobile computing (for example, Ad-hoc networks). In sensor environments, the difficulty of replacing sensor batteries exacerbates the problem. Power in sensors is used for more than one purpose: sensing, computing, and communication. Although the energy consumption rate for each type of operation is sensor and application specific, research has shown that a sensor node's energy consumption for communication activities substantially exceeds that required for sensing and computation. Thus, power aware routing is a right direction to take, since it concentrates on power savings at the communication level. It is also important to note the replacing sensor batteries in the field is not a feasible option in many if not most applications.

PlanetAnalog.com - an EE Times online community, which is the resource for analog and mixed/signal technology features up-to-the-minute news; exclusive user interviews, feature articles, commentary; and a regular online poll [10]. This website has stated in one of its headline articles on July 14, 2003 [11] as follows: "Against the backdrop of the war on terrorism, an expanding group of government researchers is at work on a nationwide sensor network that someday could provide a real-time early-warning system for a wide array of chemical, biological and nuclear threats across the United States" and much more on it.

Securing sensor networks should span multiple levels of the protocol stack, including the network routing level. Some security mechanisms can be applied at lower levels like the Medium Access Control (MAC) protocol which can be combined with techniques at higher levels thus attaining an integrated security scheme. Some of the

ideas that have been proposed and research in this area will be discussed briefly in the following sections.

2.2 Power-Aware routing

There are numerous routing algorithms and protocol ideas from traditional wired network systems and Ad-hoc systems. Some of them can be adapted for consideration for deployment in a sensor network. However, most of these protocols are more applicable for power aware routing than secure routing. Some of these protocols have been employed in a sensor network [2] [12]. Moreover, various security schemes, routing techniques and protocols are being proposed for sensor networks continuously as this is a new and rapidly developing area of technology. The rest of this chapter describes various approaches to routing and security in sensor network.

2.2.1 Flooding:

Flooding is an old technique that can also be used for routing in sensor networks. In flooding, each node receiving a data or management packet and repeats it by broadcasting, unless a maximum number of hops for the packet is reached or the destination of the packet is the node itself. Flooding is a reactive technique, and it does not require costly topology maintenance and complex route discovery algorithms. However, it has several deficiencies such as [9]:

- *Implosion*: Implosion is a situation where duplicated messages are sent to the same node. For example, if sensor node A has N neighbor sensor nodes that are also the

neighbors of sensor node B, sensor node B receives N copies of the message sent by sensor node A.

- *Overlap*: If two nodes share the same observing region, both of them may sense the same stimuli at the same time. As a result, neighbor nodes receive duplicated messages.
- *Resource blindness*: The flooding protocol does not take into account the available energy resources. Also, sending of messages redundantly due to the above two reasons means unnecessary depletion of energy from the sensor nodes. An energy resource aware protocol must take into account the amount of energy available to them at all times. This is an important consideration for sensor networks.

2.2.2 Gossiping [7]:

A derivation of flooding is gossiping in which nodes do not broadcast but send the incoming packets to a randomly selected neighbor. A sensor node randomly selects one of its neighbors to send the data. Once the neighbor node receives the data, it randomly selects another sensor node. Although this approach avoids the implosion problem by just having one copy of a message at any node, it takes a long time to propagate the message to all sensor nodes. Eventually a node with a route to the destination may be reached. The gossiping can stop at this point. Although this approach is slow, it is energy efficient.

2.2.3 Energy-efficient routes [12]:

There are three major issues involved in **energy aware routing** protocols [2]:

- 1) The goal is to find the path that either *minimizes* the absolute power consumed or *balances* the energy consumption of all mobile nodes. Balanced energy consumption

does not necessarily lead to minimized energy consumption, but it keeps a certain node from being overloaded and thus ensures longer network lifetime. Energy balance can be achieved indirectly by distributing network traffic [2].

- 2) Energy awareness has been implemented at the routing layer with help from other layers such as MAC or application layer. Information from MAC layer is beneficial because it usually supports power saving features that the routing protocols can exploit to provide better energy efficiency [2].
- 3) Some routing protocols assume that transmission power is controllable and nodes' location information is available (e.g., via GPS – Global Positioning Systems). Under these assumptions, the problem of finding a path with least consumed power becomes a conventional optimization problem on a graph where the weighted link cost corresponds to the transmission power required for transmitting a packet between two nodes of the link [2].

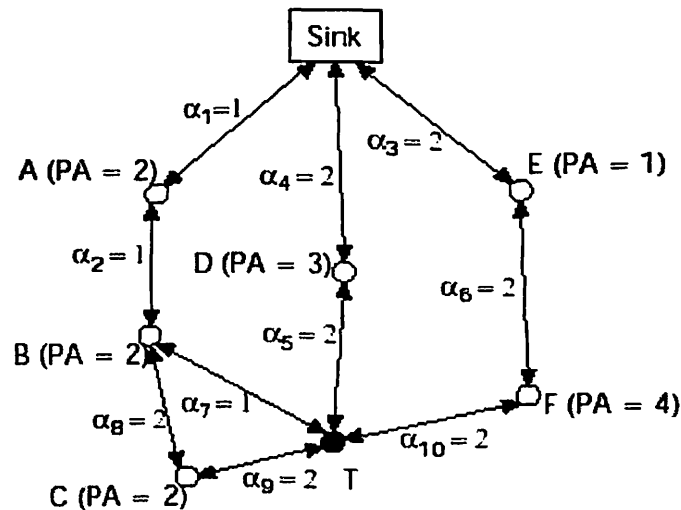


Figure 2 [12]: Energy Efficient Routes.

Energy-efficient routes can be found based on the available power (PA) in the nodes or the energy required (α) for transmission in the links along the routes. The authors of [12] (Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci) have described their network as shown in Figure 2 with specific values of α and PA. Here, the node T is one of the source sensor nodes (or an ad-hoc node) that senses the phenomena.

Assuming that α stands for unit energy consumption per node, T has the following possible routes to communicate with the sink:

- *Route 1*: Sink-A-B-T, total PA = 4, total α = 3
- *Route 2*: Sink-A-B-C-T, total PA = 6, total α = 6
- *Route 3*: Sink-D-T, total PA = 3, total α = 4
- *Route 4*: Sink-E-F-T, total PA = 5, total α = 6

An energy-efficient route is selected by one of the following approaches.

Maximum PA route: The route that has maximum total PA is preferred. The total PA is calculated by summing the PAs of each node along the route. Based on this approach, route 2 is selected in Fig. 2. However, route 2 includes the nodes in route 1 and an extra node. Therefore, although it has a higher total PA, it is not power-efficient. As a result, it is important not to consider routes derived by extending routes that can connect the sensor node to the sink as an alternative route. Eliminating route 2, we select route 4 as our power-efficient route when we use the maximum PA scheme.

Minimum energy (ME) route: The route that consumes minimum energy to transmit the data packets between the sink and the sensor node is the ME route. As shown in Fig. 2, route 1 is the ME route.

Minimum hop (MH) route: The route that makes the minimum hop to reach the sink is preferred. Route 3 in Fig. 2 is the most efficient route based on this scheme. Note that the ME scheme selects the same route as the MH when the same amount of energy (i.e., all α are the same) is used on every link. Therefore, when nodes broadcast with same power level without any power control, MH is then equivalent to ME.

Maximum minimum PA node route: The route along which the minimum PA is larger than the minimum PAs of the other routes is preferred. In Fig. 2, route 3 is the most efficient and route 1 is the second most efficient. This scheme precludes the risk of using up a sensor node with low PA much earlier than the others because they are on a route with nodes that have very high PAs.

2.2.4 Power Aware Routing Protocol: This suggests the use of different metrics when determining a routing path. Instead of shortest routing path between the source and the destination, the following metrics may be used [15]:

- Minimizing energy consumed per packet: This is a good metric, but has the disadvantage of unfairly burdening certain nodes that might get drained of battery energy finally resulting in link disconnection and network partitioning.
- Maximizing time to Network partition: This one directly addresses the network lifetime.

The following three metrics indirectly address the network lifetime.

- Minimizing variance in node power levels
- Minimizing cost/packet
- Minimizing maximum node cost

2.2.5 Alternate Path Routing Protocol (APR): This indirectly balances energy consumption by distributing network traffic among a set of diverse paths for the same source-destination pair, called an *alternate route set*. APR's performance greatly depends on the quality of the alternate route set [2], which can be measured by *route coupling*, i.e., how many nodes and links two routes have in common.

2.2.6 Localized Energy Aware Routing Protocol: This protocol directly controls the energy consumption and achieves balanced energy consumption among all participating mobile nodes [2]. The route discovery involves flooding of route-request messages. When a routing path is searched, each mobile node relies on local information on *remaining battery level* to decide whether or not to participate in the selection process of the routing path. An energy-lacking node can conserve its battery power by not forwarding data packets on behalf of others. The decision-making process here is distributed to all relevant nodes, and the destination node does not need to wait or block itself in order to find most energy-efficient path.

2.2.7 Flow Augmentation Routing Protocol: This maximizes the network lifetime by balancing the traffic among the nodes in proportion to their energy reserves [2]. The

traffic balance, in turn, can be achieved by selecting the optimal transmission power levels and optimal route.

2.2.8 Online Max-Min Routing Protocol (OOM): The data transmission sequence (or data generation rate) is not usually known in advance. Without requiring that information, the OOM protocol makes routing decision that optimizes two different metrics: *minimizing power consumption* and *maximizing the minimal residual power* in the nodes of the network [15]. Given the power level information of all nodes and the power cost between two neighboring nodes, this algorithm first finds the path that minimizes the Power consumption (P_{min}) by using *Dijkstra's algorithm*. Among the power efficient paths with some tolerance (less than αP_{min} , where $\alpha \geq 1$) it selects the best path that optimizes the second metric by iterative application of *Dijkstra's algorithm* with edge removals.

2.2.9 Power-Aware Localized Routing Protocol: Assuming that the location information of its neighbors and the destination are available through GPS, each node selects one of its neighbors through which the overall transmission power to the destination is minimized [15] [2]. Since the transmission power needed for direct communication between two nodes is super linear dependency on distance, it is usually energy efficient to transmit packets via intermediate nodes.

2.2.10 SPAN Protocol: This operates between the routing layer and the MAC layer trying to exploit the MAC layers power-saving features in its routing decision [2]. The

basic idea of the MAC layers' power-saving mechanism is to power down (*sleep*) the radio device when it has no data to transmit or receive. This allows substantial energy savings since sleep operation consumes less power. The master node must be awake all the time. In SPAN, each node periodically determines whether it should become a master or not based on the following master eligibility rule: *If two of its neighbors cannot reach each other either directly or via one or two masters, it should become a master.*

2.2.11 Geographic Adaptive Fidelity Protocol (GAF): GAF identifies many redundant nodes with respect to routing and turns them off without sacrificing the routing fidelity [2]. All nodes are in one of three states: *sleeping, discovering, and active.*

2.2.12 Data-centric routing [12]:

Routing may be based on the data-centric approach. Directed diffusion [12] is a data-centric protocol for sensor network applications. It achieves some level of energy savings by selecting empirically good paths, and by caching and processing data in-network. Rumor routing [13] is one highly data-centric routing mechanism which allows queries to be delivered to events in the network. In data-centric routing, interest dissemination is performed to assign the sensing tasks to the sensor nodes. There are two approaches used for interest dissemination: sinks broadcast the interest [4], and sensor nodes broadcast an advertisement for the available data [9] and wait for a request from the interested nodes. Data-centric routing requires attribute-based naming [5]. For attribute based naming, the users are more interested in querying an attribute of the phenomenon, rather than querying an individual node. For instance, "*the areas where the temperature is over*

70°F” is a more common query than “the temperature read by a certain node.” Attribute-based naming is used to carry out queries by using the attributes of the phenomenon. Attribute-based naming also makes broadcasting, attribute-based multicasting, and geocasting important for sensor networks.

