RESPONDING TO AN ATTACK IN SENSOR NETWORKS

By

BALAMBIKA VINOD

Master of Science in Computer Science

Oklahoma State University

Stillwater, Oklahoma

2012

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, 2012

RESPONDING TO AN ATTACK IN SENSOR

NETWORKS

Thesis Approved:

Dr Johnson Thomas

Thesis Adviser

Dr. Debao Chen

Dr.K.M.George

Dr. Sheryl A. Tucker

Dean of the Graduate College

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Wireless Sensor Networks (WSNs) consists of spatially distributed autonomous sensors used to monitor physical or environmental conditions such as temperature, sound, vibrations, etc. co-operatively [1]. These sensors are finding applications in diverse fields such as in the battlefield, healthcare, space etc. A WSN consists of collections of nodes and a base station. The sensors, which are miniature computers, have very basic functionalities with interfaces and components. The base station acts as a gateway between the sensor nodes and the end users, forwarding the data collected from the sensor nodes to a server. It has much more computational energy and communication resources than that of an individual sensor node.

The network is assumed to be a large static wireless sensor network, where the resources of all the sensor nodes are identical. The sensor nodes use multi-hop routing to communicate with the base, as they can only communicate directly with their neighboring nodes. The routing protocols for sensor networks are distributed and reactive. One such protocol used in our work is the Ad-hoc On-demand Distance Vector (AODV) routing protocol, which was initially designed for wireless ad-hoc networks, but has been used to meet the specific needs of sensor networks [2].

The sensor nodes may vary in size and the cost is similarly variable depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The basic philosophy behind WSNs is that, while the capability of

each individual sensor node is limited, the aggregate power of the entire network is sufficient for the required mission.

1.1 Security in WSNs

The simplicity of Wireless Sensor Networks with resource constrained nodes makes them extremely vulnerable to variety of attacks. Attackers can eavesdrop on radio transmissions, inject bits in the channel, replay previously heard packets and do much more. Securing WSNs is crucial as they may operate in a hostile environment. Securing the Wireless Sensor Network needs to make the network support all security properties: confidentiality, integrity, authenticity and availability [3]. But the constrained computation and communication capability of sensor nodes make the use of many traditional security methodologies such as encryption, complex cryptographic techniques, key management, etc., difficult or impossible to use.

Attackers may deploy a few malicious nodes with similar hardware capabilities as legitimate nodes that might collude to attack the system cooperatively. Also, in some cases colluding nodes might have high-quality communications links available for coordinating their attack. Sensor nodes may not be tamper resistant and if an adversary compromises a node, it can extract all key material, data, and code stored on that node [4]. Extremely effective tamper resistance tends to add significant per-unit cost, and sensor nodes are intended to be very inexpensive.

Some attacks such as wormhole attacks, where a pair of colluding attackers record packets at one end and replay them at another location using a private high speed network, pose a severe threat against packet routing that is particularly challenging to detect and prevent, as they use a private, out-of-band channel invisible to the underlying sensor network. The wormhole attack can severely deteriorate the performance and compromise the security of a sensor network through exploiting the routing protocols. These attacks are immune to cryptographic techniques,

as the attacker does not need to decode encrypted packets to be able to replay them. It is even more difficult to defend against when combined with sinkhole attacks, where nearly all the traffic from a particular area is lured through a compromised node, creating a sinkhole with the adversary at the centre [5].

In a wireless sensor network, some sensor nodes might want to communicate with the base station in order to send data/information. A pair of attackers, with their high speed private wormhole link, can create an illusion that these sensor nodes are only a few hops away from the base station. Thus the sensor nodes will eventually choose this shortest path through the attacker to send data packets to the base. Thus all traffic from these sensor nodes will be lured towards the attackers, who thereby gain access to important data flowing through the network.

Thus in the presence of a wormhole attack, our objective to protect the network is two-fold:

1. To find an alternate path for the sensor nodes that wants to communicate with the base, while avoiding the wormhole;
2. To waste the attacker's resources

The network is protected by responding to an attacker by retaliating in some form. The objective is two-fold as identified above. An appropriate response can take many forms. For example, if one part of the network is collecting critical data, the objective will be to prevent the attacker from moving to that part of the network, while inflicting some cost on the attacker (such as wasted energy). A more brute force type of retaliation would simply try to flood the attacker's area of the network so that the attacker enters a Denial of Service mode and is not able to function properly. It is important to note that our approach is a complement to more traditional security mechanisms and is not a substitute for them.

1.2 Proposed Approach

The network consists of three types of nodes, besides the attacker. There are a large numbers of tiny sensors which are very resource constrained in terms of processing, computation and storage. Secondly, Distributed Retaliation Agents (DRAs) which although fewer in number, are more powerful connecting or switching devices such as Stargates [6] or FitPCs [7]. These are typically used in sensor networks to interconnect sensor clusters for example. There is also a single powerful base station. We assume that the wormhole attack has been identified by the intrusion detection system, and that the attacker cannot change the underlying AODV routing protocol.

AODV is a multi-hop routing protocol where sensor nodes can only communicate with its neighbors directly. A node broadcasts route requests (RREQ) to discover a route to the base to send data. The base responds with a route reply (RREP) for the first RREQ from a node, which is unicasted back to the requesting node, in the same route that the RREQ reached the base. The node sends data through the route for which it received a RREP. With the high speed wormhole link, the RREP reaches the source first through the attacker route. Thus the source ends up sending data through the attacker, who thereby gains access to the data.

The retaliation takes place in three phases.

Phase I: Prepare Environment

Once the attack has been identified, the environment is prepared for retaliation by DRAs. The candidate nodes which are capable of starting the retaliation process must be within the communication range of the attackers. An overlap density based selection method is used by the DRA to select the nodes that actually perform the retaliation called sacrificial nodes. This selection should be hidden from the attacker; thus DRA should not directly communicate with the attacker. This means that the nodes between the DRA and the attacker are two-hop neighbors. If

the distance between the DRA and the attacker gets shorter, the intersection area between the two will get bigger. This increases the probability that there is a sacrificial node in the intersection area A.

As the sacrificial nodes have limited power source, their effective working area of the sacrificial nodes will shrink over time. Thus the DRA should add more sacrificial nodes if the probability that the attacker is within the communication range of at least one sacrificial node falls below a set threshold.

Phase II: Respond

The wormhole attack is a sophisticated attack where the attacker creates a fake path to the destination (base station) to gain access to the data. We assume that the attacker cannot modify the underlying AODV protocol. There are two types of nodes in the network: one that communicates normally with the attacker sending packets (Type A) and while the other is sacrificial nodes that were not communicating, but sends a RREQ packet to the attacker after the attack is detected and receives a RREP (Type B) packet in return.

With the high speed wormhole link, that attacker can make the RREP from the base to reach the defender node D1 first. A response is successful if the sacrificial nodes can find an alternate path, different from the attacker route, for the defender node to send an actual data packet to the base. To achieve this, we need to find the probability that a given RREP through the attacker route is intended for the defender, with the assumption that he can handle all packets, represented as $P(D1_{RREP}|PA)$, where PA is the number of packets flowing through the attacker and $D1_{RREP}$ is the route reply packet for the defender. The sacrificial nodes should aim to saturate the capacity/bandwidth of the attacker, so that $P(D1_{RREP}|PA)$ decreases, as the attacker gets busy processing the fake RREQs from the sacrificial nodes.

As the sacrificial nodes are not aware of the attacker's capacity, the number of sacrificial nodes may have to be increased in order to overload the attacker. A congestion estimator is used to monitor the capacity of the attacker at any time. If the bandwidth is low, the attacker is overloaded and $P(D1_{RREP}|PA)$ decreases. If the bandwidth is very high, then the attacker is not overloaded. This means that $P(D1_{RREP}|PA)$ is high and hence the number of RREQs or the sacrificial nodes has to be increased.

After the number of sacrificial nodes has been increased, the bandwidth is calculated again to check if even more sacrificial nodes are needed. The probability is calculated again at the end of a time window whose size is equal to the time of flight measure, as the sacrificial nodes aim to find an alternate path within the lifetime (round trip time from sending of RREQ and the arrival of RREP) of the packet.

The outline of the rest of the thesis is given as follows:

In chapter 2, the literature review is presented. The actual working of the AODV routing protocol is discussed in chapter 3. The outline for our proposed approach is given in chapter 4. Chapter 5 elaborates the method proposed to respond to a wormhole attack, in particular we present algorithms to prepare the environment for retaliation and algorithms for appropriate response to a wormhole attack. Chapter 6 presents simulation results to validate our approach and the thesis concludes in chapter 7.

CHAPTER II

REVIEW OF LITERATURE

2.1 Security in Sensor Networks

Research on providing security solutions for WSNs has focused mainly in four categories:

1) Key management:

A lot of work has been done [8] in establishing cryptographic keys between nodes to enable encryption and authentication. But these methods are complex and computationally too intensive, depleting the limited energy available to the sensor nodes.

2) Authentication and Secure Routing:

Several protocols [9] have been proposed to protect information from being revealed to an unauthorized party and guarantee its integral delivery to the base station. Marti et al. [10] and Buchegger and Boudec [11] consider the problem of minimizing the effect of misbehaving or selfish nodes on routing through punishment, reporting, and holding grudges. The application of these techniques to sensor networks is promising, but these protocols are vulnerable to blackmailers.

Perrig et al. [12] propose SPINS: Secure routing protocols for WSN, which has two building block security protocols optimized for use in sensor networks, SNEP and µTESLA.

SNEP provides confidentiality, authentication, and freshness between nodes and the sink, and μTESLA provides authenticated broadcast.

3) Secure services:

In a secure localization technique (SerLoc) [13] nodes can use the location information broadcasted by guards to determine their own position following some principles. But its functionality relies on the correct operation of the distinguished guard nodes, and an attacker can easily bypass the defense mechanism by compromising these guards.

A secure aggregation protocol for cluster-based WSN has been proposed in [14]. Aggregation can be seen as the process by which data sent from sensors to the base station are little-by-little processed by some nodes called aggregator nodes. Aggregators collect data from surrounding nodes and produce a small sized output, thus preventing all nodes in the network from sending their data to the base. This protocol does not rely on trusted aggregator nodes and thus is immune to aggregators compromising. In addition to security performance, it has a low transmission overhead.

A secure time synchronization toolbox has been proposed in [15] to counter attacks. This toolbox includes protocols for secure pair-wise and group synchronization of nodes that lie in each other's power ranges and of nodes that are separated by multiple hops.

4) Intrusion Detection:

Da Silva *et al.* [16] and Onat and Miri [17] propose similar IDS systems, where certain monitor nodes in the network are responsible for monitoring their neighbors, looking for intruders. They listen to messages in their radio range and store in a buffer specific message fields that might be useful to an IDS system running within a sensor node, but no details are given how this system works. In these architectures, there is no collaboration among the monitor nodes.

Loo et al. [2] and Bhuse and Gupta [18] describe two more IDSs for routing attacks in sensor networks. Both papers assume that routing protocols for ad hoc networks can also be applied to WSNs: Loo et al. [2] assume the AODV (Ad hoc On-Demand Distance Vector) protocol while Bhuse and Gupta [18] use the DSDV and DSR protocols. Then, specific characteristics of these protocols are used like "number of route requests received" to detect intruders.

2.2 Security for Wormhole Attack in Sensor Networks

Packet leashes proposed by Hu et al [19] are the most commonly cited wormhole prevention mechanisms. The general idea is to add a secure constraint (leash), such as timing or location information, to each packet. This information acts as the metric for calculating whether the packet traveled a distance larger than physically possible. This needs a special hardware for localization and synchronization. However the use of technology like GPS significantly limits the life expectancy of such a network with resource constraints.

Capkun et al [20] and have proposed authenticated distance bounding protocols for ensuring that nodes claiming to be located close together really are. They calculate the distance between two participating nodes by measuring the round trip travel time of a challenge (message and its acknowledgement) and determine whether the estimated distance is within the maximum possible communication range. The main problem is that the wireless medium introduces random delays between the time a packet is sent and the actual time it is transmitted via the radio interface.

Lazos et al [21] also proposed a solution for calculating the distance between a pair of nodes based on the location information of a few nodes called guards. The basic idea is that nodes can use the location information broadcasted by the guards to determine their own position following some principles. This approach is showed to successfully detect a wormhole with

probability close to one. However, its functionality relies on the correct operation of the distinguished guard nodes, and an attacker can bypass the defense mechanism by compromising these guards.

Hu et al [22] have proposed a solution in which all nodes are equipped with directional antennas. Each node has to examine the direction of received signals from its neighbor. Thus a pair of nodes can participate in a cooperation scheme where they share directional information to prevent wormhole endpoints from masquerading as false neighbors. There is also a witness node to confirm the above said relation. This can be applied only to networks that use directional antennas. Also the presence of this witness node can lead to the failure of the protocol since it can be easily compromised by an attacker.

Wang et al [23] have proposed an approach in which each sensor estimates the distance to its neighbors using the received signal strength. All sensors send this distance information to the central controller, which calculates the network's physical topology based on individual sensor distance measurements. If the estimated distance between the two nodes connected by a wormhole is much larger than the node's communication range, it leads to the detection of the attack in progress. This technique is susceptible to distance estimation errors especially for sparsely located network nodes. Also, its centralized nature limits its applicability in sensor networks since most of the times we are dealing with networks that operate in unattended environments.

Khalil et al [24] have proposed a protocol called LiteWorp for wormhole attack discovery in static networks. In LiteWorp, once deployed, nodes obtain full two-hop, rather than one-hop, routing information from their neighbors. This information can be exploited to detect a wormhole. Also, nodes observe their neighbor's behavior to determine whether data packets are

being properly forwarded by them. This method is impractical and only applicable to static stationary networks.

Song et al [25] have proposed a wormhole discovery mechanism based on statistical analysis of multipath routing. They observed that a link created by a wormhole is very attractive in routing sense, and thus will be selected and requested with high frequency. These factors allow for easy integration of this method into intrusion detection systems. But this method can only work with protocols that are both on-demand and multipath.

Ritesh et al [27] have proposed a detection algorithm that uses only connectivity information to look for forbidden substructures in the connectivity graph. It is completely localized and, unlike many techniques proposed in the literature, does not use any special hardware artifact or location information, making the technique universally applicable. It is also independent of wireless communication models. This approach provides a sufficiently high detection probability in connected networks, but does not guarantee the detection of wormhole in all cases.

CHAPTER III

AODV ROUTING PROTOCOL

The AODV (Ad-hoc On-Demand Distance Vector) routing protocol is a dynamic, self-starting, multi-hop reactive protocol between mobile nodes wishing to establish and maintain an ad hoc network. AODV does not need the participating mobile nodes to store the routes that are not in active communication on the routing table. It responds to link breakages and changes in the network topology over time and thus adapts to varying ad-hoc network needs. AODV is an efficient reactive routing protocol that broadcasts only route discovery packets and thus works well with a high node density mobile network. Like all the other routing protocols, AODV is also vulnerable to attacks and to increase its reliability as a protocol, the simplest solution is to add sophisticated algorithms to the nodes using the protocol [26].

AODV maintains connection only to the immediate neighbors broadcasting HELLO messages periodically. If a node fails to receive a HELLO message from a neighbor, then a link break is detected. The message packets are of three types, RREQ, RREP and RERR. They are used for broadcasting route requests, sending or receiving route replies and for reporting error messages respectively.