Data aggregation is a technique used to solve the implosion and overlap problems in data-centric routing [9]. In this technique, a sensor network is usually perceived as a reverse multicast tree, as shown in Figure 3 [12], where the sink asks the sensor nodes to report the ambient condition of the phenomena. Data coming from multiple sensor nodes are aggregated as if they are about the same attribute of the phenomenon when they reach the same routing node on the way back to the sink.

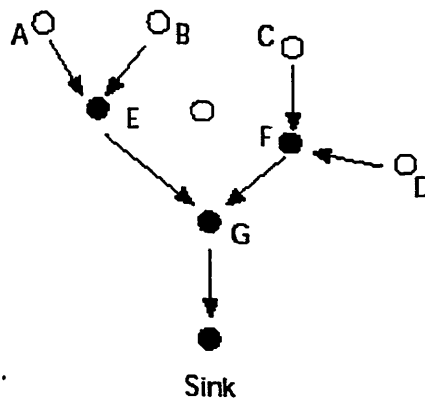


Figure 3: Data Aggregation

For example, sensor node E aggregates the data from sensor nodes A and B while sensor node F aggregates the data from sensor nodes C and D, as shown in Fig. 3. Data aggregation can be perceived as a set of automated methods of combining the data that comes from many sensor nodes into a set of meaningful information [8].

In this respect, data aggregation is known as *data fusion* [9]. Also, care must be taken when aggregating data, because the specifics of the data (e.g., the locations of

reporting sensor nodes) should not be left out. Such specifics may be needed by certain applications.

2.3 Secure Routing

2.3.1 Multi-tiered security architecture [6]:

Key attributes of wireless micro sensor networks are the severely constrained computational and energy resources, and an ad hoc operational environment. Resource limitations and specific architecture of sensor networks call for customized security mechanisms. A communication security scheme has been proposed where for each type of data a corresponding security mechanism was defined [6]. By employing a multi-tiered security architecture where each mechanism has different resource requirements, efficient resource management can be done, which is essential for wireless sensor networks. This approach to communication security in sensor networks is based on the principle that the data items must be protected to a degree consistent with their value. Types of data sent through the network can be differentiated into three:

1. Mobile code
2. Locations of sensor nodes
3. Application specific data

The deployment of security mechanisms in a sensor network creates additional overhead. Not only does latency increase due to the execution of the security related procedures, but also the consumed energy directly decreases the lifetime of the network. To minimize the security related costs, security overhead, and consequently the energy

consumption, should correspond to sensitivity of the encrypted information. Following the taxonomy of the types of data the network, three security levels are defined in [6]:

- Security level I is reserved for mobile code, the most sensitive information sent through the network [6].
- Security level II is dedicated to the location information conveyed in messages [6].
- Security level III mechanism is applied to the application specific information [6].

Security level I allows the use of strong encryption in spite of the resulting overhead. To constrain the damage only to one part of the network, location-based keys are proposed for level II encryption. The location-based keys enable separation between regions where the location of nodes are compromised and the areas where nodes continue to operate safely. The application specific data is encrypted using a weaker encryption that requires lower computational overhead. Thus, high frequency of messages with application specific data prevents the usage of stronger resource consuming encryption.

2.3.2 SPINS (SNEP and μ TESLA) [1]:

Security Protocols for Sensor Networks [SPINS] was introduced in [1]. It is comprised of two protocols: Sensor Network Encryption Protocol (SNEP) [1] and μ TESLA [1]. The function of SNEP is to provide:

- a) Confidentiality: Eavesdroppers should not be able to determine what information the sensors are sending back to the base station.
- b) Authentication: It should not be possible for an adversary to spoof messages as though they were coming from either nodes or the base station.

c) Integrity: Authentication usually implies integrity since it will be difficult for an adversary to modify a signed message, and maintain the signature while modifying the data. It can also mean that if the message is modified by "noise," the signature is still intact.

d) Freshness: It should be clear to the base station or the nodes what messages are current, and which are old.

In a SPINS sensor network all unicast communications either from the base station to a node or from a node to the base station rely on SNEP [1]. In SNEP, each node has a different key, which it shares with the base station. This provides both confidentiality and authentication. To ensure freshness each node also maintains a counter with the base station. It uses this counter as the initialization vector for the stream cipher used, which is RC5 [10]. Whenever a message is sent from either the base station to a node i or from a node i to the base station, the i th counter is incremented. Thus, the node can know the order in which messages arrived from the base station.

If some messages are lost, then the node can try decrypting with counter values around the current counter value it has stored. If it cannot decrypt the message, then it will need to resynchronize the counter with the base station. In each node two keys are generated from the key shared with the base station. That is if node i has key K_i it shares with the base station, then it generates and sends $K_{i,encr}$ and $K_{i,mac}$ to the base station. $K_{i,encr}$ is used for confidentiality, $K_{i,mac}$ is used for message authentication.

For authentication, a message authentication code or MAC is used. Since node i only shares its particular $K_{i,mac}$ with the base station, the base station knows that any message encrypted with that particular $K_{i,mac}$ must have originated from node i .

Likewise in the case that node i is receiving a message encrypted with $K_{i,mac}$, node i knows that it must have originated from the base station. In the case of SPINS, in order to save memory space on the nodes, the MAC is just the same as the encryption algorithm: RC5.

A message sent from either the base station to node i or from node i to the base station consists of the following:

$A \rightarrow B: \{D\}_{\langle K_{encr}, C \rangle}, MAC(K_{mac}, \{D\}_{\langle K_{encr}, C \rangle})$

The data is sent encrypted with K_{encr} using as an initialization vector, the counter C . The MAC is computed using key K_{mac} and plaintext $\{D\}_{\langle K_{encr}, C \rangle}$. Although it is possible to know which messages came in which order from either the base station or node i , it is not possible to know whether a message originated in response to a specific request from the base station. To enable this, the base station sends a random bit string or nonce with each request it sends to node i . When node i is responding to that request, it includes the random bit string in its response so the base station knows that it is responding to that request.

μ TESLA provides authentication to data broadcasts [1]. Authenticated broadcasts cannot be done with SNEP because if every node getting the broadcast knows the same key, then any node could pretend as the base station. The problem would be easy if the base station could securely distribute its public key and then just sign all its messages with its private key using a public key cryptosystem like in [11]. The problem is that asymmetric cryptography is too computationally intensive for the devices.

In μ TESLA protocol, the base station calculates a key chain i.e., it generates some key K_n , and then using a one way hash function F , it calculates $F(K_n) = K_{n-1}$, $F(K_{n-1}) =$

$K_{n-2} \dots F(K_1) = K_0$. It sends to all the nodes K_0 (it can use SNEP to do this). All the nodes are time synchronized with the base station. The protocol is that at periodic intervals the keys K_1, K_2, \dots, K_n are broadcast. Say at t_1 , K_1 is revealed, at t_2 , K_2 is revealed, and at t_i , K_i is revealed. In the span t_0 to t_1 all packets broadcast by the base station are encrypted with K_1 . The nodes cannot decrypt it because they do not know K_1 . However, once K_1 is revealed they can be certain it came from the base station, because only the base station knows K_1 and they can verify that $F(K_1) = K_0$. If any imposter broadcasts a message m_i encrypted with key K_i and then broadcasts the key K_i the nodes would know that it is an imposter because $F(K_i)$ would not equal K_{i-1} .

2.3.3 Secure Routing – Attacks and Counter measures [3]:

Crippling attacks against all the major routing protocols for sensor networks are presented in detail and countermeasures suggested in [3]. Two classes of novel attacks against sensor networks — sinkholes and HELLO floods are also introduced. The authors claim this to be the first such analysis of secure routing in sensor networks.

Security of all major sensor network routing protocols were analyzed. Because these protocols have not been designed with security as a goal, it is not surprising that they are all insecure. However, this is non-trivial to fix: it is unlikely a sensor network routing protocol can be made secure by incorporating security mechanisms after design has completed. The authors assert that sensor network routing protocols must be designed with security in mind, and this is the only effective solution for secure routing in sensor networks. The rest of this section describes the various attacks and countermeasures as suggested by [3].

Many sensor network routing protocols are quite simple, and for this reason are sometimes even more susceptible to attacks against general ad-hoc routing protocols. Most network layer attacks against sensor networks fall into one of the following categories:

- Spoofed, altered, or replayed routing information
- Selective forwarding
- Sinkhole attacks
- Sybil attacks
- Wormholes
- HELLO flood attacks
- Acknowledgement spoofing

Attacks:

A. Spoofed, altered, or replayed routing information

The most direct attack against a routing protocol is to target the routing information exchanged between nodes. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, increase end-to-end latency, etc.

B. Selective forwarding

Multi-hop networks are often based on the assumption that participating nodes will faithfully forward the received messages. In a selective forwarding attack, malicious nodes may refuse to forward certain messages and simply drop them, ensuring that they

are not propagated any further. A simple form of this attack is when a malicious node behaves like a black hole and refuses to forward every packet she sees. However, such an attacker runs the risk that neighboring nodes will conclude that she has failed and decides to seek another route. A more subtle form of this attack is when an adversary selectively forwards packets. An adversary interested in suppressing or modifying packets originating from a select few nodes can reliably forward the remaining traffic and limit suspicion of her wrongdoing.

C. Sinkhole attacks

In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Because nodes on, or near, the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks (selective forwarding, for example).

Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high quality route to a base station. Some protocols might actually try to verify the quality of route with end-to-end acknowledgements containing reliability or latency information. In this case, a laptop-class adversary with a powerful transmitter can actually *provide* a high quality route by transmitting with enough power to reach the base station in a single hop, or by using a wormhole attack.

Due to either the real or imagined high quality route through the compromised node, it is likely each neighboring node of the adversary will forward packets destined for

a base station through the adversary, and also propagate the attractiveness of the route to its neighbors. Effectively, the adversary creates a large “sphere of influence”, attracting all traffic destined for a base station from nodes several hops away from the compromised node.

D. The Sybil attack

In a Sybil attack [14], a single node presents multiple identities to other nodes in the network. The Sybil attack can significantly reduce the effectiveness of fault-tolerant schemes such as distributed storage, dispersity and multipath routing, and topology maintenance. Replicas, storage partitions, or routes believed to be using disjoint nodes could in actuality be using a single adversary presenting multiple identities.

E. Wormholes

In the wormhole attack [6], an adversary tunnels messages received in one part of the network over a low latency link and replays them in a different part. The simplest instance of this attack is a single node situated between two other nodes forwarding messages between the two of them. However, wormhole attacks more commonly involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the attacker. An adversary situated close to a base station may be able to completely disrupt routing by creating a well-placed wormhole. An adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the wormhole. This can create a sinkhole: since the adversary on the other side of the wormhole can artificially provide a high-quality route to the base station, potentially all traffic in the surrounding area will be drawn through her if alternate routes are

significantly less attractive. This will most likely always be the case when the endpoint of the wormhole is relatively far from a base station. Wormholes can also be used simply to convince two distant nodes that they are neighbors by relaying packets between the two of them. Wormhole attacks would likely be used in combination with selective forwarding or eavesdropping. Detection is potentially difficult when used in conjunction with the Sybil attack.

F. HELLO flood attack

The authors of [3] introduce a novel attack against sensor networks: the HELLO flood. Many protocols require nodes to broadcast HELLO packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false: a laptop-class attacker broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbor. For example, an adversary advertising a very high quality route to the base station to every node in the network could cause a large number of nodes to attempt to use this route, but those nodes sufficiently far away from the adversary would be sending packets into oblivion. The network is left in a state of confusion. A node realizing the link to the adversary is false could be left with few options: all its neighbors might be attempting to forward packets to the adversary as well. Protocols, which depend on localized information exchange between neighboring nodes for topology maintenance or flow control, are also subject to this attack. An adversary does not necessarily need to be able to construct legitimate traffic in order to use the HELLO flood attack. She can simply re-broadcast overhead

packets with enough power to be received by every node in the network. HELLO floods can also be thought of as one-way, broadcast wormholes.

G. Acknowledgement spoofing

Several sensor network routing algorithms rely on implicit or explicit link layer acknowledgements. Due to the inherent broadcast medium, an adversary can spoof link layer acknowledgments for “overheard” packets addressed to neighboring nodes. Goals include convincing the sender that a weak link is strong or that a dead or disabled node is alive. For example, a routing protocol may select the next hop in a path using link reliability. Artificially reinforcing a weak or dead link is a subtle way of manipulating such a scheme. Since packets sent along weak or dead links are lost, an adversary can effectively mount a selective forwarding attack using acknowledgement spoofing by encouraging the target node to transmit packets on those links.

Countermeasures:

A. Outsider attacks and link layer security

The majority of outsider attacks against sensor network routing protocols can be prevented by simple link layer encryption and authentication using a globally shared key. The Sybil attack is no longer relevant because nodes are unwilling to accept even a single identity of the adversary. The majority of selective forwarding and sinkhole attacks are not possible because the adversary is prevented from joining the topology.

Link layer acknowledgements can now be authenticated. Major classes of attacks not countered by link layer encryption and authentication mechanisms are wormhole attacks and HELLO flood attacks. Although an adversary is prevented from joining the network, nothing prevents her from using a wormhole to tunnel packets sent by legitimate nodes in

one part of the network to legitimate nodes in another part to convince them they are neighbors or by amplifying an overheard broadcast packet with sufficient power to be received by every node in the network.

B. The Sybil attack

An insider cannot be prevented from participating in the network, but she should only be able to do so using the identities of the nodes she has compromised. Using a globally shared key allows an insider to masquerade as *any* (possibly even nonexistent) node. Identities must be verified. In the traditional setting, this might be done using public key cryptography, but generating and verifying digital signatures is beyond the capabilities of sensor nodes. One solution is to have every node share a unique symmetric key with a trusted base station. Two nodes can then use a protocol to verify each other's identity and establish a shared key. A pair of neighboring nodes can use the resulting key to implement an authenticated, encrypted link between them. In order to prevent an insider from wandering around a stationary network and establishing shared keys with every node in the network, the base station can reasonably limit the number of neighbors a node is allowed to have and send an error message when a node exceeds it.

Thus, when a node is compromised, it is restricted to (meaningfully) communicating only with its verified neighbors. This not to say that nodes are forbidden from sending messages base stations or aggregation points multiple hops away, but they are restricted from using any node except their verified neighbors to do so. In addition, an adversary can still use a wormhole to create an artificial link between two nodes to convince them they are neighbors, but the adversary will not be able to eavesdrop on or modify any future communications between them.

HELLO flood attacks:

The simplest defense against HELLO flood attacks is to verify the bidirectionality of a link before taking meaningful action based on a message received over that link. The identity verification protocol described in the previous section is sufficient to prevent HELLO flood attacks. Not only does it verify the bidirectionality of the link between two nodes, but even if a well-funded adversary had a highly sensitive receiver or had wormholes to a multiple locations in the network, a trusted base station that limits the number of verified neighbors for each node will still prevent HELLO flood attacks on large segments of the network when a small number of nodes have been compromised.

Wormhole and sinkhole attacks:

Wormhole and sinkhole attacks are very difficult to defend against, especially when the two are used in combination. Wormholes are hard to detect because they use a private, out-of-band channel invisible to the underlying sensor network. Sinkholes are difficult to defend against in protocols that use advertised information such as remaining energy or estimate of end-to-end reliability to construct a routing topology because this information is hard to verify. Routes that minimize the hop-count to a base station are easier to verify, however hop-count can be completely misrepresented through wormhole. When routes are established simply based on the reception of a packet, sinkholes are easy to create because there is no information for a defender to verify. A technique for detecting wormhole attacks is presented [13], but it requires extremely tight time synchronization and is thus infeasible for most sensor networks. Because it is extremely difficult to retrofit existing protocols with defenses against these attacks, the best solution

is to carefully design routing protocols in which wormholes and sinkholes are meaningless.

E. Leveraging global knowledge

A significant challenge in securing large sensor networks is their inherent self-organizing, decentralized nature. When the network size is limited or the topology is well structured or controlled, global knowledge can be leveraged in security mechanisms. Consider a relatively small network of around 100 nodes or less. If it can be assumed that no nodes are compromised during deployment, then after the initial topology is formed, each node could send information such as neighboring nodes and its geographic location (if known) back to a base station. Using this information, the base station(s) can map the topology of the entire network. To account for topology changes due to radio interference or node failure, nodes would periodically update a base station with the appropriate information. Drastic or suspicious changes to the topology might indicate a node compromise, and the appropriate action can be taken.

F. Selective forwarding

Even in protocols completely resistant to sinkholes, wormholes, and the Sybil attack, a compromised node has a significant probability of including itself on a data flow to launch a selective forwarding attack if it is strategically located near the source or a base station. Multipath routing can be used to counter these types of selective forwarding attacks. Messages routed over n paths whose nodes are completely disjoint are completely protected against selective forwarding attacks involving at most n compromised nodes and still offer some probabilistic protection when over n nodes are compromised. However, completely disjoint paths may be difficult to create. Braided

paths may have nodes in common, but have no links in common (i.e., no two consecutive nodes in common). The use of multiple braided paths may provide probabilistic protection against selective forwarding and use only localized information. Allowing nodes to dynamically choose a packet's next hop probabilistically from a set of possible candidates can further reduce the chances of an adversary gaining complete control of a data flow.

G. Authenticated broadcast and flooding

Since base stations are trustworthy, adversaries must not be able to spoof broadcast or flooded messages from any base station. This requires some level of asymmetry: since every node in the network can potentially be compromised, no node should be able to spoof messages from a base station, yet every node should be able to verify them. Authenticated broadcast is also useful for localized node interactions. Many protocols require nodes to broadcast HELLO messages to their neighbors. These messages should be authenticated and impossible to spoof.

H. Countermeasure summary

Link-layer encryption and authentication, multipath routing, identity verification, bidirectional link verification, and authenticated broadcast can protect sensor network routing protocols against outsiders, bogus routing information, Sybil attacks, HELLO floods, and acknowledgement spoofing, and it is feasible to augment existing protocols with these mechanisms. Sinkhole attacks and wormholes pose significant challenges to secure routing protocol design, and it is unlikely that there exist effective countermeasures against these attacks that can be applied after the design of a protocol has completed. It is crucial to design routing protocols in which these attacks are

meaningless or ineffective. Geographic routing protocols are one class of protocols that holds promise.

Chapter 3

THESIS STATEMENT

This thesis investigates the problems of power consumption and security in sensor networks with respect to routing. While providing a routing algorithm that caters to any of these factors is a nontrivial exercise, the dual factors create more complexity. But as described in the previous chapter, it is difficult to fix a protocol after it is designed or implemented. So, the main objectives of this thesis are:

1. To propose a Secure, Energy- Efficient algorithm for Sensor networks.
2. Evaluate the effectiveness of these secure protocols.
3. Suggest other existing security schemes that can be applied at other levels in the network to maximize the effect.

Though most of the existing security schemes for sensor networks are application specific, the approach here is purely application independent and can be applied for routing in sensor networks in general. Also, as emphasized in the previous chapter, it is always good to combine security schemes at various levels to obtain best results. So, an attempt has been made to make suggestions on some future work as to what other existing schemes can be combined with the proposed solution.

Chapter 4

PROPOSED SOLUTION

Wireless sensor networks are an evolving area of technology and research. These are being deployed in areas where communication security is highly crucial. The information transferred through a sensor network can be about weather conditions, the presence of toxins in the environment, structural soundness of buildings during/after an earthquake, and the progression of hazards such as fires to direct evacuation of building occupants or guide firefighters. The Department of Defense is particularly interested in the security of these networks.

Our objective is to design and develop a Secure, Energy-Efficient routing algorithm for sensor networks in this thesis, with routing security and power balance in the network as main criteria under consideration. There are other considerations also, which will be explained in their specific contexts. An aim is that our secure data should not be tracked or tapped by an adversary.

Game theory [13] presents one possible solution to our problem. The security problem may be viewed as a game between two players – the designer of the routing algorithm and the attacker who attempts to intercept packets. In the approach considered in [13], the designer wants to minimize the time it takes for a packet to be safely

transmitted, whereas the attacker wants to minimize this time, so that the attacker has more time to intercept. Although our approach is driven by the work of Hespanha and Bobacek [13] who consider routing games, our proposed solution is not a formal derivation based on game theory. Our approach provides a solution to secure routing that is approximately based on the work reported in [13].

The cost to be minimized by the designer of the routing algorithm and maximized by the attacker is $J(\epsilon) := E[\chi(\epsilon)]$, where $\epsilon \geq 0$ is a design parameter and $\chi(\epsilon)$ is a random variable that is equal to $(1+\epsilon)^{t-1}$ if the packet is intercepted at the t th hop and 0 otherwise. Therefore this approach discourages long paths and will favor short paths. In other words, for $\epsilon \neq 0$, $J(\epsilon)$ bias the solution sought by the player that designs the routing policy towards shorter paths since, when being caught is inevitable, it incurs in less cost if it is caught sooner than later. For $\epsilon = 0$, the random variable $\chi(\epsilon)$ is equal to 1 if the packet is intercepted and 0 otherwise and therefore $J(0)$ is simply the probability that the packet will be intercepted. The cost $J(0)$ assumes that all paths are equal. Clearly, as formulated above, it is too simplistic to model a realistic attack.