A RREQ packet consists of the following fields:

<Source address, source sequence number, broadcast id, destination address, destination sequence number, hop count>

The node seeking a route to the destination is called the requesting node. The requesting node initially broadcasts a RREQ to all of its adjacent nodes once, to find a route to the destination node. The node receiving the RREQ can either be the destination node or an intermediate node. If the node has not received this RREQ before, is not the destination node and does not have a current route to the destination node, then this node rebroadcasts the RREQ to its set of neighbors. On the other hand, if the node receiving the RREQ is the destination node or has a current route to the destination in its route table, then this node generates a RREP message, which is unicasted to the requesting node in a hop-by-hop fashion. A RREP packet has the following fields:

<Source address, destination address, destination sequence number, lifetime, hop count>

When a RREQ is received at each intermediate node, a reverse path is created to the requesting node. When the destination node is found, the RREP will travel through this path, thus eliminating the need for any more broadcasts. While the RREP message propagates, a forward path is set up, which the requesting node uses to send its data packets to the destination. This includes storing the destination IP address, source IP address, broadcast id, expiration time for reverse path route entry and source node's sequence number. A route is determined when the RREQ reaches either the destination itself, or intermediate nodes with a 'fresh enough' route to the destination. The RREQ's are rebroadcasted only till the 'lifetime' is over. If a destination is not found by then, the source node increases the lifespan and repeats the process. Nodes that are not along the path are determined by the RREP reverse path and the pointers are deleted after an ACTIVE_ROUTE_TIMEOUT of 3000msec.

If the data is flowing through the network and a link break is detected, a RERR message is sent to the adjacent nodes in a hop-by-hop fashion. A RERR packet has the following fields:

< Unreachable Destination IP address, Unreachable destination sequence number, destCount>

Each intermediate node invalidates routes to any unreachable destination. When the requesting node receives this RERR message, it invalidates the route to destination and reinitiates route discovery if necessary. A node can broadcast a RERR in 3 situations:

a. When it receives a data packet, but does not have a route to the destination – the real problem is that some other node thinks that the correct route is through this node.
b. A node receives a RERR message for a node which is in its route table – then this node sends a RERR with all the new nodes which are now unreachable.
c. A link break is detected with the node's neighbor – the node checks the route table for routes that use the neighbor as the next hop and marks them invalid and then sends a RERR with the neighbor and the invalid routes.

Sequence numbers are the most important feature of AODV. They serve as timestamps and removes old and invaluable messages from the network, thus preventing AODV from the "counting to infinity" problem faced by other distance vector routing protocols. They allow nodes to compare how "fresh" their information on other nodes is. Every time a node sends out any type of message it increases its own sequence number. Each node records the sequence number of all the other nodes it talks to. A higher sequence numbers signifies a fresher route. This it is possible for other nodes to figure out which one has more accurate information. The destination sequence number is created by the destination for any route information it sends to requesting nodes. It is stored in the route table and is updated when the requesting node receives the RREP message with a greater sequence number. After forwarding a RREP, the node can get another RREP which is either discarded or forwarded, depending on its destination sequence number:

a. If sequence number of the new RREP is greater than the stored one, it is forwarded.
b. If old RREP sequence number is the same as the new one and the new RREP has a smaller hop count, then the new RREP is forwarded.
c. Else, all later arriving RREPs are discarded

To understand the working of AODV, consider an example of five mobile nodes as shown in Figure1. Node 4 wants to communicate with node 3, but is uncertain of the route. Thus 4 broadcast RREQ, which is received by its neighbors node 5 and node 1. Node 5 does not have a route to node 3. Thus node 5 rebroadcasts RREQ. As node 4 is the only neighbor of node 4, it receives the RREQ, but drops it. If node 1 has a sequence number greater than that in the RREQ, node 1 discards the RREQ. Otherwise, node 1 updates the sequence number in the route table and forwards the RREQ to node 2. Node 2 has a route to node 3; thus replies to node 1 by sending a RREP, and establishes a forward path to node 3. Node 1 forwards the RREP from node 2 to node 4 and establishes a forward path to node 2. Thus the route 4 -> 1 -> 2 -> 3 is confirmed to send data from node 4 to node 3.

Imagine a node 6 in the communication range of Node 1 and Node 2. As shown in Figure 2, node 1 moves out of network. Suppose node 6 detects it first by not getting any HELLO message from node 1and marks the respective route table entry for route as invalid. It sends out an RERR with the invalid route which is received by node 2. This is how node 2 comes to know from node 6 that node 1 is no longer its neighbor.

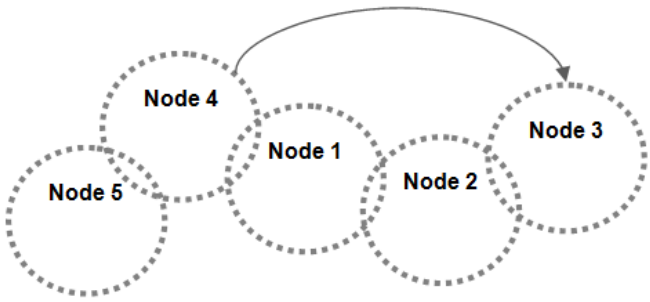

Figure 1. Communication between nodes in a WSN using AODV routing protocol

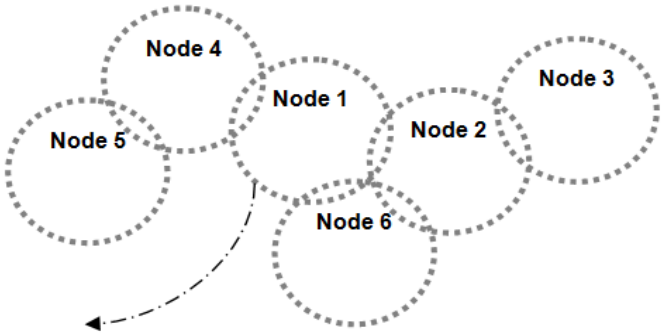

Figure 2. Node1 moves out of communication range

CHAPTER IV

PROPOSED APPROACH

4.1 Problem Statement

The problem is how to respond to an attack such as a wormhole in wireless sensor networks when the sensor nodes have limited resources for complex security mechanisms. It is important to note that the network can respond to an attacker only if the attacker can be identified. The danger is that if an attacker cannot be identified, the response may be directed at a friend rather than a foe. Fortunately intrusion detection techniques exist that can identify the approximate location of an attacker if not pinpoint the attacker. For example, detecting wormhole attacks based on topology information [27] not only detects a wormhole attack, but also identifies the place in the network where the wormhole attack is taking place. Similarly, detecting the sinkhole attack as proposed by [28] places the location of the attacker. Hence, although the approach proposed in this paper cannot be applied if the location in the network of the attacker cannot be identified approximately at least, there are many intrusion detection approaches reported in the literature that identify the location of the attacker [29]. It is beyond the scope of this thesis to investigate the intrusion detection techniques that also identify the location of the attacker. Thus for our work, we assume that an intrusion detection system has already detected the attack and the location of the attack.

After finding the location of attack in the wireless sensor network, our response is two-fold:

- Avoid the attack (the wormhole)
- Deplete attacker resources

4.2 Outline of Proposed Approach

The sensor network should immediately respond to the attacker, once the location of the attack has been identified by the intrusion detection system, before the attacker gains control of the entire network. The steps in the proposed approach are as follows:

a. Set up the network -

A few sensor nodes (sacrificial nodes) are selected by the DRA to perform the retaliation process, while other nodes carry normal communication within the network. Then the retaliation variables and the steps to defend against the attacker are also loaded into the sacrificial nodes by the DRA. The function of the sacrificial nodes is to keep the attacker busy in order to deplete the attacker's resources.

b. Respond -

Once the network is set up in step a, the sacrificial nodes perform the actual retaliation process. The sacrificial nodes start sending a number of Route requests (RREQ packets) to the destination node through the attacker route, in order to keep the attacker busy from processing the actual RREQ from the defender node. The retaliation algorithm includes a prediction component that predicts the number of route replies (RREP) from the attacker route. The faster the replies come back, the more the number of RREQ that have to be sent out by the sacrificial nodes. The objective is that the RREP from the destination node through the attacker route should not reach the defender before the time of flight measure. If it does, then the retaliation is not fully successful; thus a larger number of sacrificial nodes have to be selected and the process be repeated, until the attacker is fully overloaded that he is not able to process all the RREQ from the defender.

c. Detect a change in attacker behavior –

Even before the sacrificial nodes succeed in retaliating against the attacker, the attacker might change his attack pattern. This is not considered in this work.

CHAPTER V

RESPONSE TO A WORMHOLE ATTACK

A Wormhole attack [30] is a sophisticated attack where the attacker creates a fake path to the destination to gain access to the data. It is the technique by which 'a pair of colluding attackers record packets at one location and replay them at another location using a private high speed network. The attackers disrupt routing by 'short circuiting' the routing flow of packets.

Wormhole attack is a kind of replay attack that is particularly challenging in WSN to defend against. Even if the routing information is confidential, encrypted or authenticated, it can be very effective and damaging. An attacker can tunnel a request packet RREQ directly to the destination node without increasing the hop-count value. Thus it prevents any other routes from being discovered. It may badly disrupt communication as AODV would be unable to find routes longer than one or two hops. It is easy for the attacker to make the tunneled packet arrive with better metric than a normal multi-hop route for tunneled distances longer than the typical transmission range of a single hop. Malicious nodes can retransmit eavesdropped messages again in a channel that is exclusively available to attacker.

A wormhole attack commonly involves two remote malicious nodes shown as X and Y in Fig.3 [30]. X and Y both are connected via a wormhole link and they target to attack the requesting node S. During the path discovery process, S broadcasts a RREQ to a destination node D. Thus, A and C, neighbors of S, receive RREQ and forward the RREQ to their neighbors.

Now the malicious node X receives RREQ forwarded by A. It records and tunnels the RREQ via the high-speed wormhole link to its partner Y. Malicious node Y forwards the RREQ to its neighbor B. Finally, B forwards it to destination D. Thus, RREQ is forwarded via S-A-X-Y-B-D. On the other hand, another RREQ packet is also forwarded through the path S-C-D-E-F-G-D. However, as X and Y are connected via a high speed bus, the RREQ from S-A-X-Y-B-D reaches D first. Therefore, destination D ignores the RREQ that reaches later and chooses D-B-A-S to unicast a RREP packet to the requesting node S. As a result, S chooses S-A-B-D route to send data that indeed passes through X and Y malicious nodes that are very well placed compared to other nodes in the network.

Thus, a wormhole attack is not that difficult to set up, but still can be immensely harmful for a WSN. Moreover, finding better techniques for detection of wormhole attacks and securing AODV against them still remains a big challenge in wireless sensor networks.

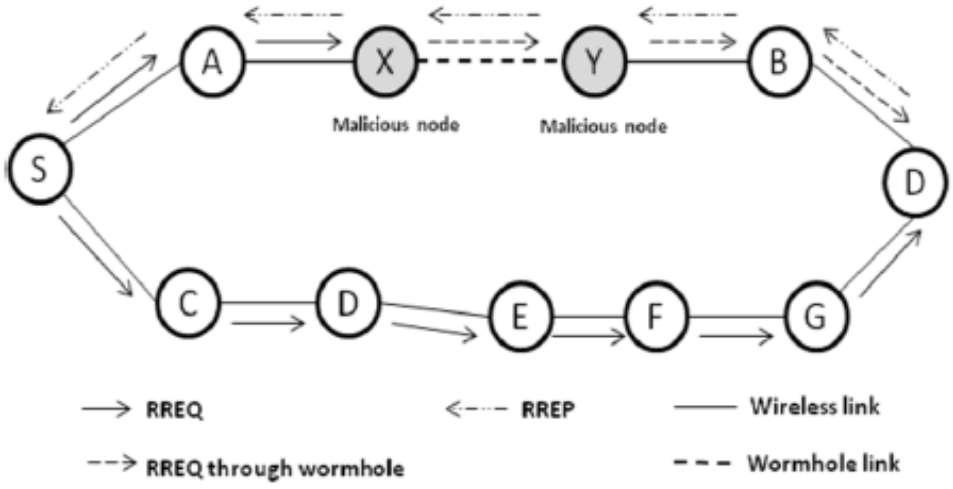

Figure 3. Wormhole attack on AODV in WSN

5.1 Setting up the Network

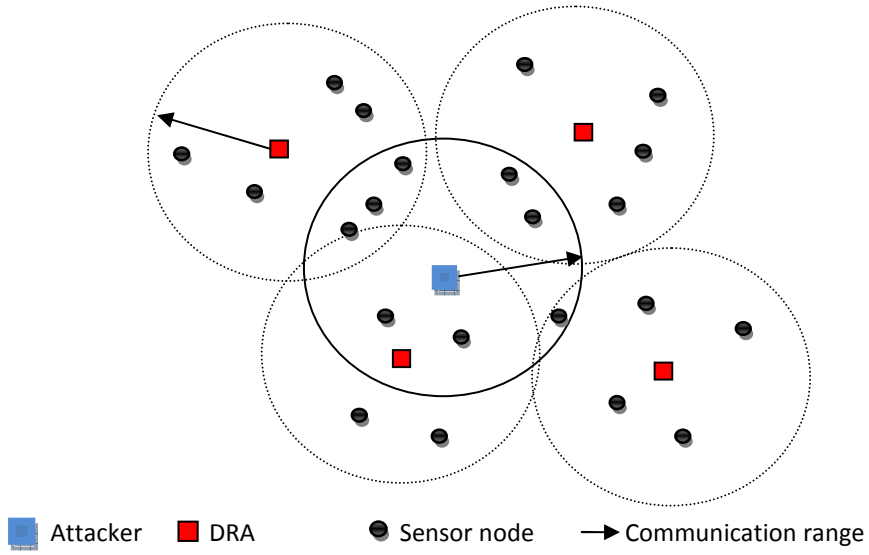

Figure 4. Network model

This research identifies certain nodes to retaliate against the attacker which are based on the following assumptions: It is assumed that the attacker is a sophisticated intelligent attacker and a mobile attacker. It is assumed all nodes have the same communications range.

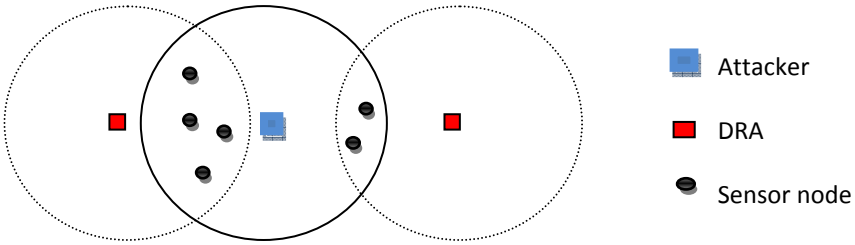

Figure 5. An illustration of Candidate retaliation nodes

Definition 1: *Candidate nodes:* these are nodes within transmission range of the attacker that can do the retaliation. These nodes also belong to the DRA, which can start the retaliation process.

Definition 2: *Sacrificial nodes:* these are the nodes selected from the candidate nodes to ultimately do the retaliation. An overlap density based selection method chooses these nodes from the set of candidate retaliation nodes.