Consider a data transmission network with nodes $N := \{1, 2, \dots, n\}$ connected by unidirectional links. We denote by L the set of all links and use the notation $j \rightarrow i$ to represent a link from node j to node i . Without loss of generality, the source and destinations nodes can be taken to be 1 and n , respectively. Hespanha [13] considers stochastic routing policies where under such policies, whenever a packet arrives at node $k \in N$, it will be routed through link $k \rightarrow i \in L$ with probability $r_{k \rightarrow i} \geq 0$. In our approach as far as the routing is concerned, each routing policy is characterized by a list $R := \{r_l : l \in L\}$ that satisfies

$$\sum_{k: i \rightarrow k \in L} r_{i \rightarrow k} = C_m, \forall i \in N$$

where C_m is a constant for the m^{th} iteration. In our proposed solution we consider both power and probability of a route going through a node (considering its selection history). $C_m = 1$ if we consider only routing probability, in other words, the probabilities of the links input to a node will be 1. Therefore C_m is an indication of the node's remaining power and the routing probability. In [13] they restrict their attention to stochastic cycle-free routing policies. These are stochastic routing policies for which a packet will not return to a node where it has been before with probability one. We denote by R_{nocycle} the set of lists with this property.

Similarly an attacker will have an attack policy. An attacker stochastic policy

$D = \{d_l : \sum_l d_l = 1, l \in L\}$ which simply consists of a distribution over the elements of L , where d_l is the probability that the adversary will attempt to intercept packets in the link l . We need to determine a security policy $R^*(\epsilon)$ such that

$$J^*(\epsilon) := \min_{R \in R_{\text{nocycle}}} \max_{D \in \{d_l\}} J_{RD}(\epsilon) = \max_{D \in \{d_l\}} J_{R^*(\epsilon)D}(\epsilon),$$

where $J_{RD}(\epsilon)$ denotes the value of the cost $J(\epsilon)$ incurred when the routing policy R is used and the attacker selects the policy D . We assume that we are not aware of the attacker's policy. Our objective therefore is to minimize $J^*(\epsilon) := \min_{R \in R_{\text{nocycle}}} J_{RD}(\epsilon)$.

Finding the minimum will always result in the shortest path. Our routing policy takes into account both security and power. Our routing policy aims to consider both the previous selection of a node and power available for the node. Our objective therefore is to maximize $J^*(\epsilon) := \max_{R \in R_{\text{nocycle}}} f(PS(\epsilon) \cdot PA(\epsilon))$ where PS and PA are previous selection of a node and power available respectively, f is a function and in our case we choose ϵ to be a node.

Although not strictly stochastic, the algorithm chooses multiple routes which brings in unpredictability of routing and also avoiding frequently repeating routes. However, the routing is not just based on some chance random numbers. The algorithm also always achieves balanced energy consumption. The algorithm aims to minimize the effectiveness of user attacks. Our algorithm is therefore an approximate implementation of the ideas behind game theory.

The algorithm is proposed by considering various factors both at the time of creation of the route and at the selection of the routes for packet transmissions. At the stage of creation of routes the two factors taken into account are how many times a node is selected previously for route and the remaining power level of the node. At the stage of selection of the complete routes for packet transfers, the two factors taken into account are the length of the route and the selection history of the route. Also, the final routes selected will not be directly used in the generated sequence for additional security benefit. Pre-calculated route pattern is kept aside and stochastically a random route from the pattern is selected to minimize the possibility that an adversary can detect patterns out of our routes.

The assumptions of the algorithm are:

- Each sensor node has its location information and its neighbors' with the help of a GPS system or a location tracking system.
- Each node forms a cluster where it is the cluster head. The cluster is simply the set of nodes reachable from cluster head. This does not incur a big overhead given a handy Global Positioning System.

- Location information is updated periodically at reasonable intervals. Location information is fairly precise (with little or no approximation) and reliable between update intervals.
- Each of the sensor nodes can act as a Source and the destination node D is the sink node.
- Destination D can send a request for a message. This request can be sent in regular (insecure) mode. Interception of this request should not pose a compromise to the secure message itself, because we assume the attacker is aware of a secure message being sent from S to D.
- The message is decomposed by S into a number of packets. The decomposition process embeds redundancy (e.g., parity) so that only p packets are needed to reconstruct the message. The message is compromised if at least p packets are intercepted.
- D determines the number of routes (paths) R needed for secure routing of the sensor information message.
- Each route i carries the following values with it:
 - L_j = Route Length of route j.
 - S_j = Number of previous selections of route j.
- Each node i carries the following values with it:
 - i_p = Remaining power level of itself and all its neighbors.
 - i_{sel} = Number of times the node is selected in the route discovery algorithm and the same information about its neighbors.
 - The partial route for which it is a head node.

- A set of all the nodes on all the paths
- The entire set of computed routes if the node is a source node.

Let

h_i : The set of cluster heads that cover node i .

n_i : The number of cluster heads that cover node i .

A : The set of nodes that are on any route.

r : The number of initial paths that might be needed to obtain R .

Secure, Energy-Efficient Route Discovery Algorithm executing at S for each message:

1. $A = \{D\}$
2. Randomly select 75% of nodes from among D_i and from this set select r nodes with highest values of $((1/ i_{sel}) \times i_p)$.
3. Connect these nodes to D . These are now the heads of r paths to D
4. Add these nodes to A
5. Loop for each head node of path i
 - a. If the head is S , add the entire path to R and exit this loop. Otherwise continue
 - b. Select randomly a cluster head node of i from half of the set of h_i , the half set being those nodes with highest values of $((1/ i_{sel}) \times i_p)$.
 - c. If the head is already in A , then select another from h_i . If h_i is empty, then remove this incomplete path this path fails to connect S and D .
 - d. Connect the head to path i .
 - e. Remove old head node and insert new head node to A .

Sending sensor data packets

6. Compute and select p routes (for each of the p packets) from R with highest value of $(1/L_j) \times (1/S_j)$ of the routes and add them to a set P .
7. For each of the p packets:
 - a. Randomly select a route from P .
 - b. Send this packet over this path. Route information is stored in the packet that may be used for authentication at other levels of the network.
 - c. Node information of each node on the route is adjusted based on the remaining power level after message transmission
 - d. If a node falls below the “threshold” power level
 - i. It is dropped out of the network and the route is removed for this message transmission.
 - ii. All its occurrences in set P are removed and replaced with another route from R with highest value of $(1/L_j) \times (1/S_j)$.
 - e. Remove this route from the set P .

(Note that by the time all packets are successfully transmitted over the network to the sink node, the set P should be empty)

Introduction of Step-6 in the algorithm has great significance by itself. It makes sure that even when the dual criteria of highest probability routing and balanced energy transfer demands specific routes, we would pre-calculate those routes and use them in a stochastic manner so that repetition is not seen, which means that an adversary would not be able to decipher the information of any message unless p packets are compromised.

It should be specifically noted here that the probability factors involved in steps 2, 5(b) and 7(a) are selected for convenience, good understanding and simulation purposes. In actuality, these probability factors may be complex functions of remaining power level of the nodes, length of the routes, number of times a node is used in route selection process, number of times a route is used (based on what the pattern is of the recently used routes) etc. So, the proposed algorithm may be used in a much customized manner when used in real world secure energy-efficient sensor network applications.

Complexity Analysis of the algorithm:

When there are n nodes in the network, the complexity of the algorithm is $O(n)$. At the beginning of the route discovery in the sensor network, the algorithm starts at the destination D and processes all of the other $(n-1)$ nodes. It determines which of them are its neighbors and depending on the constraints and conditions described in the algorithm above, and decides on how many routes are required and thus proceeds to the rest of the algorithm to complete each route. This infers that the complexity of the algorithm is $O(n)$.

Properties of this algorithm:

- No common sub-paths among paths: As seen from Step 4 of the Algorithm, we maintain a set A to hold all the nodes located on all the paths of the routes. Step 5(c) makes sure that no node that is already on a route is selected again. This eliminates the possibility of a node being repeated on the route which means no common sub-paths.

- No node lies on two different paths: For the same reason that Step 5(c) makes sure that a node which is already on one of the routes will not be allowed to be selected on another route, the result is that, a same node does not appear on two different paths for a single message transmission.
- Some paths may not reach S: While creating a route, it is made sure by step 5(c) that a node does not repeat on two different paths. So, in this next-node selection process if all the neighboring nodes are already placed on other routes, this route cannot be completed, in which case this path will not reach the source in this back to front approach. Hence this route will be deleted from the route domain and the algorithm continues.
- No cycles in the paths: There is no possibility from the algorithm that a node that is on a particular route is selected again on to any of the routes. That infers that a node will never appear twice on a route, meaning no cycles in the routes.

It should be noted that this is a distributed algorithm, which means that there is no single centralized node which does all the calculations. All the calculations done by individual nodes in the sensor network are as follows:

- When finding the next node on the route, each node calculates the $((1/ i_{sel}) \times i_p)$ of all its neighboring nodes
- When the source node needs to transfer the packets, it determines the pattern of routes to be used for packet transfer by calculating $(1/ L_j) \times (1/ S_j)$.

Chapter 5

SIMULATION AND RESULTS

In order to investigate the performance of the proposed algorithm, it has been implemented as a C++ program. Classes like “node” and “route” are created so that their attributes and behaviors can be implemented in object-oriented manner. To generate the numerous network topologies necessary for testing, a highly efficient and flexible tool called “BRITE”[8] has been utilized. The topologies have been generated using BRITE on a Sun Solaris machine using the java version of “BRITE”. The algorithm has been implemented taking BRITE’S output file as an input to the C++ program as is and tested for performance with respect to energy efficiency and security issues.

5.1 BRITE

BRITE is a universal topology generator developed by researchers at Boston University [7]. BRITE was designed to be a flexible topology generator, not restricted to any particular way of generating topologies. As such, it supports multiple generation models. BRITE reads the generation parameters from a configuration file that can be either hand written by the user or automatically generated by BRITE's GUI. BRITE provides the capability of importing topologies generated by other topology generators or

topological data. It is possible to generate topologies using BRITE and then reuse them to generate other topologies by combining them with BRITE models or other imported formats [7]. For simulation of the proposed algorithm, a flat topology model is chosen.

The parameters that BRITE takes are shown in Figure 4.

Parameter	Meaning	Values
HS	Size of one side of the plane	int ≥ 1
LS	Size of one side of a high-level square	int ≥ 1
N	Number of nodes	int $1 \leq N \leq HS * HS$
Model	model id	int ≥ 1
alpha	Waxman-specific exponent	$0 < \alpha \leq 1, \alpha \in R$
beta	Waxman-specific exponent	$0 < \beta \leq 1, \beta \in R$
Node Placement	how nodes are placed in the plane	1: Random. 2: HT
m	Number of links per new node	int ≥ 1
Growth Type	how nodes join the topology	1: Incremental. 2: Random
BWdist	bandwidth assignment to links	1: Const. 2: Unif. 3: Exp. 4: HT
MaxBW, MinBW	min. max link bandwidth values	float > 0

Figure 4: Flat topology parameters for BRITE [6]

5.2 Format of BRITE Output

A BRITE-formatted output file contains three sections [6]:

1. Model information: Information about the topology contained in the file. Includes number of nodes and edges, and information specific to the model used to generate the topology.
2. Node Section: For each node in the graph, a line is written into the output file with the following format: NodeId xpos ypos indegree outdegree ASid type.
3. Edge Section: For each edge in the topology, information like edge id, from node, to node, length, delay, bandwidth, ASfrom, ASto and type are written to the output file.

A sample of BRITE's output generated with 10 nodes and 20 edges, HS = 1000 and LS = 100 is shown in Figure 5.

```

Topology: ( 10 Nodes, 20 Edges )
Model (3 - ASWaxman): 10 1000 100 1 2 0.15 0.2 1 1 10.0 1024.0

Nodes: ( 10 )
0      179   771   2     2     0     AS_NONE
1      958   484   2     2     1     AS_NONE
2      198   705   5     5     2     AS_NONE
3       51   566   3     3     3     AS_NONE
4      398   552   4     4     4     AS_NONE
5      441   932   4     4     5     AS_NONE
6      486   267   6     6     6     AS_NONE
7       36   774   6     6     7     AS_NONE
8      663   867   3     3     8     AS_NONE
9      327   600   5     5     9     AS_NONE

Edges: ( 20 )
0       7     9     339.0531  -1.0  10.0  7     9     E_AS_NONE  U
1       7     8     633.8596  -1.0  10.0  7     8     E_AS_NONE  U
2       6     9     369.0122  -1.0  10.0  6     9     E_AS_NONE  U
3       6     7     677.90045 -1.0  10.0  6     7     E_AS_NONE  U
4       5     6     666.5208  -1.0  10.0  5     6     E_AS_NONE  U
5       5     7     434.72864 -1.0  10.0  5     7     E_AS_NONE  U
6       4     9     85.70297  -1.0  10.0  4     9     E_AS_NONE  U
7       4     6     298.2767  -1.0  10.0  4     6     E_AS_NONE  U
8       3     7     208.54016 -1.0  10.0  3     7     E_AS_NONE  U
9       3     6     527.85034 -1.0  10.0  3     6     E_AS_NONE  U
10      2     6     524.2023  -1.0  10.0  2     6     E_AS_NONE  U
11      2     7     176.08237 -1.0  10.0  2     7     E_AS_NONE  U
12      1     4     564.11346 -1.0  10.0  1     4     E_AS_NONE  U
13      1     5     684.10016 -1.0  10.0  1     5     E_AS_NONE  U
14      0     2     68.68042  -1.0  10.0  0     2     E_AS_NONE  U
15      0     4     309.71277 -1.0  10.0  0     4     E_AS_NONE  U
16      9     2     166.331   -1.0  10.0  9     2     E_AS_NONE  U
17      9     5     351.02707 -1.0  10.0  9     5     E_AS_NONE  U
18      8     2     492.4114  -1.0  10.0  8     2     E_AS_NONE  U
19      8     3     682.0154  -1.0  10.0  8     3     E_AS_NONE  U

```

Figure 5: Sample BRTE output

All of the parameters output by BRITE are not necessary for our purpose. Only the nodes', their neighbors' and edges' data is required.

5.3 System attributes

The simulations were performed at two levels of network sizes, with 10 sample points selected in level 1 and 30 sample points selected at level 2. The two network sizes are as follows:

- a) Sensor networks of size from 5 to 100 nodes.
- b) Sensor networks of size from 100 to 300 nodes.

Other attributes considered for simulation are based on a representative example called Mica *mote* [3], a small sensor/actuator unit with a CPU, power source, radio, and several optional sensing elements. The CPU consumes 5.5 mA (at 3 volts) when active. And two orders of magnitude less when sleeping. The radio is a 916 MHz low-power radio from RFM, delivering up to 40 Kbps bandwidth on a single shared channel and with a range of up to a few dozen meters. The RFM radio consumes 4.8 mA (at 3 volts) in receive mode, up to 12 mA in transmit mode, and 5 μ A in sleep mode.

For the purpose of simulation, it is considered that all the nodes are in sleep mode except when receiving, computing or transmitting. Also, the sleep mode power consumption is ignored for comparison as it is of several orders less when compared to power consumption in active mode.

The ability of putting nodes to sleep except when necessary can be obtained at lower levels of the network by using various possible techniques at the MAC layer. This thesis is limited to the network layer and routing algorithm. As mentioned earlier in the thesis, an optimal power saving sensor network is obtained by taking measures at different layers of the network.

5.4 Graphs

The following is the graph for comparing the energy consumption (based on the mica mote system attributes [3]) between the proposed algorithm and the flooding algorithm for the first level of network size.

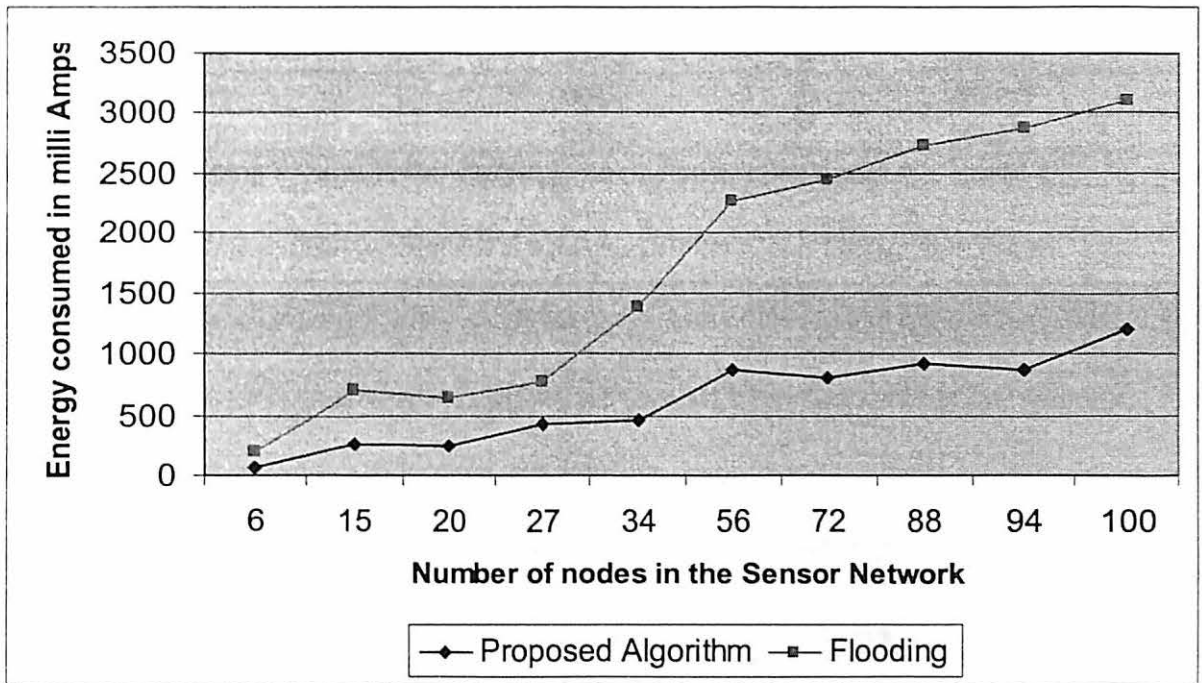


Figure 6: Energy Consumption - Flooding versus proposed algorithm – Level 1

The graph in Figure 6 is drawn based on the results shown in Figure 7. Column 1 lists the number of nodes in the network. Columns 2 and 3 list the amount of power consumed in milli Watts. These results obtained are for transmitting 40 Kbps over a one second period. The average power consumption for the two algorithms is as follows:

- a) Proposed Secure Energy Efficient algorithm – 1115 mW
- b) Flooding – 3117.945 mW

This shows that the power consumed in case of flooding is almost three times that of the proposed algorithm.

Number of nodes	Power consumed in mW using Proposed Algorithm	Power consumed in mW using Flooding Algorithm
6	66.9	200.7
15	267.6	713.6
20	245.3	646.7
27	423.7	780.5
34	468.3	1382.6
56	869.7	2274.6
72	802.8	2453
88	914.3	2720.6
94	869.7	2876.7
100	1204.2	3099.7

Figure 7: Simulation Output Data of Power consumption – Level 1

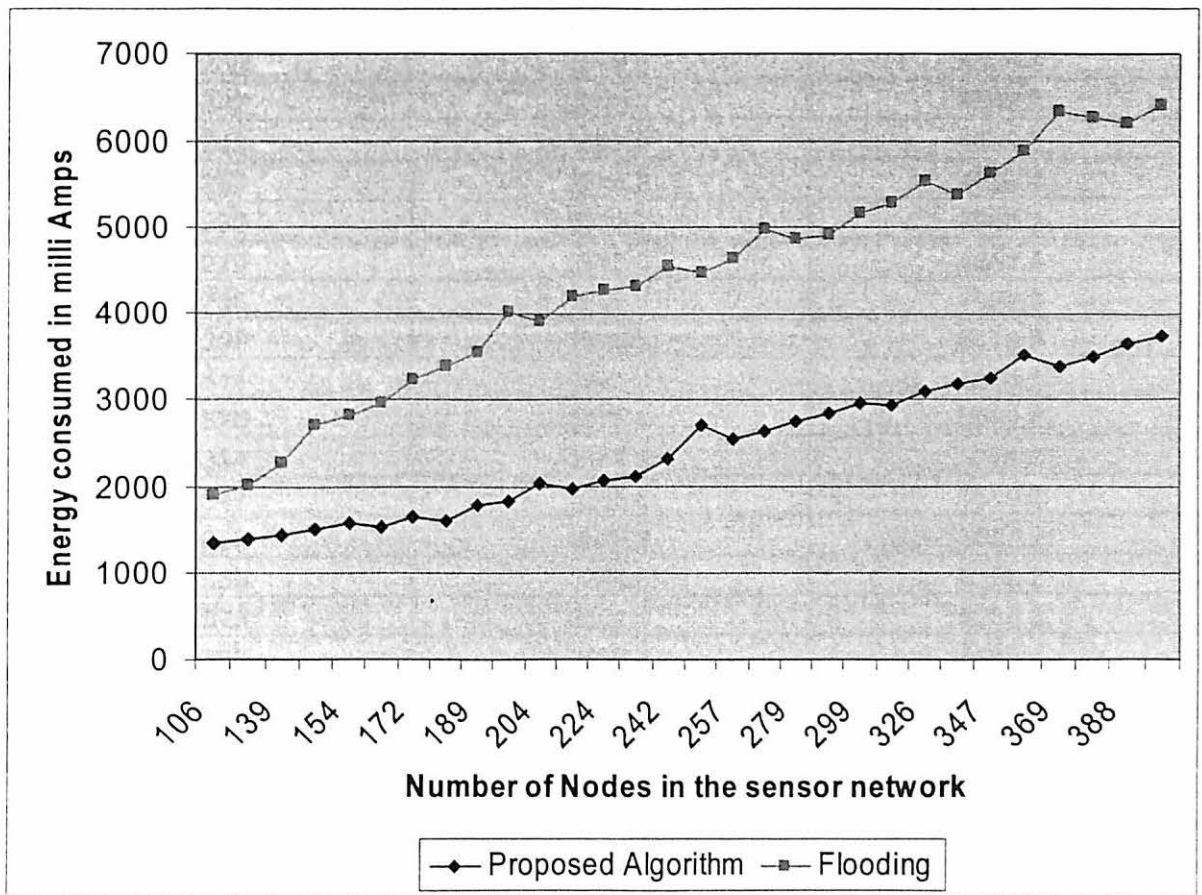


Figure 8: Energy Consumption - Flooding versus proposed algorithm – Level 2

The graph in Figure 8 shows the comparison of energy consumption between the proposed algorithm and the flooding algorithm for the second level of the network. This comparison graph is based on the simulation output data shown in Figure 9.