Identifying the ideal number of sacrificial nodes is presented in section 5.3. The response model will use an overlap density based selection method to distribute the nodes to maximize the coverage while retaliation is taking place. The criteria for candidate nodes are specified below.

5.1.1 Criteria for candidate nodes

First, the candidate nodes must be able to monitor attackers and also communicate with the attackers. Second, DRAs must not be able to communicate directly with the attackers, because each DRA has the responsibility of identifying candidate nodes and this communication must be hidden from attackers. Therefore, the monitoring range refers to the two-way communication between the candidate nodes and the attacker node. After selecting the sacrificial nodes, the rest of the nodes in the network excluding the nodes in the attacking area will operate as normal. Other candidate nodes will be set to sleep.

Since it may be difficult and expensive to retaliate against an attacker, our work uses a probabilistic approach to monitor the retaliation process. The probabilistic approach will be calculated based on several indicators. These will include factors such as Link Quality Indicator (LQI), Received Signal Strength Indicator (RSSI), and also the probability that a sacrificial node is close to the attacker as specified by eq.(3).

Indicators such as the Link Quality Indicator (LQI) or Received Signal Strength Indicator (RSSI) can be used as relative measurements about the reliability of the links between the attacker and the sensor node, and between the sensor node and the DRA. Other indicators such as success rate, which is defined as the ratio of number of packets sent to number of packets received can also be used. The DRA can calculate the reliability from the sensor node to itself

using these indicators. Also the LQI and RSSI can be used to produce an estimate of the distance between the attacker and the DRA, and data from multiple sensor nodes can be used to determine the approximate location of the attacking node relative to the DRA, if there is uncertainty about the location of the attacker.

Because LQI and/or RSSI are not the only factors to determine if a node can be a candidate node, there are additional estimators that affect the efficiency. The probability of the existence of a sensor node in the overlap region also plays a critical role in the probabilistic approach. One method is to model the supposed probability of the existence of a sensor node. We first measure the probability that nodes between the DRAs and attackers as two-hop neighbors, that is, they are not within direct communications range:

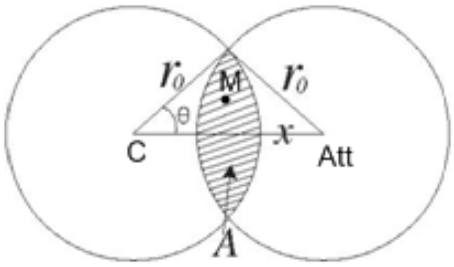

Figure 6. An illustration of two-hop between the DRA and attacker
C is a DRA. Att is an attacker. M is a sacrificial node. A is an intersection area.

From [31] the probability that there is a candidate retaliation node in area A between the DRA and attacker is

$$\Phi(x) = 1 - e^{-\rho A} \tag{1}$$

Where

$$A = 2{r_0}^2 arcsin\left(\sqrt{1-\frac{x^2}{4{r_0}^2}}\right) - xr_0\sqrt{1-\frac{x^2}{4{r_0}^2}} \quad (2)$$

Where r_0 is the communication range of the attacker and DRA, x is the distance between the DRA and attacker. Figure7 shows the intersection area will increase when the x value decreases. It means when x is between r and $2r$, as the distance gets shorter, the intersection area will get bigger. Then Φ(x) also gets bigger. x cannot be smaller than r since the DRA should not directly communicate with the attackers. In other words, the probability that there is a candidate retaliation node in the area A will get bigger if the DRA is close to the attacker but not in the attacking area. Also by elementary algebraic calculation, $dA/dx > 0$ for $x \in [r, 2r]$, it means A is a monotonically decreasing function.

If x is equal to or close to r, the probability that a sacrificial node can be close to the attacker will be large. If x is equal or close to $2r$, the probability of that will be small.

Suppose α_i is the probability that a sacrificial node is close to the attacker. Then

$$\alpha_i = \Phi(x) = 1 - e^{-\rho A} \quad (3)$$

Suppose P_i is the probability that a node can be a good sacrificial node which can get the command from the DRA and have good communication with the attacker. P_i is based on one or more of the factors mentioned earlier in this chapter, that is, LQI, RSSI and the probability of a node existing in the right region.

Definition 3: Monitoring activity $\mu(q, t)$: is the probability that q (the attacker) is within the monitoring range of at least one sacrificial node. We assume that the monitoring activity is invariant with respect to time, that is, $\mu(q)$.

The total monitoring activity of q is given by,

$$\boldsymbol{\mu(q) = 1 - \prod_{\forall i \in S(q)} (1 - P_i)} \quad (4)$$

Where $S(q)$ denotes the set of all sacrificial nodes.

From this equation, it shows the higher probability of the sacrificial model running well, the bigger the $\mu(q)$ value. However if only sensor nodes with a higher P_i are selected, it will be easy for an attacker to detect fraud because it may be possible that the distribution of the nodes is uneven. So using the ratio (see below) is the best way to select the sacrificial nodes based on density of candidate nodes.

5.1.2 Response

Depending on the retaliation algorithm, the DRAs can calculate the number of sacrificial nodes needed (section 5.3). A fixed ratio of sacrificial nodes to candidate retaliation nodes is used to select the sacrificial nodes. In addition to calculating the number of sacrificial nodes needed, each DRA also calculates the number of candidate retaliation nodes. The overlap density based selection method sets an invariable ratio that makes the percentage of the sacrificial nodes to candidate retaliation nodes unchanged for each DRA and attacker. The ratio can be decided in many ways. If the density of sensor nodes is higher, then the ratio can be lower. If the other part of the network is collecting important information, increasing the ratio will increase the probability of success of the retaliation model.

The model sets an initial threshold for the total monitoring activity. It also relates with the original purpose of the sensor network. If the sensor network is working on sensitive data or the attacker has enough intelligence, then the threshold should be set higher.

After the retaliation process starts, the effective working area of a sacrificial node will shrink over time because of declining power levels in the node's power source. This will cause P_i to decline over time. If a sacrificial node's P_i falls below a certain value, that node cannot be

counted on as a sacrificial node anymore. It will not have the ability to finish the retaliation process. The node will be placed into a dormant state at the command of the DRA. When the probabilistic approach falls below the initial threshold, then reconfiguration takes place. The Model will be used to add more sacrificial nodes.

The total monitoring activity is based on the monitoring activities of the sacrificial nodes within each DRA region. Each DRA region will provide a monitoring of:

$$\boldsymbol{\mu(q) = 1 - \prod_{\forall i \in S(q)} (1 - P_i)} \\ \boldsymbol{= 1 - (\prod_{\forall i \in C_1(q)} (1 - P_1) \times \prod_{\forall i \in C_2(q)} (1 - P_2) \times \ldots)} \quad (5)$$

where $C_i(q)$ is the set of sacrificial nodes in a DRA *i.* If a sacrificial node within a DRA 'dies,' then the monitoring activity of the cluster will be reduced. However, the total monitoring as indicated by eq.(5) may still be above the threshold and in such a case there is no need to select a new sacrificial node. If the total monitoring is below the threshold, a node selection process takes place which is described below.

5.1.3 Selecting Sacrificial nodes

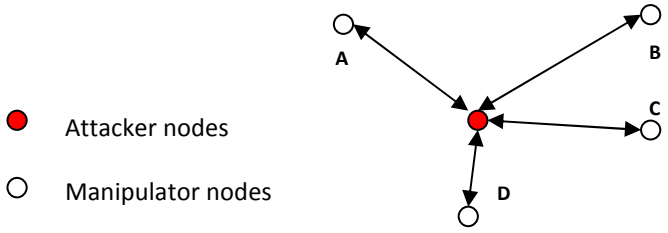

Figure 7. Manipulation Network model

The algorithm for selecting sacrificial nodes is outlined. An overlap density based selection method is used to maximize coverage.

Algorithm for retaliation model:

Step 1: The DRAs use context awareness or intrusion detection to determine if there is a suspicion of an attack.

Step 2: If there is a suspicion of an attack, then it will:

i. Determine which DRAs should respond to defeat the attack – If the DRA within communication range of an attacker receives a message from their own nodes and from the attacker with the same source and timestamp, then the DRA will not send the command to retaliate. Otherwise it would select the sacrificial nodes. Hence the attacker cannot overhear the instructions by a DRA.

ii. Determine nodes m_1, m_2, …m_n (candidate retaliation nodes) within the attacker's communication range.

iii. Use the overlap density based selection method to distribute the nodes to maximize the coverage while retaliation is taking place.

iv. Send the command to the nodes selected in the last step.

Step 3: Sacrificial nodes n_1, n_2, …n_n apply retaliation algorithm for manipulating the attacker.

Step 4: At regular intervals, the total monitoring activity μ (q) of the attacker point q is checked by each DRA. If this falls below the threshold, the DRA which has the lowest ratio should add a new sacrificial node to do the retaliation. If several DRAs have the same ratio at the same point, the DRA which has the smallest ID will be selected to add a new node. The frequency of checking is determined by the initial monitoring activity level.

Step 5: Retaliate until objectives are met or retaliation is not successful.

5.2 Response Model

To the defender the objective of retaliation is two-fold:

- To establish a secure connection between the defender D1 and the destination

- To deplete attacker resources

The objective of the attacker is to launch a wormhole attack by establishing a seemingly natural route with the defender and monitor the incoming and outgoing packets to the destination. We make an assumption that the attacker cannot modify the underlying AODV routing protocol.

An attack may not be detected immediately. Therefore, some nodes may start using the wormhole path (type A). It is only after the wormhole is detected, the retaliation kicks into action. Hence there are two types of nodes involved in the retaliation:

- Nodes that are communicating normally with the attacker – type A
- Nodes that send RREQs – these are nodes that were not communicating, but send RREQ after the attack is detected – type B

Type A nodes - nodes *a,…,m* are nodes that are sending normal data packets.

Type B nodes – nodes *n,…,p* are sending RREQ packets on a wormhole attack being detected

As the attacker is using a private high speed wormhole link, the RREP from the destination node will reach the defender first through the attacker route. Thus the objective of Type B nodes is to make sure that the RREP packets from the destination node does not reach the defender first through the attacker route. This means that the Type B nodes have to find an alternate path for the RREP from the base, other than that through the attacker route. To achieve this, given a RREP, we need to find the probability that this RREP is for the RREQ of D1, with the assumption that the attacker is able to handle all the packets.

Probability of response to D1 given attacker is able to handle all packets

From [32], the maximum per node throughput in a wireless network featuring many-to-one communication is upper bounded by W/n bits per sec where W is the transmission capacity of the channel and n sources. In our work we assume a worst-case scenario, that is, the upper bound.

Let all the packets that go through the attacker be *PA*. *PA* therefore includes all the *RREQs*, *RREP*s and all the data packets.

Using Bayes theorem: $P(D1|PA) = P(D1)\frac{P(PA|D1)}{P(PA)}$ (6)

where

$$P(PA) = P(a)P(PA|a) + \cdots + P(m)P(PA|m) + P(n)P(PA|n) + \cdots + P(p)P(PA|p) \quad (7)$$

Since D1 sends one *RREQ* and receives one *RREP*, $P(D1_{RREP}|PA)$ is the probability that *D1* receives a *RREP* from all the packets *PA,* which is therefore from (6):

$$P(D1_{RREP}|PA) = P(D1_{RREP})\frac{P(PA|D1_{RREP})}{P(PA)} \quad (8)$$

Assume each of nodes *a,..,m* send *x* packets and there are #(*a,..,m*) nodes. Therefore the total number of packets traveling through the attacker by nodes *a,..,m* is $x \times$ #(*a,..,m*). We assume a worst case scenario that there are no acknowledgement ACK packets.

Nodes *n,..,p* send one RREQ each and receive one RREP. Total number of packets traveling through the attacker is $2 \times$ #(*n,..,p*) of which #(*n,..,p*) are RREP packets

The constraint based on [1] is:

- $x \times$ #(*a,..,m*) $+ 2 \times$ #(*n,..,p*) = *W*

Therefore:

- each node $i \in \{ a,..,m\}$ sends $\frac{x}{x\times\#(a,\ldots,m)+2\times\#(n,\ldots,p)} \times 100\%$ of packets (9)
- each node $j \in \{ n,..,p\}$ sends or receives $\frac{2}{x\times\#(a,\ldots,m)+2\times\#(n,\ldots,p)} \times 100\%$ of packets of which $\frac{1}{x\times\#(a,\ldots,m)+2\times\#(n,\ldots,p)} \times 100\%$ are RREP packets (10)

Hence we can obtain $P(PA)$ and $P(PA|D1_{RREP})$ and calculate $P(D1_{RREP}|PA)$

From (7) $P(PA) = 1$ as all packets are considered. Similarly $P(PA|D1_{RREP}) = 1$

From (10) $P(D1_{RREP}) = \frac{2}{x \times \#(a,\ldots,m) + 2 \times \#(n,\ldots,p)} \times 100$

$P(D1_{RREP}|PA)$ decreases as the capacity of the attacker is exceeded, that is,

$$x \times \#(a,..,m) + 2 \times \#(n,..,p) \geq W$$

Since $x \times \#(a,..,m) + 2 \times \#(n,..,p) \geq W$, that is, $x \times \#(a,..,m) + 2 \times \#(n,..,p) = mW$, where m >1, then, $P(D1_{RREP}|PA)$ reduces by m, that is, $\frac{P(D1_{RREP}|PA)}{m}$

$$\text{Probability measure} = \begin{cases} \frac{P(D1_{RREP}|PA)}{1} & \text{if attacker is not overloaded} \\ \frac{P(D1_{RREP}|PA)}{m} & m > 1 \text{ if attacker is overloaded} \end{cases} \quad (11)$$

The probability measure gives an indication of the chances of the attacker sending a RREP for D1.

The selection of the sensor nodes that actually does the retaliation process (sacrificial nodes or Type B nodes) depends on the capacity of the attacker. We assume that the capacity of the attacker is higher than that of the sensor nodes. But only the attacker is aware of his capacity. We estimate the bandwidth/capacity of the attacker, based on which we adjust the number of sacrificial nodes. An overlap density based selection method will be used to distribute the sacrificial nodes to maximize the coverage while retaliation is taking place. The sacrificial nodes must be selected from the overlap region (intersection area) between the clusters surrounding the cluster in which the attacker belongs, and thus within the communication range of the attacker.

As the selected sacrificial nodes keep the attacker busy by sending RREQ packets, its power level also reduces. Thus the probability that the node is an effective sacrificial node decreases over time and hence the DRA should elect some more sacrificial nodes to complete the

retaliation process. This selection is a prediction/learning process meaning that initially some number of sacrificial nodes may be selected by the DRA to send RREQ packets to the attacker. If the attacker is able to process all the RREQs and send RREPs back within the time of flight measure, then the DRA adds more number of sacrificial nodes.

It may take time to detect the attack. Therefore some nodes may already be communicating with the attacker. Others send RREQ packets when an attack is detected. The load at the attacker must be increased if the attacker is able to process all the RREQs that go through the attacker's route, which is shown graphically in Fig. 8.

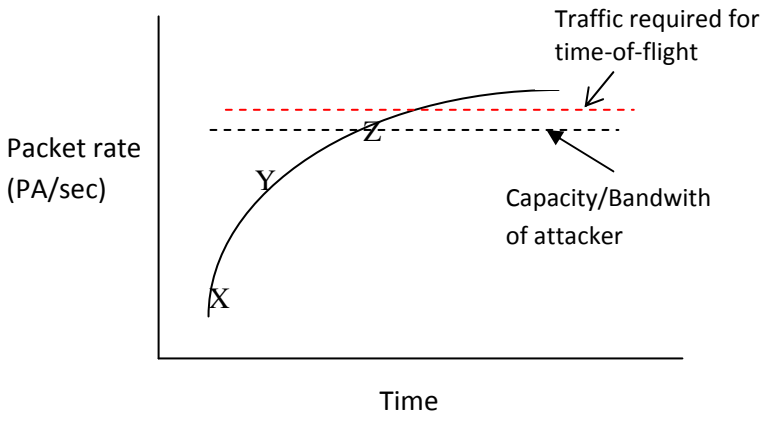

Figure 8. Packet rate through attacker

The first step in achieving our objective of overloading the attacker is the congestion estimator. In order to know if the attacker is at his full capacity, we need to monitor/estimate the congestion at the attacker's area of the network. The congestion may be due to network bandwidth being congested or the capacity of the attacker, which is measured in bandwidth.

After estimating the capacity/bandwidth of the attacker, the next step is to control the number of RREQ packets, as shown in Fig. 9, which needs to be sent out by the Type B nodes, in order to keep the attacker busy by wasting his resources.

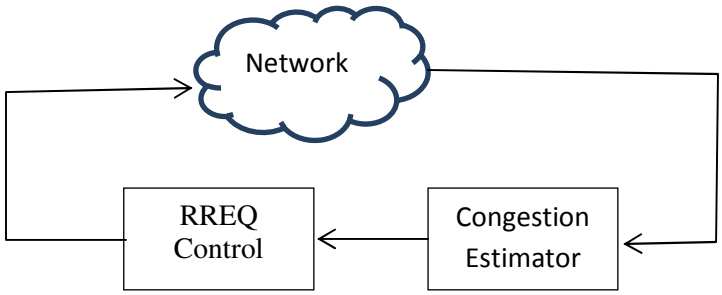

Figure 9. RREQ Control

From [33] the bandwidth (or congestion) is estimated as:

$$B[k] = \left(1 - e^{\frac{-T|k|}{K}}\right) * \frac{L[k]}{T[k]} + e^{\frac{-T|k|}{K}} * B[k-1] \qquad (12)$$

B is the estimated bandwidth, $L[k]$ is the number of RREP received, $T[k]$ is the last RREP inter-arrival, $L[k]/T[k]$ is the rate of the RREP stream, and K is a time constant. k and k - 1 represent the actual and the previous values of the variables.

Referring to Fig. 8, if the bandwidth is very high, then we are at point X. If the bandwidth is low, then we are at point Y. At point Z, we are at bandwidth of zero, which means that the attacker is at his full capacity. Below bandwidth of 0, the attacker is overloaded and thus $P(D1_{RREP}|PA)$ decreases. Hence we need to calculate the value of m in equation (6) to obtain the probability measure. As long as the bandwidth is above 0, our assumption is that all the RREQs will be received. However, the probability will be low if many RREQ packets are being sent by the sacrificial nodes.

The bandwidth estimator is used to increase the number of RREQs or sacrificial nodes. For example, if the bandwidth estimator says 100bits/sec, then increasing the RREQs by 50bits/sec (and 50bit/sec for RREPs), will saturate the bandwidth. As the number of sacrificial nodes is increased, the new bandwidth is calculated to determine if more sacrificial nodes are needed. The probability is also calculated. It is important to note that if the bandwidth is high,

then all the RREPs will be received. The time window is the time of flight. At the end of this time the bandwidth and probabilities are calculated.

If the bandwidth is high, then the probability of an RREP through the attacker being intended for the RREQ from D1 is defined by the first part of equation (11) where the attacker is not overloaded. For example, if the bandwidth is 100bits/sec and time of flight is 2 sec, then the sacrificial nodes must transmit 200bits or more. If the bandwidth is below zero, shown as red line in Figure9, then the probability of any packet is determined by the second part of equation (11) where the attacker is overloaded.

Algorithm for the Response Model

Step 1: The defender node D1 notifies all the selected sacrificial nodes to establish a connection with the destination node D through the attacker route.

Step 2a: Learn the attacker pattern. The attacker pattern depends on the number of route replies RREPs that come back to the defender node D1 through the attacker route.

Step 2b: The sacrificial nodes (n... p) send RREQ packets to the destination node D through the attacker route. The faster the replies come back, the more the number of route requests that have to be sent out by the sacrificial nodes. The probabilistic measure will predict the number of RREP packets from the attacker route.

i. Estimate the capacity/bandwidth of the attacker (Step 3)
ii. If the probability that the RREP reaches the defender before the time of flight measure is high, the attacker is not overloaded. Thus sacrificial nodes send more RREQ through the attacker route, to keep him busy.
iii. Else, the attacker is overloaded; that is the probability that the RREP will reach the defender first through the attacker route decreases by a constant factor m.

The objective is for the desired path to be found with a high probability, that is, the RREP from the attacker is unlikely to be sent to the defender before the time of flight measure. The probability measure gives an indication of the chances of the attacker sending a RREP for D1.

$$\text{Probability measure} = \begin{cases} \frac{P(D1_{RREP}|PA)}{1} & \text{if attacker is not overloaded} \\ \frac{P(D1_{RREP}|PA)}{\text{m}} & \text{m} > 1 \text{ if attacker is overloaded} \end{cases}$$

Step 3: Estimate the capacity or bandwidth of the attacker

$$B[k] = \left(1 - e^{\frac{-T|k|}{K}}\right) * \frac{L[k]}{T[k]} + e^{\frac{-T|k|}{K}} * B[k-1]$$

i. Time Window is set to the time of flight measure. Calculate bandwidth and probabilities in step 2 at the end of this time.

ii. If bandwidth is high (>0), then the attacker is not overloaded; thus number of RREQs or sacrificial nodes is increased to saturate the bandwidth.

iii. Else, if bandwidth is below zero, attacker is overloaded.

Step 4a: If an RREP is received by a sacrificial node (n… p), it sends a data packet to the destination node D through the attacker route. Send at rate that keeps attacker busy.

Step 4b: If an RREP is received by the defender node D1, then the retaliation is only partially successful. The number of sacrificial nodes is increased as in step 3 and another RREQ is sent.

Step 5: Defender node D1 sends an actual data packet to destination once the RREP for a route that is different from the attacker route is received – retaliation is successful

CHAPTER VI

SIMULATIONS

6.1 Sensor Network Simulator and Emulator (SENSE)

To validate our approach, we simulated a wormhole attack and responded to make the attacker busy by sending RREQs from the sacrificial nodes, using the simulation tool SENSE (Sensor Network Simulator and Emulator) [36]. SENSE was chosen for numerous reasons. It is a user-friendly simulator implemented in C++, which has a good compiler support and the execution speed is generally faster. With the Standard Template Library, C++ programs can easily achieve high efficiency while maintaining a high level of code reuse. SENSE is an efficient and powerful sensor network simulator that is also easy to use. Unlike object-oriented network simulators, SENSE is based on a novel component-oriented simulation methodology that promotes extensibility and reusability to the maximum degree. At the same time, the simulation efficiency and the issue of scalability are not overlooked.

SENSE runs on top of COST, a component based discrete event simulator that is written in CompC++, a component extension to C++. The component-port model gives the users a great deal of freedom in configuring sensor nodes. It makes simulation models extensible i.e., a new component can replace an old one or if they have compatible interfaces and inheritance is not required. This also promotes reusability: a component developed for one simulation can be used in another if it satisfies the latter's requirements on the interface and semantics. Components interact with each other only via input and output ports, and thus development of a component

becomes completely independent of others. An inport implements certain functionality, so it is similar to a function. In contrast, an outport defines what functionality it expects out of others (abstraction of a function pointer).

The currently available components and simulation engine in SENSE are given below:

1. Battery Model: Linear Battery, Discharge Rate Dependent and/or Relaxation Battery
2. Application Layer : Random Neighbor; Constant Bit Rate
3. Network Layer: Simple Flooding; a simplified version of ADOV without route repairing, a simplified version of DSR without route repairing
4. MAC Layer: NullMAC; IEEE 802.11 with DCF

Fig. 10 shows the internals of a typical sensor node [36]. The sensor node is a composite component. It consists of a number of smaller primitive components, each implementing certain functionality. Normally a sensor node has some layered network protocol components, a power component and a battery component both of which are related to power management and others such as mobility and sensor. The inports and outports of the sensor node component are directly connected to the corresponding inports and outports of internal components.

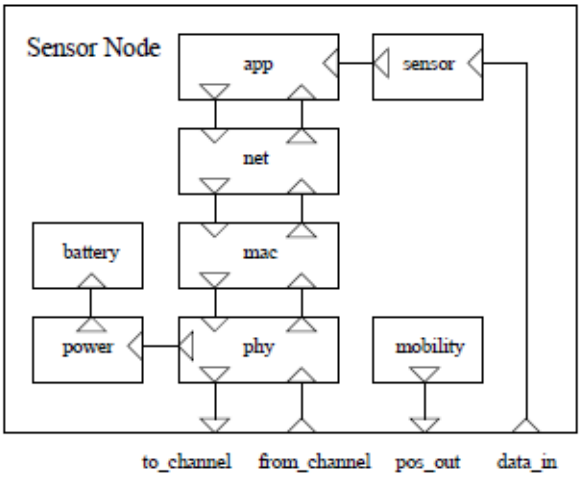

Figure 10. The internal structure of a typical sensor node

6.2 System Specifications

The proposed solution was implemented on Toshiba Laptop with the following specifications:

- Processor: Intel Pentium Dual CPU T3400 2.16GHz
- Installed Memory (RAM): 4.00GB
- System Type: 64-bit Operating System (Windows 7 Professional)

To run SENSE, the SSH (Secure Shell) client PUTTY was used to connect to the CentOS 5.6 Linux server of the Department of Computer Science at Oklahoma State University.

The following changes were made in the configuration to enable the SENSE simulator to execute on the Linux server:

- The basicDefinitions.mk file was modified to update the compile command to link the visualizer library:

  G++ -Wall –o sim_aodv sim_aodv.cxx ../../libraries/visualizer/lib/libvisualizer.a

- The compiler flag in definitions.mk file was modified:

  CCFLAGS += -DVISUAL_ROUTE -DVR_SIZE=30 –fpermissive

- The path for the linker libraries was updated:

  LD_SHARED = #-Wl, --rpath –Wl,/usr/lib

- The development package of Bison (thus bison itself) was installed as the parser generator, with the help of the system administrator.

6.3 Implementation Details

The proposed solution was implemented and validated by running numerous simulations in SENSE. The implementation of the network setup (described in section 5.2), candidate node selection (described in section 5.2.1), and sacrificial node selection (described in section 5.2.3) are explained in section 6.3.1. The response model of choosing the sacrificial nodes depending on

the capacity of the attacker (described in section 5.3) is presented in section 6.3.2. The estimation of the attacker's capacity is presented in section 6.3.3. Finally, the process of finding the monitoring activity of response model (described in section 5.2.2) is presented in section 6.3.4.

6.3.1 Network Setup

The network is constructed with approximately 300 sensor nodes. These sensor nodes are identical in terms of its resources like power, battery levels etc. The mesh topology was chosen to place the sensor nodes at a distance of 100 units. The source and destination were randomly selected. Assuming only one attacker node is present in the wormhole, it is positioned such that the route from the source to destination is the shortest path. The power levels of the attacker and the rest of the sensor nodes were selected, after various possible tries, such that the path from the source to the destination passes through the attacker, creating a wormhole attack. While doing so, the communication range of the attacker goes up when compared to that of the rest of the sensor nodes. The chosen power level of the attacker was 0.003 dBm, while that of the rest of the sensor nodes was 0.0003 dBm. For this selected power levels, the communication range of the attacker was 6 times higher than that of the normal sensor nodes.

The positions of the Distributed Retaliation Agents (DRAs) were selected in such a way that their communication ranges overlap with that of the attacker node, in order to select the sacrificial nodes from the intersection area. For this work four such DRAs were selected, whose communication range was the same as that of the attacker. With these power levels and the communication ranges, the total number of candidate nodes in all four of the DRAs was found to be 72, which are capable of responding against the wormhole attack. The sacrificial nodes which actually respond to the wormhole attack will be selected from this set of 72 candidate nodes, depending on the power level of the attacker.

6.3.2 Response Model Simulation

As the wormhole attack may not be detected immediately, some of the sensor nodes may start communicating with the attacker through the wormhole link (Type A). We assume that there is one such sensor node that starts communicating with the attacker. The next step is to find the required number of sacrificial nodes (Type B) that send RREQs to keep the attacker busy, in order to find an alternate path. The ideal situation is to keep increasing the number of sacrificial nodes dynamically at runtime, until an alternate path is found. However, in SENSE for a source node to communicate with a destination node, a connection needed to be established between them before runtime. Hence it was not possible to determine the required number of sacrificial nodes dynamically at runtime. Thus to overcome this, for a given power level of the attacker, sacrificial nodes were added manually until an alternate path was found. Thus each addition of a sacrificial node was a different simulation, until the attacker was not present in the route from source to destination node.

For example, say for the given power level of the attacker, it requires `n' different simulations to get an alternate path. Simulation 1 will have one Type A node to communicate with the attacker with no sacrificial nodes at all. Simulation 2 will have two nodes: one Type A node and one sacrificial node. Simulation 3 will have three nodes: one Type A node and two sacrificial nodes, and so on. Thus each simulation will have one Type A node and an additional sacrificial node compared to the previous simulation.

The above mentioned approach was repeated for five different power levels of the attacker: 0.003, 0.0035, 0.004, 0.0045 and 0.005 (dBm). Each of these was a separate simulation with different number of sacrificial nodes sending RREQs to the attacker. The number of such sacrificial nodes required to keep the attacker busy depends on the power level of the attacker. An attacker with a low power level requires fewer sacrificial nodes compared to an attacker whose power level is high. For example, for a power of 0.003 dbm, 10 sacrificial nodes are required to

get an alternate path from source to destination, and it requires 30 sacrificial nodes for a power level of 0.005 dBm. From this data it was clear that as the power level of the attacker increased, the number of sacrificial nodes required to keep the attacker busy also increased. That is, the number of sacrificial nodes is proportional to the power level of the attacker node, as shown in the graph below (Fig. 11).