Number of nodes	Power consumed in mW using Proposed Algorithm	Power consumed in mW using Flooding Algorithm
106	1360.3	1895.5
123	1385.7	2007
139	1449.5	2274.6
148	1516.4	2698.3
154	1583.3	2809.8
164	1538.7	2943.6
172	1650.2	3233.5
177	1605.6	3389.6
189	1784	3545.7
194	1828.6	4036.3
204	2029.3	3902.5
218	1962.4	4214.7
224	2051.6	4281.6
238	2096.2	4326.2
242	2319.2	4549.2
249	2698.3	4482.3
257	2542.2	4638.4
262	2631.4	4995.2
279	2742.9	4861.4
286	2832.1	4928.3
299	2943.6	5173.6
311	2921.3	5285.1
326	3077.4	5530.4
334	3188.9	5374.3
347	3255.8	5619.6
363	3523.4	5887.2
369	3389.6	6333.2
375	3501.1	6266.3
388	3657.2	6199.4
396	3746.4	6400.1

Figure 9: Simulation Output Data of Power consumption – Level 2

In this case, the average power consumption for the two algorithms is as follows:

- a) Proposed Secure Energy Efficient algorithm – 4697.587 mW
- b) Flooding – 8521.477 mW

At level 2 it is observed that the power consumed with flooding is almost twice the amount of power consumed with the proposed algorithm. This shows that

implementation of the proposed algorithm saves a significant amount of energy in a sensor network.

When the two graphs are compared, there is approximately 3:1 ratio of power consumption in the first graph and approximately a 2:1 ratio in the second. As the network size increases, the proposed algorithm has numerous routes for packet transmission whose length may be much longer than shortest paths. So, when flooding is used, one of the paths taken by the flooding routes will be the shortest path and thus the packet reaches the destination faster at which point further flooding stops. As expected, there is extra amount of energy invested in flooding the packets all along different other routes. The fact that the difference in energy consumption is more in smaller networks is that the proposed algorithm computes fairly short routes if not the shortest routes. As the network size increases, the routes become longer, which means more energy is consumed.

The security of the algorithm was validated by measuring the unpredictability of the sequence of routes used. This is justified by the fact that the steps 6 and 7 of the algorithm ensure 2 things:

1. The optimal route is selected based on the length of the route and the number of previous selections of the route, thus making sure that the frequency of use of the computed routes is correctly balanced.
2. At the same time, because this sequence of selection may have repeated patterns like for example with 6 routes and 12 packets for transmission, if the generated pattern is:
3-4-6-1-3-4-6-1-2-5-2-5.

Here, the pattern 3-4-6-1 and 2-5 have repeated. At this point steps 6 and 7 of algorithm ensure that for a particular transmission, the route selection sequence does not have any such patterns. For each transmission, a route from the generated sequence is selected randomly and is removed from the sequence before selecting route for next transmission. The pattern used may therefore be as shown below as generated from a c++ simulation program:

1-3-2-4-3-6-5-1-6-4-5-2.

This clearly does not have any predictable patterns.

Thus the optimal set of routes for the transmission of packets is ensured and at the same time the predictability of routes is reduced by randomly selecting the routes for the message. This ensures fewer opportunities for an attacker to eavesdrop and intrude.

We next evaluated the routing overhead using our protocol. Figure 10 and Figure 11 show that at all the times, the average number of hops on the shortest path is smaller than the average number of hops on a route obtained by the proposed algorithm. But this has always been expected as the penalty to be paid for routing security. A strongly equipped adversary can easily compute the shortest path which means routing using the shortest path is not advised when security is one of the primary concerns in any sensor network application.

This overhead is significant as the average path size increases. At level 1 (Figure 10), with the proposed algorithm, the average of all the nodes on all sizes of the networks is about 1.53 times larger than the average with shortest path algorithm. At level 2 (Figure 11), the proposed algorithm's same average increases to about 2.25 times the average with the shortest path algorithm.

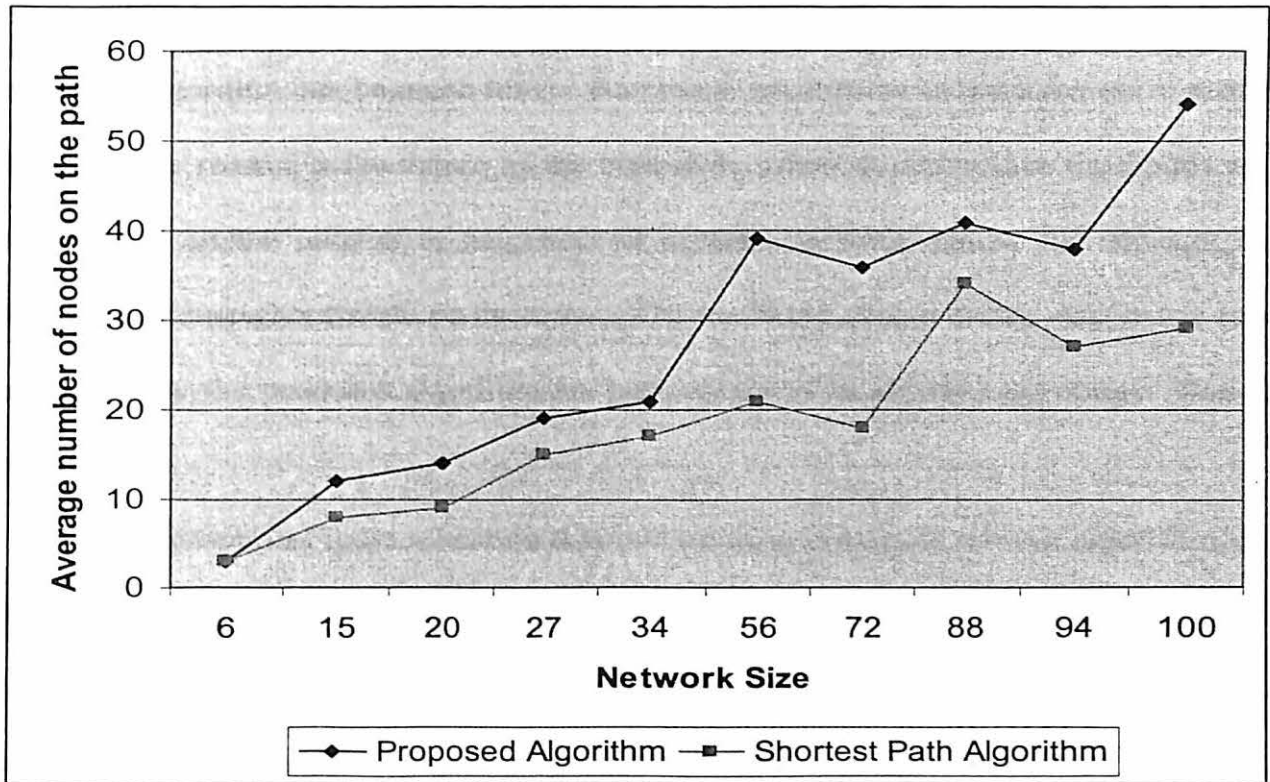


Figure 10: Number of hops on a route comparison – Level 1

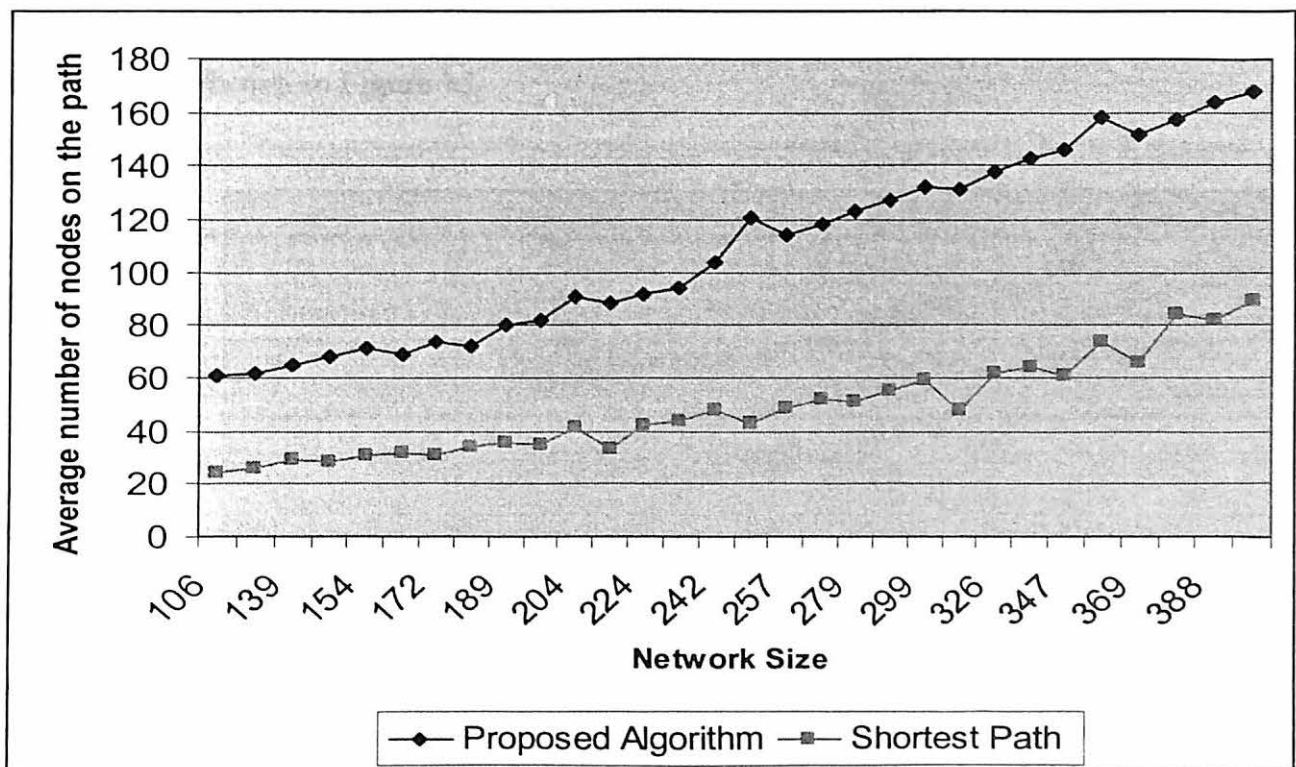


Figure 11: Number of hops on a route comparison – Level 2

This happens because at larger network sizes, the routes calculated by the proposed algorithm can be much longer than the shortest paths as there are more routing choices. The reason is the nature of the algorithm, where it determines the nodes on a route based on the number of neighbors of a particular node and as the network size increases, the neighbor node set increases. The increased choice means that many route calculated by the proposed algorithm are more likely to be significantly longer than the shortest path.

The graph in Figure 12 shows that in case of shortest path routing algorithm, it is always only one route that is available for any packet transmission from a particular source to a destination. But in case of the proposed algorithm, there will be multiple options of routes available for each packet transmission, out of which one is selected based on an efficient criterion. This enhances security. A table listing the number of routes available for various network sizes as computed and used by the proposed routing algorithm is shown in Figure 13.

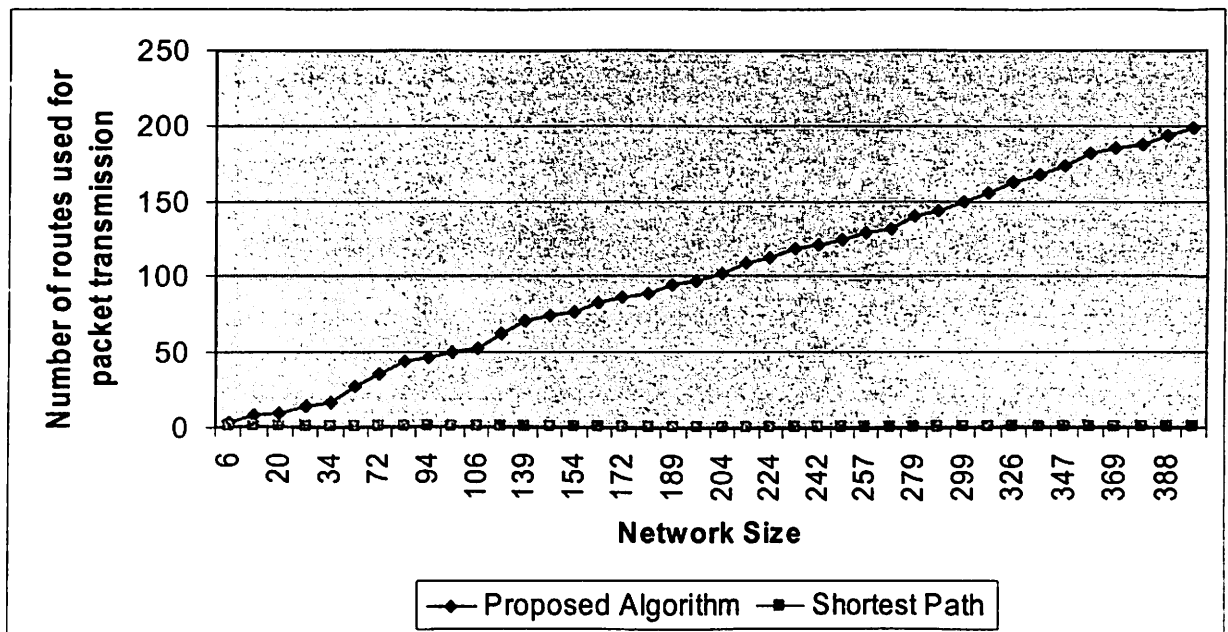


Figure 12: Number of routes used – A comparison

Number of nodes	Number of routes computed and used for routing
6	3
15	8
20	10
27	14
34	17
56	28
72	36
88	44
94	47
100	50
106	53
123	62
139	70
148	74
154	77
164	82
172	86
177	89
189	95
194	97
204	102
218	109
224	112
238	119
242	121
249	125
257	129
262	131
279	140
286	143
299	150
311	156
326	163
334	167
347	174
363	182
369	185
375	188
388	194
396	198

Figure 13: Number of routes used by the proposed algorithm

Figure 15 and 16 are interesting comparisons between the shortest path algorithm and the proposed algorithm. Here there is a mobile adversary node that tries to intercept the packet transmission in the region of our sensor network. This mobile adversary keeps moving in the network region trying to intercept the sensor message information similar to a worm hole or a sink hole as described in [3] and [13]. The adversary when successful in intercepting a message by placing itself somewhere in the middle of two nodes doing a packet transmission stays there, intercepting all the information between them.

At regular time intervals the adversary reconfirms continual packet transmission in the path it has intruded. In the absence of any continued transmission between the nodes, it starts moving again, trying to find another susceptible link. This scenario is modeled using a simulation program and the patterns and time period during which the adversary is successful or not is plotted as graphs in figure 15 and figure 16. Figure 15 depicts the situation where there is only one mobile adversary and figure 16 when there are two mobile adversaries.

The time or rate at which an adversary can detect or intercept the sensor nodes for tracking sensitive information depends on how powerful it is and how fast the adversary can move.

In our simulation program the adversary node moves randomly in the vicinity of the sensor network. It makes movements at regular time intervals and the pattern of its movement will be random. Whenever the adversary changes its position in the network, it links itself to the two nearest nodes and tries to detect whether there is any information transmission between these nodes as shown in Figure 14.

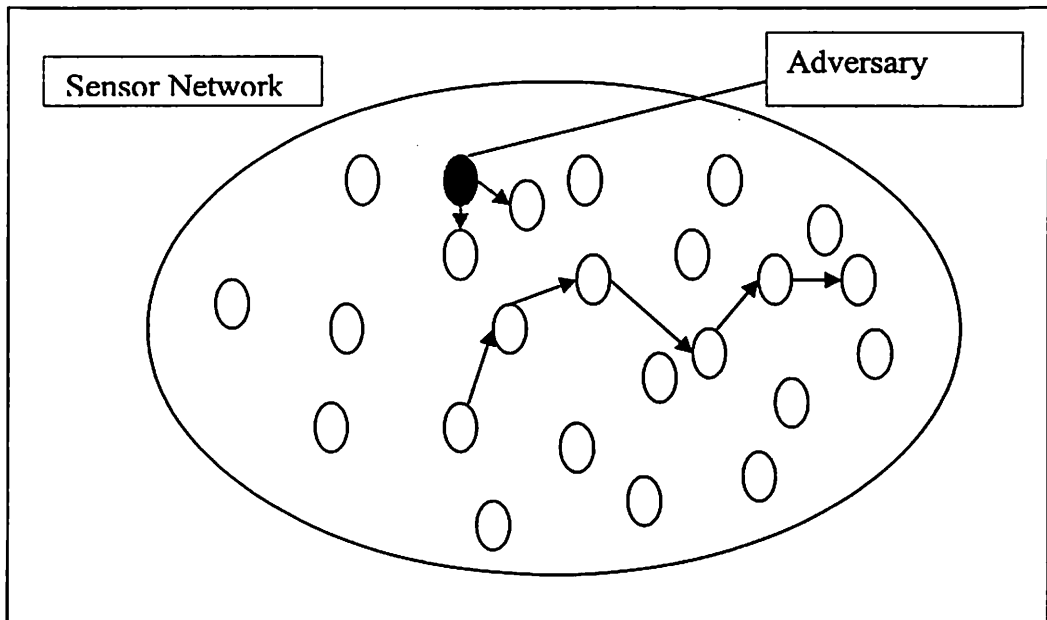


Figure 14: Adversary trying to create a faulty link between two nodes

If it does not find any packet traffic between the two nodes, it continues changing its position until it finds one. The adversary node once it has detected packet transmission between two nodes, continues to stay there and intercept all subsequent packets. Thus, in case of shortest path routing, once a packet transmission is compromised, all the rest of the packets transmitted from that instance on will be intercepted by the adversary.

However, in case of the proposed algorithm, it is highly difficult for an adversary to track down the pattern of paths used for packet transmission. Even if intermittently, an adversary has successfully placed itself between two nodes performing a packet transmission, it will not be able to make any further success based on that, because of the probability factor involved in the routing algorithm.