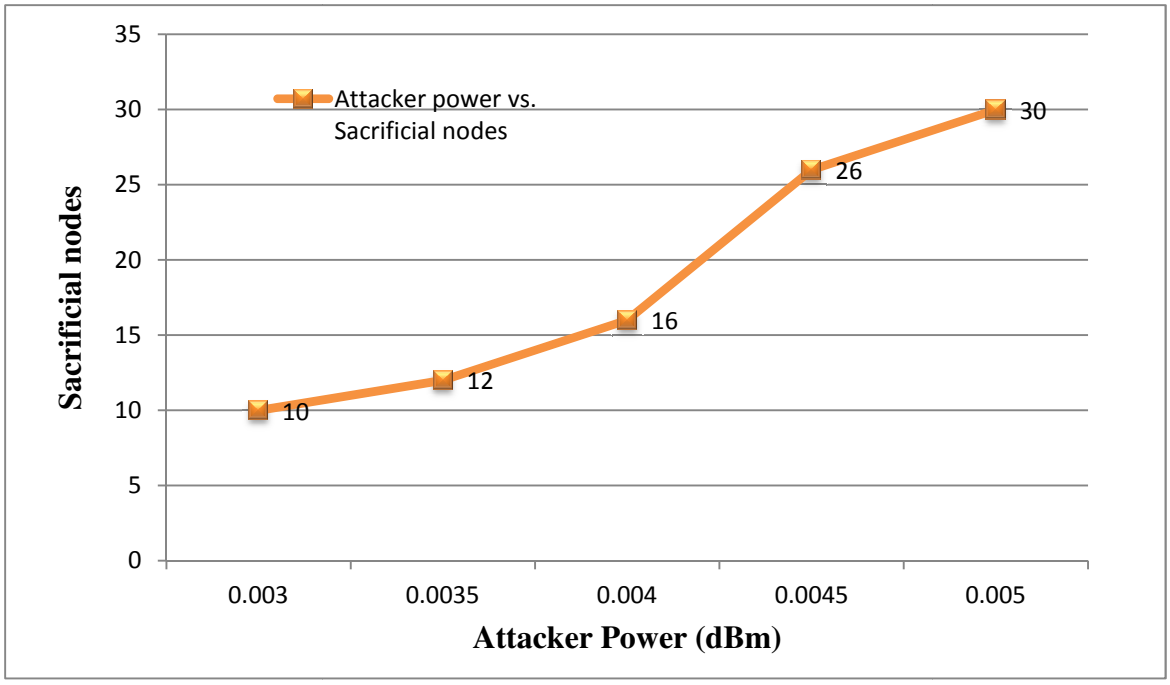

Figure 11. Results of Response Model Simulation

6.3.3 Bandwidth/Congestion Estimation

The wormhole attacker has a high speed network with which he is able to gain access to the information flowing through the network. The exact capacity/bandwidth of the attacker is unknown. But the utilization of the attacker can be found, which depends on the number of Route Replies (RREPs) that go through the attacker, as in eq.12. The objective of the sacrificial nodes is to keep the attacker busy by sending Route Requests (RREQs) in order to waste his resources. The utilization measure determines how busy the attacker is at any given point in time, that is, it measures the congestion at the attacker's area of the network. The more the utilization of the

attacker, the busier or more congested the attacker is. This means that as the utilization of the attacker increases, the attacker won't be able to process the route requests quickly, and thus an alternate path from source to destination can be found.

For the highest power level of the attacker (0.005 dBm), the time taken to find an alternate path from source to destination was 178.589 seconds. The time period in which all the RREPs go through the attacker, within 178.589 seconds, was found to be (70.0, 70.3). This time frame was divided into 3 intervals, each with a time constant (K) of 0.1 seconds, which is the range of the intervals. For each interval, the number of RREPs received by the attacker, the last RREP inter-arrival, the rate of RREP stream was calculated. At the beginning, the attacker is at his full capacity as his resources are not yet used up. Thus the initial value of his utilization is taken to be zero. With these values, the utilization of the attacker was determined at the end of each interval as given in Table 2 below.

As the sacrificial nodes keep sending RREQ packets to the attacker, the attacker starts processing those requests, until his bandwidth/capacity is fully overloaded. Thus the utilization of the attacker keeps going up, so that at one point he won't be able to process all the RREQs from the sacrificial nodes, which is when an alternate path from source to destination is found. The results at the end of each interval obtained from eq.12 are shown in Table 1 and Fig. 12.

Interval	1	2	3
# of RREPs (L[k])	4	25	5
Last RREP inter-arrival (T[k])	0.00205	0.00108	0.00447
Rate of RREP stream (L[k]/T[k])	1954.14	23203.7	1117.85
Time Constant (K)	0.1	0.1	0.1
Previous utilization value (B[k-1])	0	39.5934	287.827
Actual utilization value (B[k])	39.5934	287.827	324.135

Table 1. Utilization/Congestion Estimation

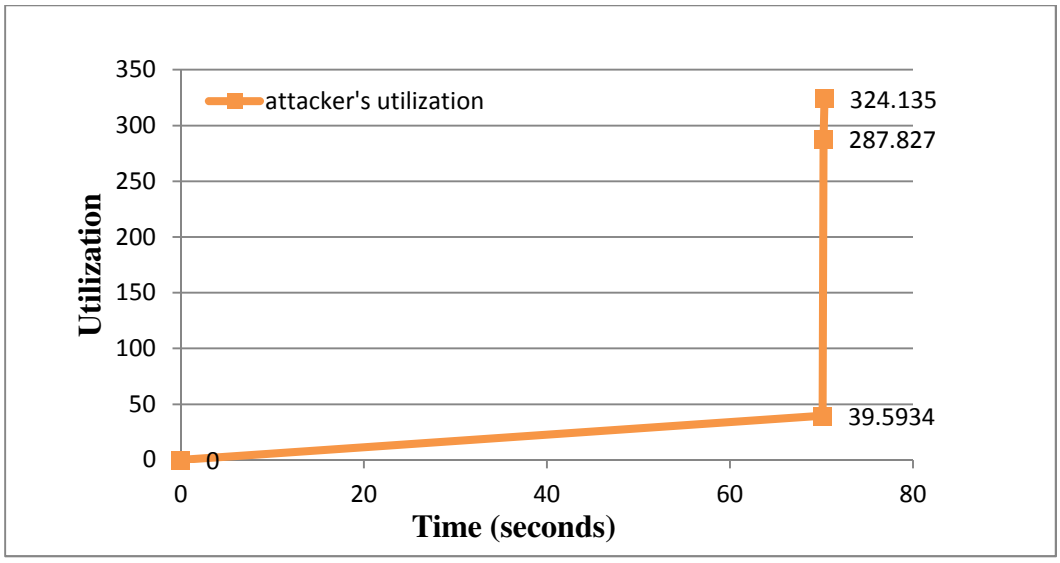

Figure 12. Utilization/Congestion Estimation

The aim of the proposed solution is to show that if an attacker gets busier by processing the RREQs from the sacrificial nodes, the attacker will not be able to process the RREQ from the source and hence an alternate path from the source to the destination is found after the attacker gets overloaded. To prove this, the following data were collected from the simulation for the highest power level of the attacker (0.005 dBm), and represented graphically below:

1. The number of packets of all types sent by the attacker (Fig. 13)
2. The number of packets of all types received by the attacker (Fig. 14)
3. The total number of packets sent and received by the attacker (Fig. 15)
4. The time when the attacker is overloaded and when an alternate path is found (Fig. 16)

The measure of the attacker being overloaded depends on the number of packets sent and received by the attacker. Initially when the attacker is at his full capacity, he will be able to send out a packet for every packet that he is receiving from other nodes. As the sacrificial nodes are sending out Route Requests (RREQs), the attacker gets busier. This is mainly due to the number of duplicate RREQs that go through the attacker, as his transmission power is higher than the rest

of the nodes. This can be proved by comparing the graphs in Figure 13 and 14. Thus the attacker sends out fewer packets compared to the number of packets he receives, at which time the attacker is overloaded or busy enough that he is not able to process the RREQ from the source. Due to this, the RREP from the destination reaches the source through a path that does not go through the attacker, at which time an alternate path is found.

For the given power level of 0.005 dBm, as shown in Figure 18, the time when the attacker gets overloaded is at 4.9125 seconds. At this time, the attacker sends out only two packets, but he receives 57 packets in total. The time when the alternate path is found is shown to be 178.543 seconds. The data for two other power levels of the attacker 0.003 dBm and 0.0035 dBm are given in Appendix A.

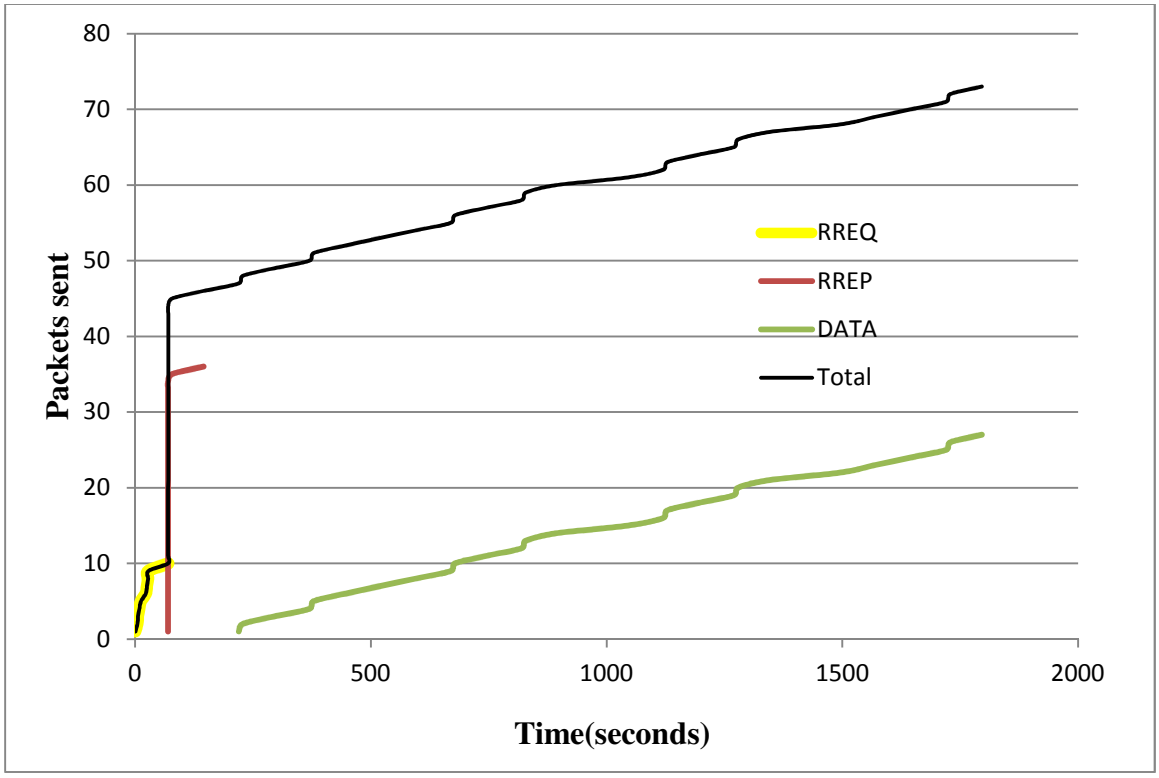

Figure 13. Packets sent by the attacker at power level 0.005

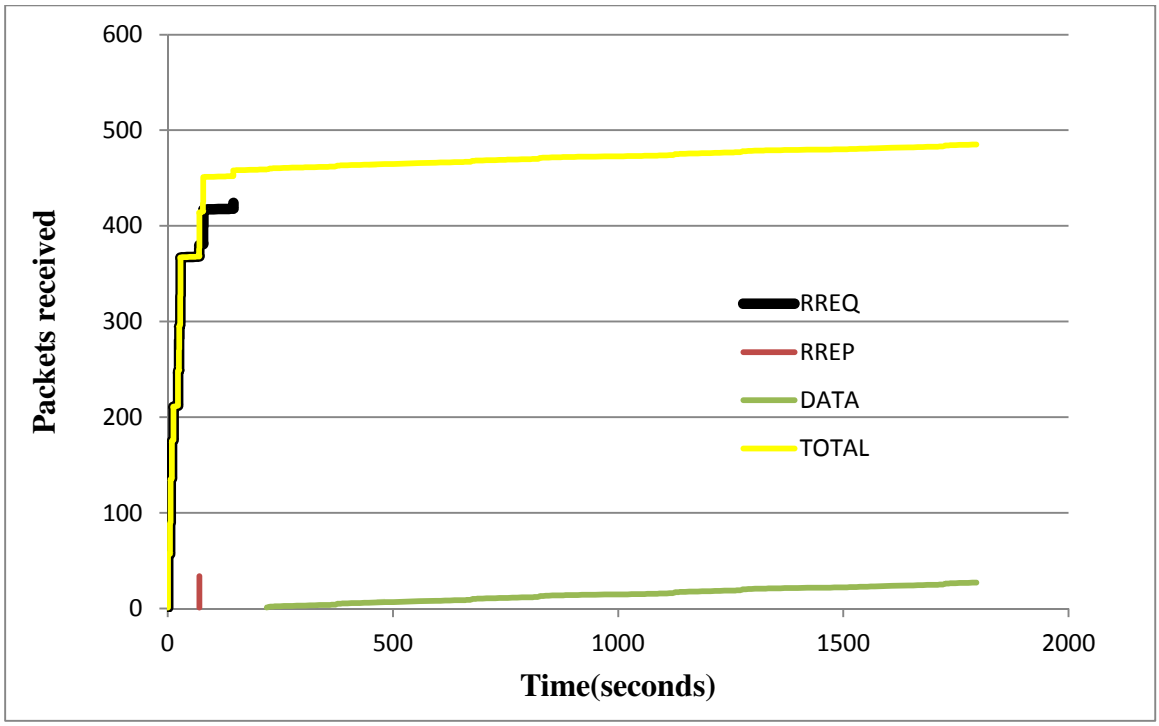

Figure 14. Packets received by the attacker at power level 0.005

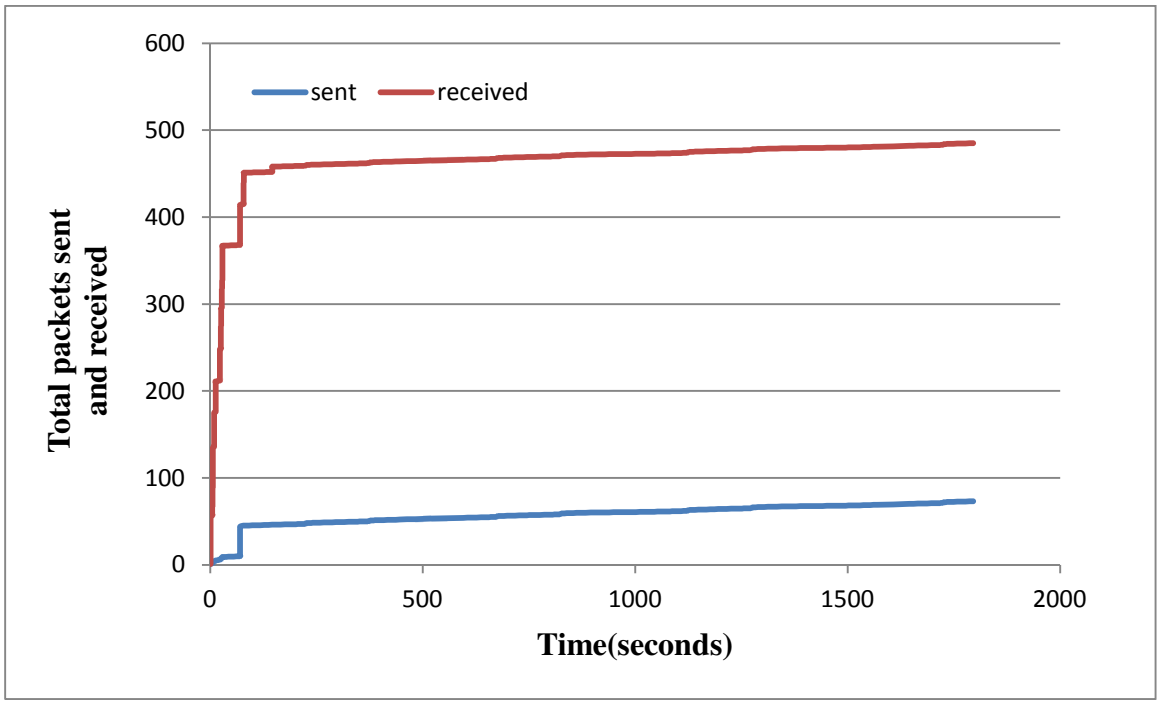

Figure 15. Packets sent and received by the attacker at power level 0.005

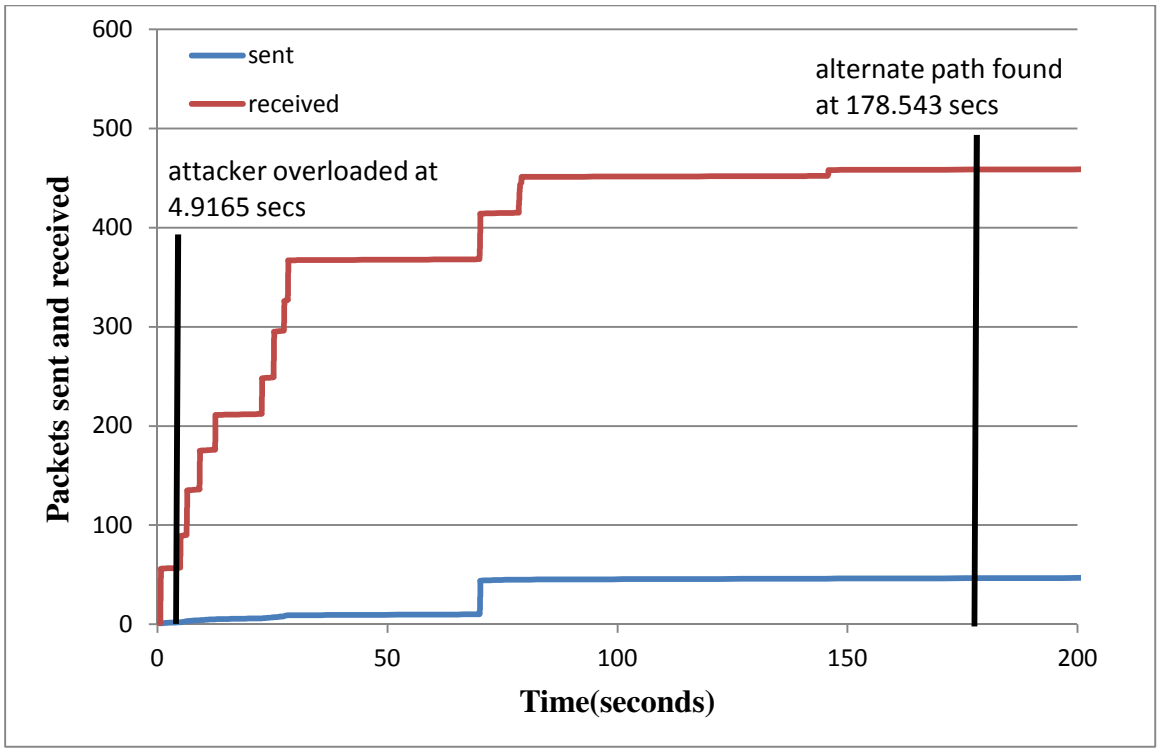

Figure 16. Time when attacker is overloaded and alternate path found at power 0.005

6.3.4 Monitoring Activity of Response Model

In section 6.3.3 we carried out our experiments for different power levels of the attacker node, with all the rest of the nodes, including the sacrificial nodes having the same power levels. Now we experiment with different power levels of the sacrificial nodes for one power level of the attacker, with the rest of the nodes having the same power level as in previous section (0.0003 dBm). As the highest power level of the attacker (0.005 dBm) required 30 sacrificial nodes to keep the attacker away from the route between source and destination nodes, we took that simulation as a base for the simulations for finding the monitoring activity of the response model. Four different simulations with different power levels for the sacrificial nodes were carried out: 0.0001 dBm, 0.0002 dBm, 0.0006 dBm and random power levels for the sacrificial nodes in the range (0.0001, 0.0006) dBm.

From the results of the simulation in Phase I, for the highest attacker power level of 0.005 dBm, the time taken to find an alternate path from source to the destination node was calculated as 178.589 seconds. The success rate of each of the sacrificial nodes used in the simulation was found out. The success rate gives us the ratio between the packets sent by the sacrificial nodes to the packets received by them.

We ran the above mentioned simulations with different power levels of the sacrificial nodes, to find the monitoring activity of the attacker, i.e., to check if the attacker is within the monitoring range of at least one sacrificial node. As described in chapter 5, the monitoring activity of the attacker is directly proportional to the probability of the sacrificial node having sufficient resources and being within range. In our case, the probability value was measured as the success rate of all the sacrificial nodes used in Phase I. The average of the success rates was calculated as 3.08391, which was taken as the initial threshold value for the total monitoring activity.

As the sacrificial nodes continue to respond against the attacker by sending RREQs, their effective working area will shrink over time. Thus at some point the success rate of some of the sacrificial nodes will fall below the set threshold value. Hence we replace those sacrificial nodes with other sacrificial nodes, in order to effectively respond against the attack. We kept repeating this, until an alternate path from source to destination node was found. For the different power levels of the sacrificial nodes, the number of changes it takes to get an alternate path, the number of sacrificial nodes to be changed each time and the total number of nodes changed is shown in Table 2. The graphical representation showing the number of nodes changed each time for the four power levels of the sacrificial nodes is shown in Fig. 17 and a bar chart showing the total number of nodes changed for each power level of the sacrificial nodes is shown in Fig. 18 below.

Table 2. Phase II Results

Power (dBm) / # of changes	0.0001	0.0002	0.0006	0.0001–0.0006
1	15	10	10	8
2	24	27	18	20
3	24	29	2	2
4	29	23		
5	26			
6	3			
Total # of nodes changed	121	89	30	30

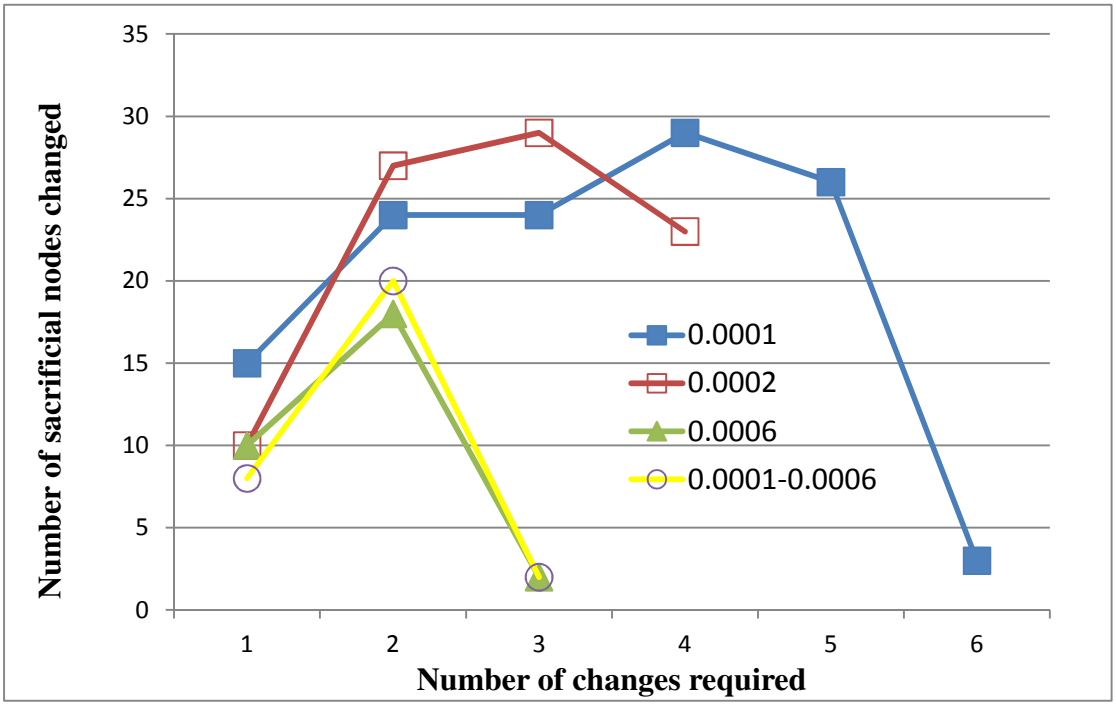

Figure 17. Number of changes vs. number of nodes changed each time

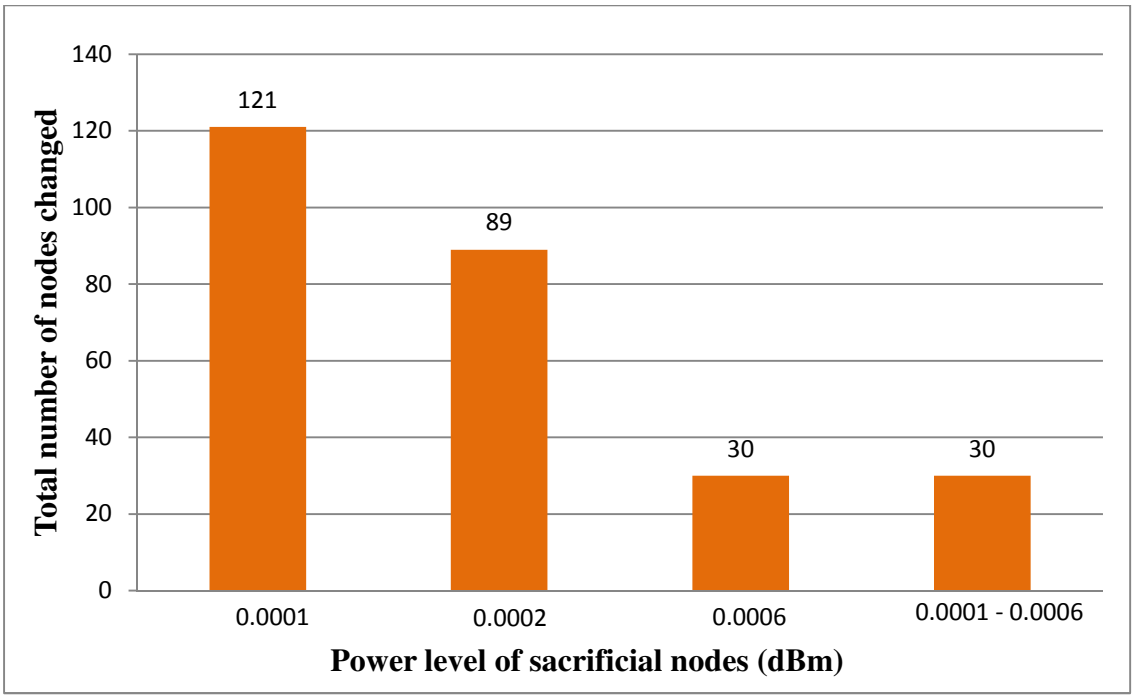

Figure 18. Total number of nodes changed for the given power of sacrificial nodes

It is clear from the above results that as the power of the sacrificial nodes increases, the number of times we need to change the sacrificial nodes decreases. For each power level, the number of nodes changed starts at a low number and then goes up, as the combined power of all the sacrificial nodes falls below the threshold value. As we keep changing the sacrificial nodes, the combined power of the sacrificial nodes goes above the threshold value, at which point the number of nodes to be changed decreases. Also, it is obvious that the total number of nodes changed is inversely proportional to the power level of the sacrificial nodes.

CHAPTER VII

CONCLUSION AND FUTURE WORK

The network setup in SENSE was composed of 289 nodes. Three nodes were designated to be the source, destination and attacker nodes. Four DRA's were setup yielding a total of 72 available sacrificial nodes. Due to limitations in SENSE the addition of sacrificial nodes was performed manually. The first set of simulations, for modeling the response process, was to study the effect of attacker power on the network's response. The five runs completed showed, as expected, that the number of sacrificial nodes needed to respond to an attack is proportional to the attacker power. The second set of simulations was to study the effect of the sacrificial node power level on the network's response. The four runs completed proved that the number of node changes required to respond to an attack is inversely proportional to the node's power level.

The attacker might change his attack pattern, even before the sacrificial nodes succeed in responding against the wormhole attack. The study of change in behavior of the attacker is recommended for future work.

.

REFERENCES

[1] Introduction to Wireless Sensor Networks, *http://www.worldscibooks.com/comps ci/etextbook/6288/6288_chap1.pdf*, (Date of access: September 23, 2011).

[2] Chong Eik Loo, Mun Yong Ng, Chirstopher Leckie, and Marimuthu Palaniswami, "Intrusion detection for routing attacks in sensor networks," *International Journal of Distributed Sensor Networks*, Vol.2, No.4, pp313-332, 2006.

[3] Xiaojiang Du and Hsiao-Hwa Chen, "Security in Wireless Sensor Networks", *IEEE Wireless Communications*, Vol. 15, No. 4, pp. 60-66, 2008.

[4] Jorge Granjal, Ricardo Silva and Jorge S. Silva, "Security in Wireless Sensor Networks", *Centre for Informatics and Systems of the University of Coimbra*, 2008.

[5] Chris Karlof and David Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *AdHoc Networks Journal*, Vol. 1, No. 2–3, pp. 293–315, September 2003.

[6] Stargate: X-Scale, Processor Platform, http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/Stargate_Datasheet.pdf, (Date of Access : July 2011)

[7] Fit-PC2i specifications, http://www.fit-pc.com/web/fit-pc2/fit-pc2i-specifications/, (Date of Access: July 2011)

[8] Seyit A. Camtepe and Bulent Yener, "Key distribution mechanisms for wireless sensor networks: a survey," *Rensselaer Polytechnic Institute, Troy, New York, Technical Report TR-05-07*, 2005.

[9] Elaine Shi and Adrian Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 38–43, December 2004.

[10] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker, "Mitigating routing misbehavior in mobile ad hoc networks", *Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 255–265, 2000.

[11] Sonja Buchegger, Jean-Yves Le Boudec, "Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks", *Proceedings IEEE Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pp. 403–410, 2002.

[12] Adrian Perrig, Robert Szewczyk, J. D.Tygar, Victor Wen and David E. Culler, "SPINS: security protocols for sensor networks", *Proceedings ACM Seventh Annual International Conference on Mobile Networking and Computing*, pp. 189-199, 2001.

[13] Loukas Lazos and Radha Poovendran, "Serloc: Robust localization for wireless sensor networks," *ACM Transactions on Sensor Networks*, Vol. 1, No. 1, pp. 73–100, 2005.

[14] Yang Xiao, *Security in Sensor Networks*, CRC Press, 2006.

[15] Saurabh Ganeriwal, Srdjan Capkun, Chih-Chieh Han, and Mani B. Srivastava, "Secure time synchronization service for sensor networks," *Proceedings 4th ACM Workshop on Wireless Security*, pp. 97–106, 2005.