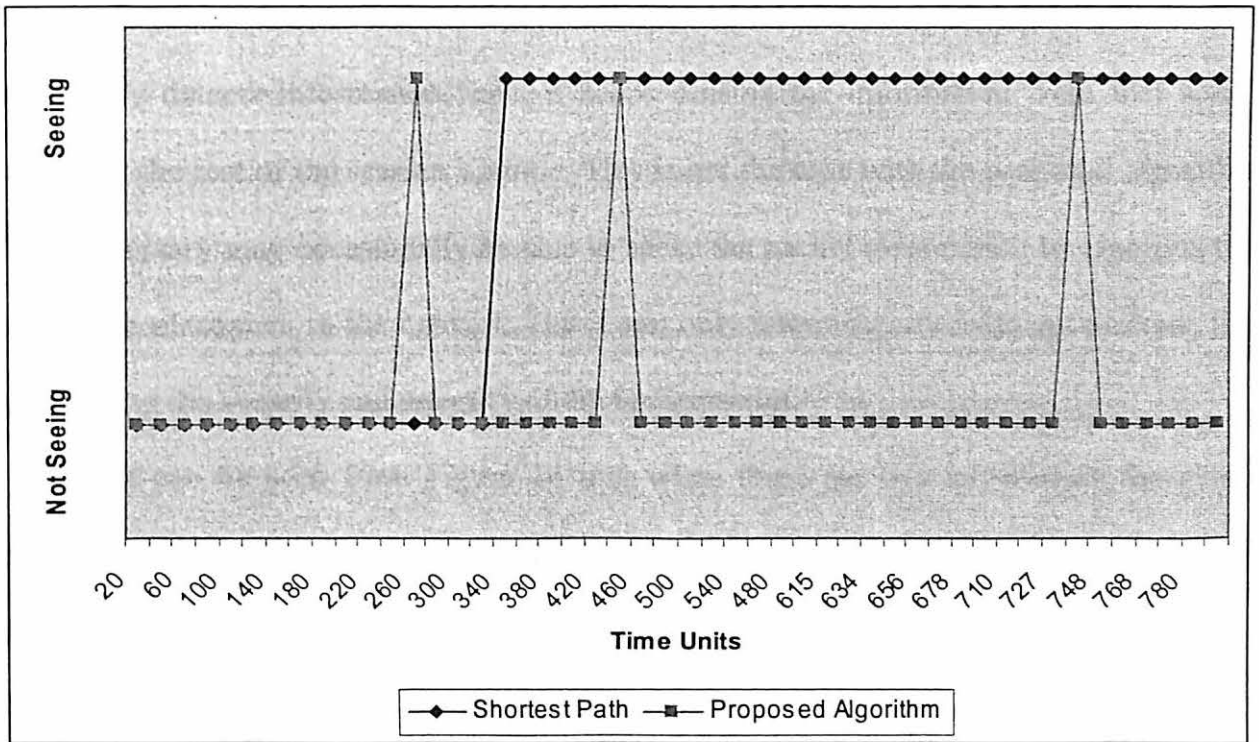


Figure 15: Visibility of sensor message packets to the adversary – 1 adversary

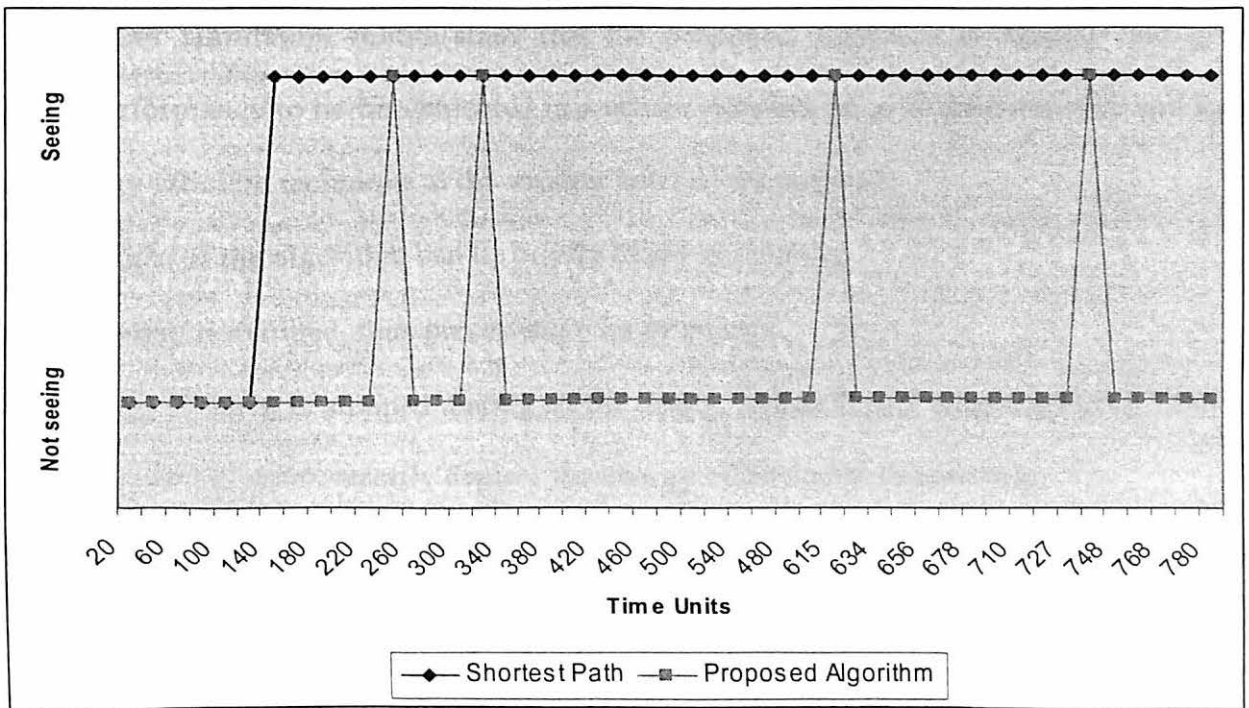


Figure 16: Visibility of sensor message packets to the adversary – 2 adversaries

Figures 15 and 16 show that in the case of the shortest path algorithm, once the adversary detects information flow, it keeps sensing the information from that source node for the rest of the session lifetime. This is not the case with the proposed algorithm. The adversary may occasionally be able to sense the packet information by rigorous trial and error placement in the network. But it can only intermittently intercept packets, thus preserving the security and integrity of the transmission.

It can be seen from Figure 16 that when there are two adversaries the sensor message information can be intercepted at a faster rate as expected. However, as the graphs show, the interception is much lower when the proposed algorithm is used for routing.

5.5 Results

Our simulation results show that the proposed algorithm is feasible and gives good performance to be implemented in a sensor network so as to gain security and have an energy efficient technique at the routing level of the network.

Advantages of the algorithm can be briefly listed as follows:

- Flooding is avoided, thus preserving a lot of energy.
- Energy balance is attained among all the nodes, which favors longer network lifetime and does not unnecessarily deplete the energy of one node excessively.
- The algorithm supports the incorporation of Identity Verification and Authentication suggested in *Sybil attack/ HELLO flood attacks*' countermeasures in Chapter 4.
- The stochastic nature of the algorithm sums up for good security levels.

- When used with SNEP and μ TESLA, might give reasonably good performance of overall data and routing protection.
- The proposed algorithm can be easily extended to Ad-hoc networks with little changes.
- The proposal algorithm is application independent and thus may be deployed for any sensor network or may be modified to suit any specific sensor application model.

Chapter 6

CONCLUSIONS

6.1 Conclusions

As with any other type of networks, communication security is an important factor to consider in sensor network routing schemes. In addition to security, since sensor nodes may be installed in locations where they cannot be easily replenished with battery power, intelligent power savings is also an important factor. This thesis investigates the problems of power consumption and security in sensor networks with respect to routing. A secure, energy efficient routing algorithm is proposed and is tested for feasibility and performance. The proposed algorithm is effective for security and is energy efficient in a sensor network.

6.2 Future work

The techniques used in the algorithm can be applied to application dependent routing algorithms, combined with security mechanisms like μ TESLA and SNEP at MAC layer of the sensor network to obtain optimal security and power savings in a real time sensor application.

REFERENCES

- [1] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar, “*SPINS: security protocols for sensor networks*”, Proceedings of the seventh annual international conference on Mobile computing and networking, July 2001, Rome, Italy.
- [2] Amitabh Mishra, Ketan M. Nadkarni, “*Security in Wireless Ad Hoc Networks*”, Chapter 25, “The handbook of Adhoc wireless networks”, 2003.
- [3] Chris Karlof and David Wagner, “*Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures*”, First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.
- [4] C. Intanagonwiwat, R. Govindan, and D. Estrin, “*Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks*”, Proceedings of ACM MobiCom 00, Boston, MA, 2000.
- [5] C. Shen, C. Srisathapornphat, and C. Jaikaeo, “*Sensor Information Networking Architecture and Applications*”, IEEE Pers. Commun., Aug. 2001.
- [6] <http://www1.cs.columbia.edu/~abk2001/SIMPSON.html>
- [7] <http://www.cs.bu.edu/brite/>
- [8] <http://www.cs.bu.edu/brite/download.html>
- [9] <http://www.cs.bu.edu/lists/brite-users/2003/000051.html>
- [10] <http://www.planetanalog.com/about.html>
- [11] <http://www.planetanalog.com/news/OEG20030714S0049>
- [12] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci; “*A Survey on Sensor Networks*”; IEEE Communications Magazine, August 2002.

- [13] J. P. Hespanha and S. Bohacek, "*Preliminary results in routing games*", in Proc. of the 2001 American Control Conference, June, 2001.
- [14] J. R. Douceur, "*The Sybil Attack*", in 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), March 2002.
- [15] Qun Li, Javed Aslam, Daniela Rus, "*Hierarchical Power-aware Routing in Sensor Networks*", Department of Computer Science, Dartmouth College.
- [16] Rivest, Ronald L., Adi Shamir, Leonard M. Adleman, "*A method for obtaining digital signatures and public-key cryptosystems*", Communications of the ACM, 1978.
- [17] R. L. Rivest, "*The RC5 encryption algorithm*", Proc. 1st Workshop on Fast Software Encryption, 1995.
- [18] Sasha Slijepcevic, Miodrag Potkonjak, Vlasios Tsiatsis, Scott Zimbeck, Mani B. Srivastava, "*On communication security in wireless Ad-Hoc Sensor Networks*", Proceedings of 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002.
- [19] S. Hedetniemi, and A. Liestman, "*A Survey of Gossiping and Broadcasting in Communication Networks*," *Networks*, vol. 18, 1988.
- [20] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "*Energy-Efficient Communication Protocol for Wireless Microsensor Networks*", IEEE Proc. Hawaii Int'l. Conf. Sys. Sci., Jan. 2000.
- [21] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "*Adaptive Protocols for Information Dissemination in Wireless Sensor Networks*", Proceedings of ACM MobiCom, Seattle, WA, 1999.

- [13] J. P. Hespanha and S. Bohacek, "*Preliminary results in routing games*", in Proc. of the 2001 American Control Conference, June, 2001.
- [14] J. R. Douceur, "*The Sybil Attack*", in 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), March 2002.
- [15] Qun Li, Javed Aslam, Daniela Rus, "*Hierarchical Power-aware Routing in Sensor Networks*", Department of Computer Science, Dartmouth College.
- [16] Rivest, Ronald L., Adi Shamir, Leonard M. Adleman, "*A method for obtaining digital signatures and public-key cryptosystems*", Communications of the ACM, 1978.
- [17] R. L. Rivest, "*The RC5 encryption algorithm*", Proc. 1st Workshop on Fast Software Encryption, 1995.
- [18] Sasha Slijepcevic, Miodrag Potkonjak, Vlasios Tsiatsis, Scott Zimbeck, Mani B. Srivastava, "*On communication security in wireless Ad-Hoc Sensor Networks*", Proceedings of 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002.
- [19] S. Hedetniemi, and A. Liestman, "*A Survey of Gossiping and Broadcasting in Communication Networks*," *Networks*, vol. 18, 1988.
- [20] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "*Energy-Efficient Communication Protocol for Wireless Microsensor Networks*", IEEE Proc. Hawaii Int'l. Conf. Sys. Sci., Jan. 2000.
- [21] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "*Adaptive Protocols for Information Dissemination in Wireless Sensor Networks*", Proceedings of ACM MobiCom, Seattle, WA, 1999.

- [22] Y. C. Hu, A. Perrig, and D. B. Johnson, "*Wormhole detection in wireless ad hoc networks*", Department of Computer Science, Rice University, Tech. Rep. TR01-384, June 2002.
- [23] Y. Yu, R. Govindan, and D. Estrin, "*Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks*", University of California at Los Angeles Computer Science Department, Tech. Rep. UCLA/CSD-TR-01-0023, May 2001.

VITA ①

Malleswari Akkineni

Candidate for the Degree of

Master of Science

Thesis: SECURE, ENERGY EFFICIENT ROUTING ALGORITHM FOR SENSOR NETWORKS

Major Field: Computer Science

Biographical:

Personal Data: Born in Nuzvid, Andhra Pradesh, India, on January 21, 1980, the daughter of Mr. Venkateswara Rao Akkineni and Mrs. Lakshmeswari Akkineni.

Education: Graduated from High School, Nalanda Institutions, Vijayawada, India in June 1997; received Bachelor of Technology degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in June 2001. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University, Stillwater, USA in May, 2004.

Experience: Employed by Oklahoma State University, Department of Computer Science as a graduate research assistant from January 2003 to June 2003. Employed by Autodesk Inc., as a programmer intern during summer 2003. Employed as a graduate Teaching Assistant by Department of Computer Science Oklahoma State University from August 2003 to present.