[16] Ana Paula R. da Silva, Marcelo H. T. Martins, Bruno P. S. Rocha, Antonio A. F. Loureiro, Linnyer B. Ruiz, and Hao Chi Wong, "Decentralized intrusion detection in wireless sensor networks," *Proceedings 1st ACM International Workshop on Quality of service & Security in Wireless and Mobile Networks*, pp. 16–23, 2005.

[17] Ilker Onat and Ali Miri, "An intrusion detection system for wireless sensor networks," *Proceedings IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Vol. 3, pp. 253–259, 2005.

[18] Vijay Bhuse and Ajay Gupta, "Anomaly intrusion detection in wireless sensor networks," *Journal of High Speed Networks*, Vol. 15, No. 1, pp. 33–51, 2006.

[19] Yih-Chun Hu, Adrian Perrig and David B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks", *Proceedings ACM Workshop on Wireless Security*, pp. 30-40, 2003.

[20] Srdjan Capkun, Levente Buttyan and Jean-Pierre Hubaux, "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks", *ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 21-32, 2003.

[21] Loukas Lazos and Radha Poovendran, "Serloc: Secure Range-Independent Localization for Wireless Sensor Networks", *Proceedings of the ACM Workshop on Wireless Security*, pp. 21-30, 2004.

[22] Lingxuan Hu and David Evans, "Using Directional Antennas to Prevent Wormhole Attacks", *Proceedings 11th Network and Distributed System Security Symposium*, pp. 21-30, 2003.

[23] Weichao Wang and Bharat Bhargava, "Visualization of wormholes in sensor networks", *Proceedings ACM workshop on Wireless Security*, pp. 51-60, 2004.

[24] Issa Khalil, Saurabh Bagchi and Ness B. Shroff, "LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks", *Proceedings International Conference on Dependable Systems and Networks*, pp. 612-621, 2005.

[25] Lijun Qian, Ning Song and Xiangfang Li, "Detecting and Locating Wormhole Attacks in Wireless Ad Hoc Networks through Statistical Analysis of Multipath", *IEEE Wireless Communications & Networking Conference*, Vol. 4, pp. 2106-2111, 2005

[26] Ad hoc On-Demand Distance Vector (AODV) Routing, http://www.ietf.org/rfc/rfc3561.txt, (Date of last access: November 25, 2011)

[27] Ritesh Maheshwari, Jie Gao and Samir R. Das, "Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information", *Proceedings IEEE INFOCOM 26th International Conference on Computer Communications*, pp.107-115, 2007.

[28] Edith C. H. Ngai, Jiangchuan Liu and Michael R. Lyu, "On the Intruder Detection for Sinkhole Attacks in Wireless Sensor Networks", *Proceedings IEEE International Conference on Communications*, Vol. 8, pp. 3383-3389, 2006

[29] Takeshi Takahashi, Hiroaki Hazeyama, Daisuke Miyamoto and Youki Kadobayashi, "Taxonomical Approach to the Deployment of Traceback Mechanisms", *Proceeding Baltic Congress on Future Internet Communications*, pp. 13-20, 2011

[30] Rashid H. Khokhar, Md A. Ngadi and Satria Mandala, "A Review of Current Routing Attacks in Mobile Ad Hoc Networks", *International Journal of Computer Science and Security*, Vol. 2, No. 3, pp. 18-29, 2008.

[31] Deepayan Chakrabarti and Christos Faloutsos, "F4: Large Scale Automated Forecasting Using Fractals", *Proceedings ACM Eleventh International Conference on Information and Knowledge Management*, pp. 2-9, 2002

[32] Enrique J. Duarte-Melo and Mingyan Liu, "Data-gathering wireless sensor networks: organization and capacity", *Computer Networks,* Vol. 43, No. 4, pp. 519–537, 2003

[33] Antonio Capone, Luigi Fratta and Fabio Martignon, "Bandwidth Estimation Schemes for TCP over Wireless Networks", *IEEE Transactions on Mobile Computing*, Vol. 3, No. 2, 2004.

[34] Seung-Hwan Lim, Heejin Park and Sang-Wook Kim, "Using multiple indexes for efficient subsequences matching in time series databases", *Information Sciences*, Vol. 177, No. 24, pp. 5691-5706, 2007.

[35] Yang-Sae Moon, Kyu-Young Whang and Wook-Shin Han, "General match: a subsequence matching method in time series databases based on generalized windows", *Proceedings ACM SIGMOD International Conference on Management of Data*, pp. 382-393, 2002.

[36] SENSE - Sensor Network Simulator and Emulator, http://www.ita.cs.rpi.edu/sense/index.html, (Date of last access: December 2, 2011).

APPENDIX A

The data collected to support the proposed approach of finding when the attacker is overloaded and also when an alternate path is found is presented here for two other power levels of the attacker: 0.003 and 0.0035.

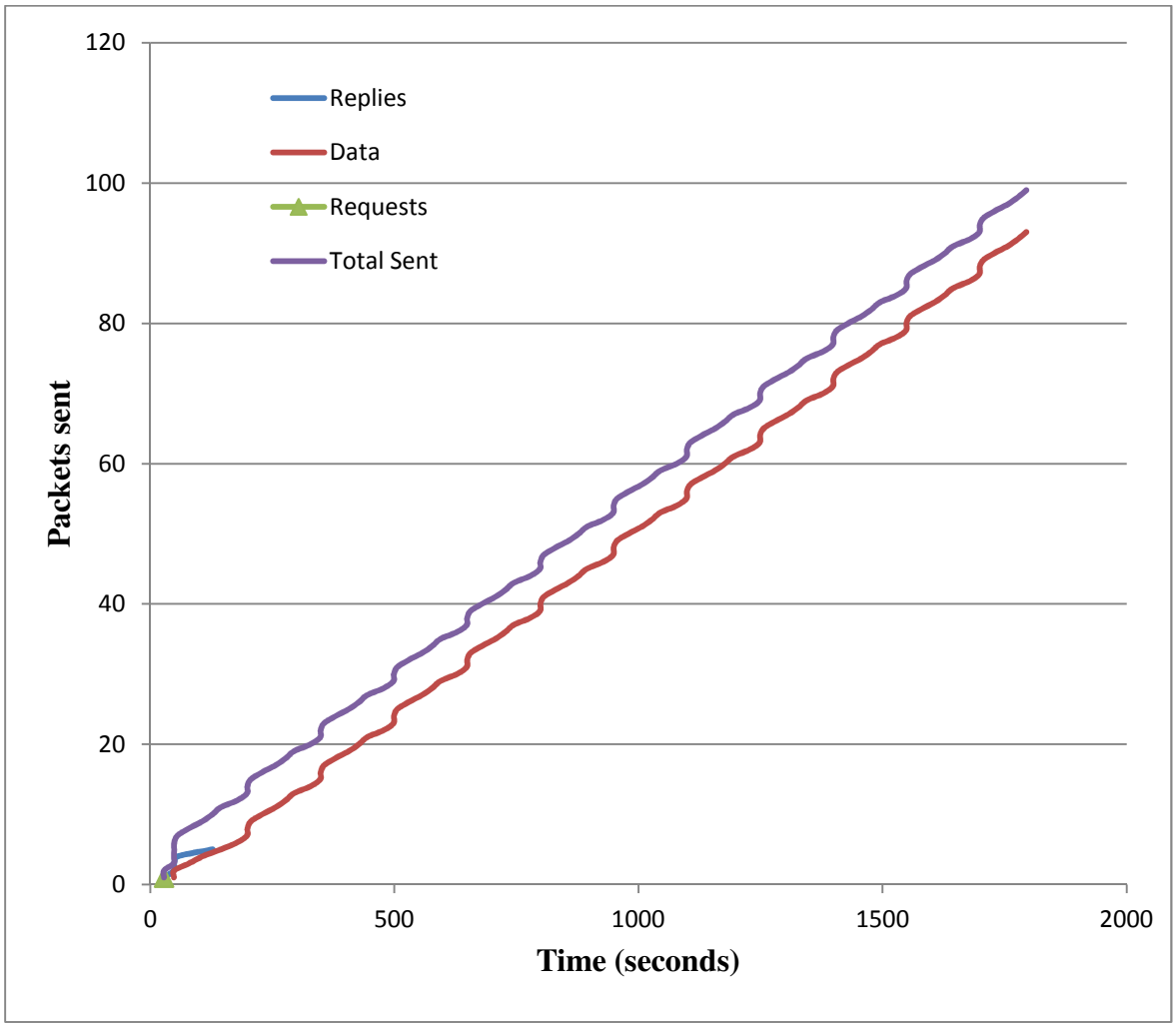

Figure A-1. Packets sent by the attacker at power level 0.003

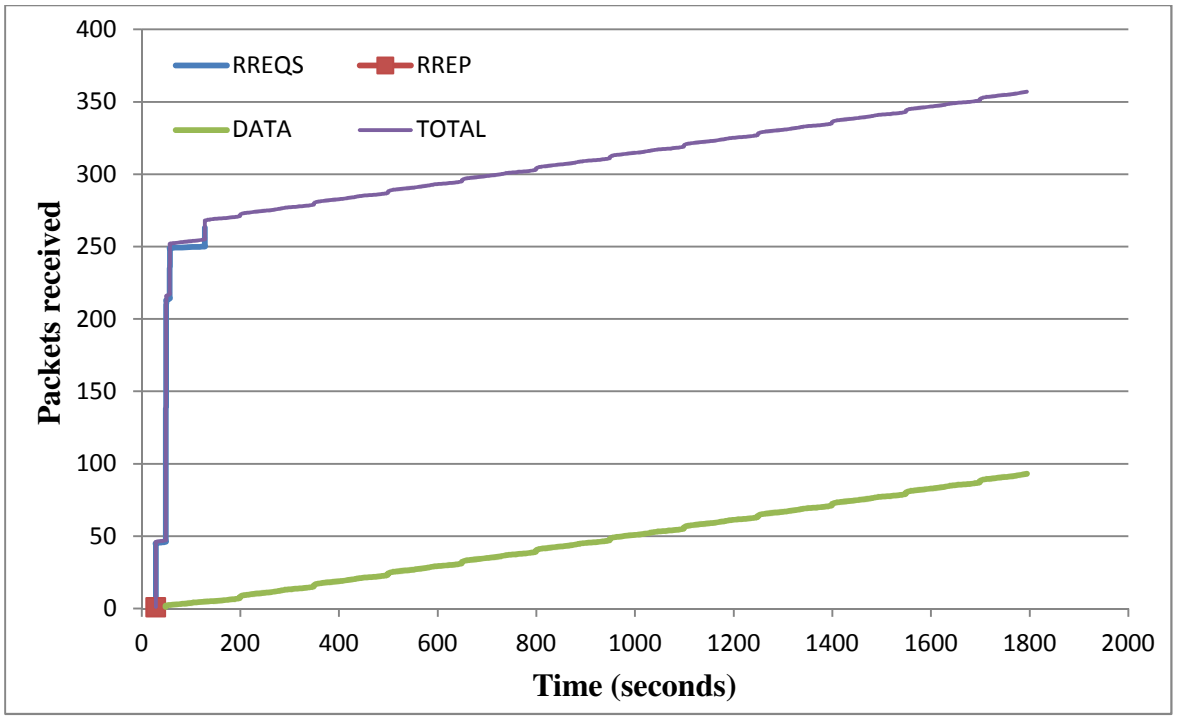

Figure A-2. Packets received by the attacker at power level 0.003

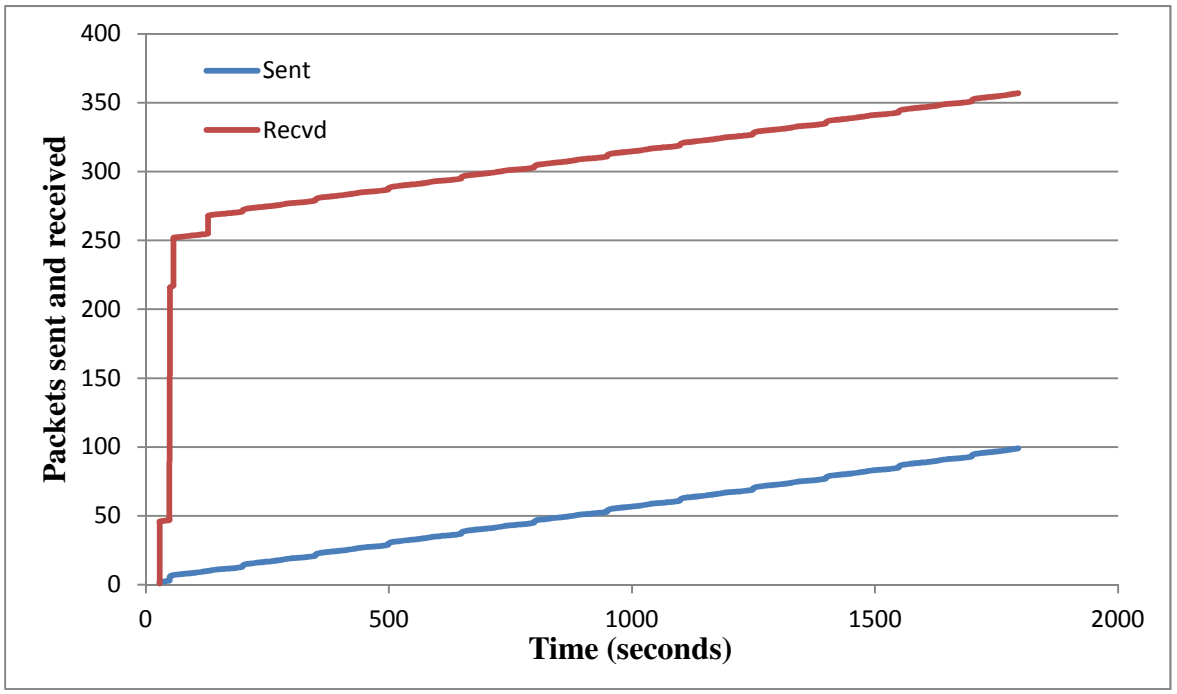

Figure A-3. Total packets sent and received by the attacker at power level 0.003

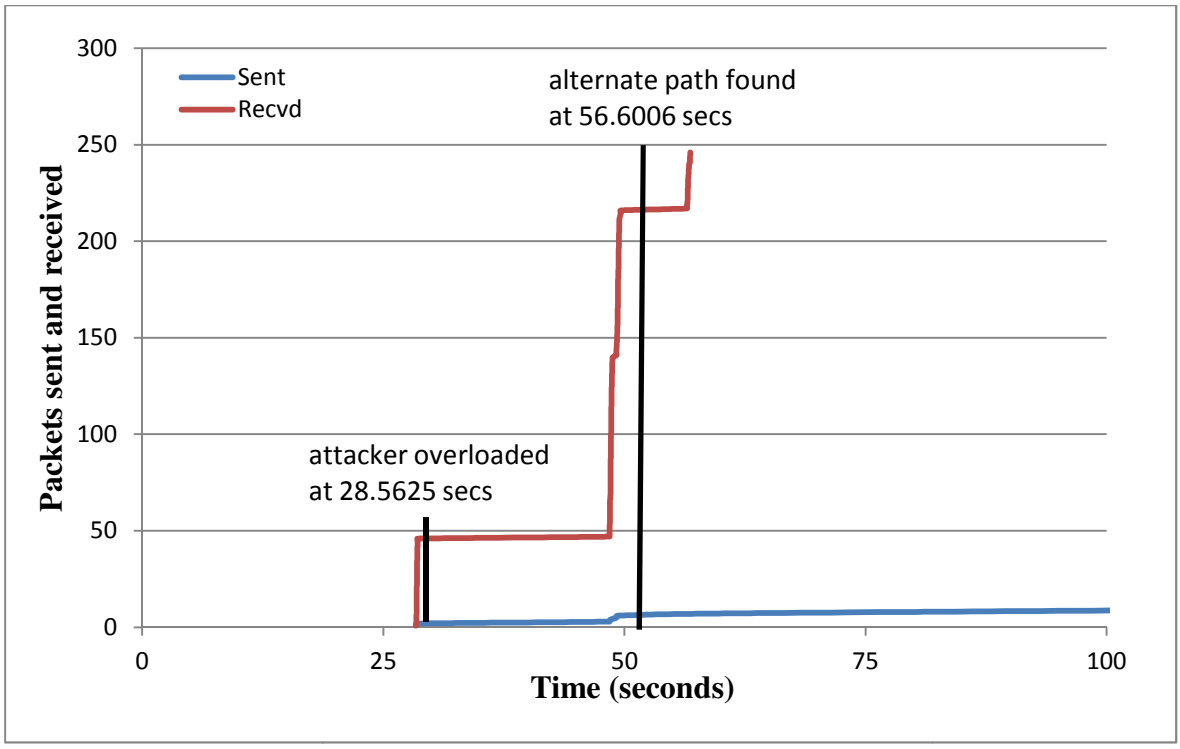

Figure A-4. Time when attacker is overloaded and alternate path found at power 0.003

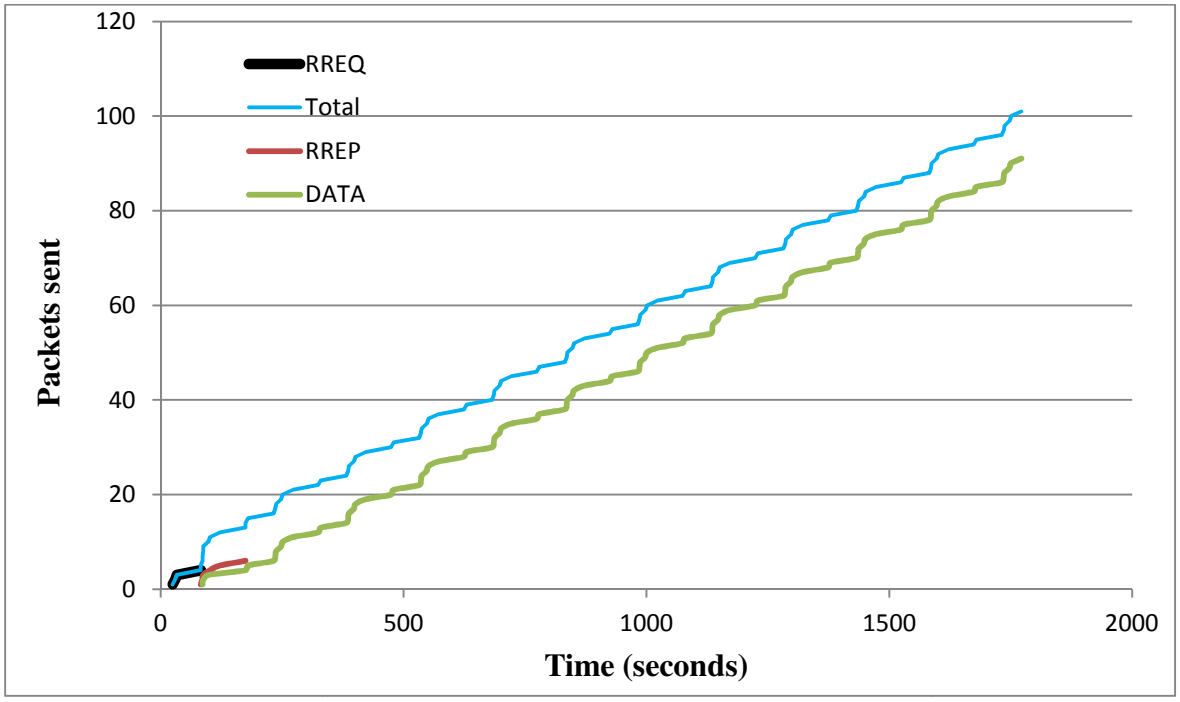

Figure A-5. Packets sent by the attacker at power 0.0035

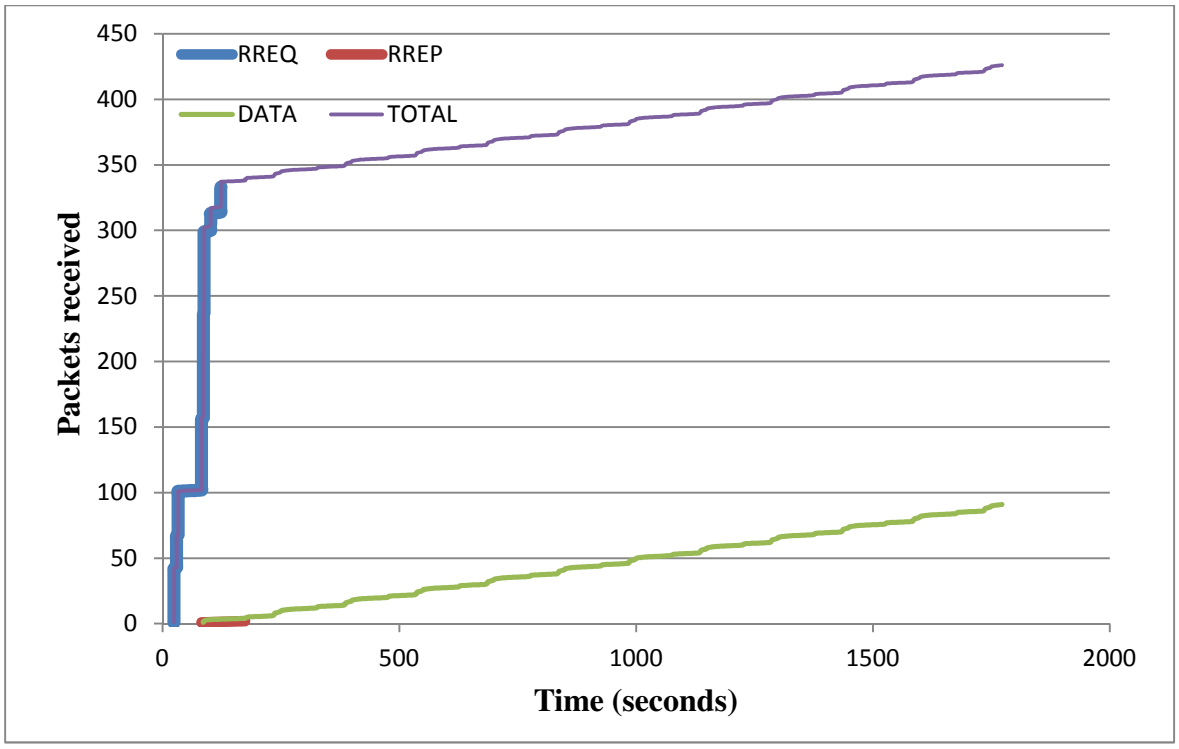

Figure A-6. Packets received by the attacker at power 0.0035

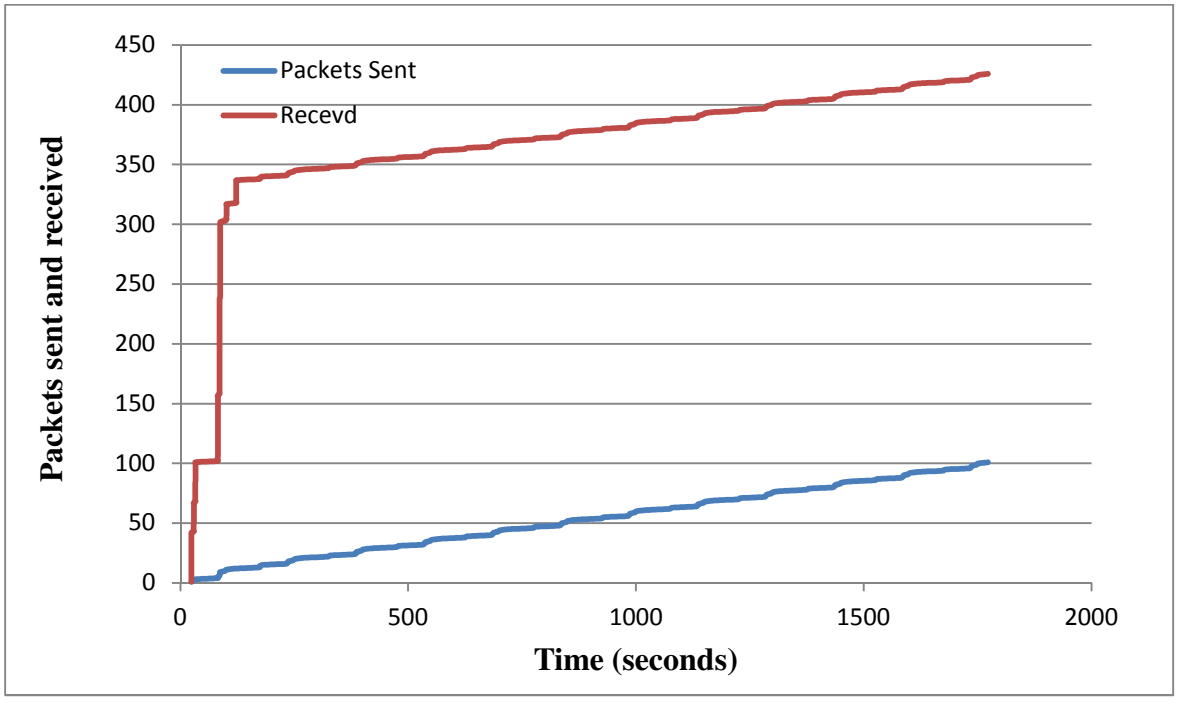

Figure A-7. Total packets sent and received by the attacker at power 0.0035

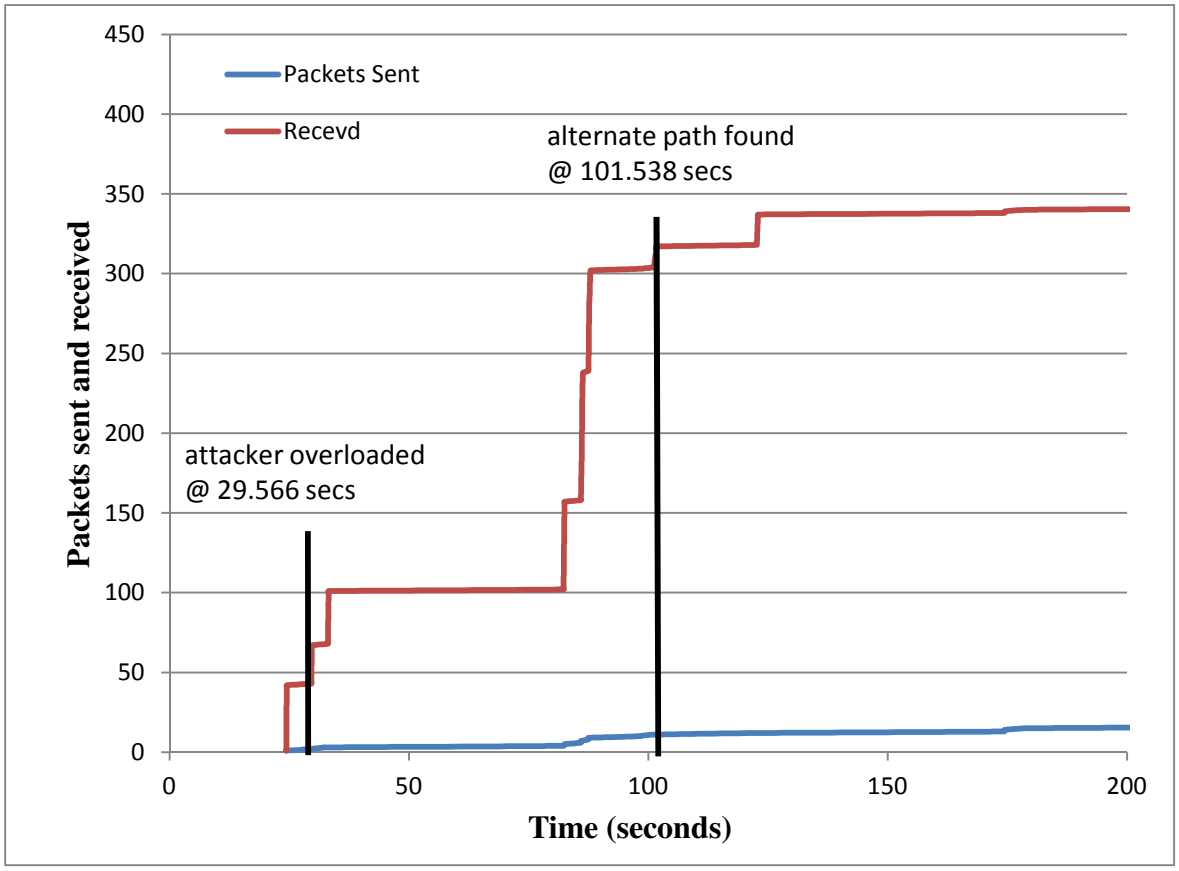

Figure A-8. Time when attacker is overloaded and alternate path found at power 0.0035

VITA

Balambika Vinod

Candidate for the Degree of

Master of Science

Thesis: RESPONDING TO AN ATTACK IN SENSOR NETWORKS

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in May, 2012.

Completed the requirements for the Bachelor of Engineering in Computer Science at Velammal Engineering College, Chennai, TamilNadu/India in 2006.

Experience: Project Engineer, Wipro Technologies, 2006-2007

Name: Balambika Vinod Date of Degree: May 2012

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: RESPONDING TO AN ATTACK IN SENSOR NETWORKS

Pages in Study: 58 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study:

Responding to an attacker who has infiltrated a network has received little attention. In this thesis we propose an appropriate response model to a wormhole attack in sensor networks. The response to the attack will maximize the benefit to the defender sensor node by obtaining an alternate path that will avoid the wormhole, thus mitigating the effects of the attack. This is achieved by using sacrificial nodes to engage the attacker while alternatives are explored by the defender. This approach has the added advantage of expending the attacker's resources and time. A probabilistic response model that determines the success of an alternative route being discovered is developed. A monitoring scheme is also proposed to ensure that sufficient sacrificial nodes are engaging the attacker.

Findings and Conclusions:

The study was validated using the wireless sensor network simulator and emulator SENSE. Results show that the wormhole attacker's presence in the route is eliminated when he reaches his maximum utilization or capacity. In addition, as theorized the number of sacrificial nodes needed to respond to an attack is directly proportional to the attacker's power and inversely proportional to that of the sacrificial nodes.

ADVISER'S APPROVAL: Dr. Johnson Thomas