DETECTING SELECTIVE FORWARDING ATTACKS IN WIRELESS SENSOR NETWORKS

By

VENKATA M. MULPURU

Bachelor of Engineering in Computer Science and

Osmania University

Hyderabad, India

2005

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, 2008

DETECTING SELECTIVE FORWARDING ATTACKS

IN WIRELESS SENSOR NETWORKS

Thesis Approved:

Dr. Johnson P. Thomas

Thesis Adviser

Dr. Venkatesh Sarangan

Dr. Nohpill Park

Dr. A. Gordon Emslie

Dean of the Graduate College

TABLE OF CONTENTS

Chapter Page

IV. FINDINGS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Wireless sensor networks consist of low-cost, low-power electronic devices called sensor motes. The sensor motes are densely deployed to cooperatively detect and transmit back environmental and physical conditions of the environment in which they are deployed. As they can be deployed in a variety of environments, sensor networks have many applications including military applications, environmental applications, health applications, home applications and other commercial applications [1][2].

Motes usually contain a processor, transceiver, sensor board and a power source. The network also consists of a central base station and optional cluster heads. Sensor boards on the motes contain a range of possible sensors that can detect temperature, light intensity, humidity, mechanical stress and many other values. The data collected is analyzed by the processor and resulting values are transmitted using the transceiver to the base station through multi-hop routing protocols. Sensor networks suffer from inherent limitations like low power, low memory, low computational capability, high degree of failure and unpredictable environmental conditions in which they are deployed. Unlike normal motes, cluster heads and base station have higher memory and processing capabilities.

Sensor networks are prone to a variety of attacks that include spoofed, altered or replayed routing information, selective forwarding, sinkholes, black-holes and worm-holes [3]. These attacks can cripple a sensor network and render them useless. Many detection algorithms are available for normal networks [4][5][6]. These intrusion detection algorithms cannot be used in sensor networks due their resource constrained nature. The algorithms must involve low computation, communication and memory usage to ensure the longevity of the network. Due to this limitation the detection algorithm involves a tradeoff between resource consumption and, accuracy and speed of detection.

We focus our work on detecting selective forwarding attacks in sensor networks. Previous work done in detecting selective forwarding attacks [3][7] involves the use of multi-path routing and complex key mechanisms. These approaches introduce high energy consumption due to processing and communication requirements. Our approach to detecting the selective forwarding attacks has two main goals; decreasing energy consumption and achieving high accuracy. We also need to be able to trace back the attacker to aid in countermeasures to be taken in the network.

Our solution for selective forwarding attack involves use of an acknowledgment based scheme like the one proposed in [7]. We limit the communication overhead involved by localizing the acknowledgments that are transmitted in the network. The acknowledgments transmitted are used to identify dropped packets in a network. We introduce a metric called maliciousness indicator (MI) which can tell us if a node is an attacker or not. A high value of maliciousness indicator indicates that a node is more

malicious and thus more likely an attacker. Using such a metric might decrease the accuracy of detection due to increased false positives. To counter this we have a maliciousness reduction constant (MRC), which can reward a node for normal behavior by reducing the maliciousness indicator on the motes.

Any lost acknowledgements in the network might decrease the effectiveness of our algorithm. To solve this problem we make use of an acknowledgement method and propose other solutions that can be used. As maliciousness indicator is an important metric and must be tamper proof, we do not trust any node with its own maliciousness indicator value. Each node's MI is maintained by other nodes in the network. This node increases and decreases the MI based on the behavior it observes.

In Chapter II we review the previous done in detecting selective forwarding attacks in sensor networks and also introduce hardware used in wireless sensor networks. In Chapter III, we provide the details of our approach to detect selective forwarding. In Chapter IV we provide and compare the results from our experiments running the algorithm with previous work. We conclude the thesis with Chapter V.

CHAPTER II

REVIEW OF LITERATURE

In this chapter we describe the radio hardware used on the motes and previous work done on selective forwarding.

2.1. CC2420 Radio

The type of motes in our network is TelosB [8]. Each TelosB mote uses a CC2420 transceiver [10] which follows the IEEE 802.15.4 [9] standard for personal area networks. It also supports the Zigbee standard [11] which builds on the IEEE 802.15.4 standard to add upper level layer standards. The IEEE 802.15.4 standard specifies that a link quality indicator (LQI) is to be calculated for each packet a CC2420 radio receives, along with other information about the packet like received signal strength indicator (RSSI). LQI calculation is based on the chip error rate observed during communication between two motes in the network [9]. The computed value is encapsulated in the packet header and transmitted to the applications running in the network. Applications like routing can make use of this value to differentiate between good and poor links in a network.

A higher LQI value observed on a link indicates a good communication link. On the other hand a low LQI value on a link means that the link is poor in quality. In [12], it is identified that the packet reception rate (PRR) observed on a communication link has strong correlation with LQI value when sufficiently large number of LQI values are collected and averaged. The packet reception rate thus computed can be used to estimate the number of packets lost due to bad link quality in a network. Estimating the normal packet loss due to collisions and poor communication links can be useful in separating it from malicious dropping of packets in a network.

2.2. Selective Forwarding Attacks

A wireless sensor network employs different multihop routing protocols to transmit data from the nodes sensing the information to the base station. A packet originating at a node must be received and retransmitted by other nodes on the path towards the base station. This proper forwarding of packets is integral to the proper functioning of the routing protocol and hence the proper functioning of the wireless sensor network.

A malicious node on the path towards the base station might not forward the packets properly to other nodes. If an attacker chooses to drop all the packets it is supposed to forward, it is called a black-hole [3]. Such an attack is easily detected due to the huge loss of packets easily detected by the base station. Rather than dropping all the packets, an attacking node can refuse to forward some of the packets that it should forward. The loss of information and control packets being transmitted in the network will be high in such cases. This kind of attack where a node selectively drops some packets is called selective

forwarding attack [3]. A node performing selective forwarding is hard to detect [3] as the number of messages being dropped is chosen to minimize the risk of detection.

2.3. Previous Work and Our Approach

In [7] the authors propose an acknowledgment based scheme to detect selective forwarding attack. In their algorithm, each packet being generated in the network has two fields added to its header; ACK_span and ACK_TTL. At each hop the ACK_span value in a packet is decreased. When it reaches zero, an acknowledgement messages is generated and transmitted on the path towards the source. This acknowledgement messages has a time to live value of ACK_TTL. Each node which forwarded the packet waits for a specified amount of time to see if an acknowledgement is received. If the packet is not received within the specified time period, it raises an alarm packet and transmits it back to the source. All the nodes raise such alarm packets and they get transmitted to the source. The source node analyzes all the alarm packets received and decides if a node not sending the acknowledgement message is an attacker. If an attacker is found, the source informs the base station.

There are many drawbacks in the approach taken in [7]. There is an acknowledgement message being transmitted for every normal message being sent. Furthermore alarm packets are sent back all the way to the source or the base station. This increases the communication overhead for every packet generated by the source nodes in a network. As long battery life is a key component in the success of a wireless sensor network; high

communication overhead reduces the battery life. The approach also uses a synchronized clock for time-stamping the alarm packets. This is necessary in the algorithm as many alarm packets might be generated by a node and the source needs to know which of them is a fresh packet. A synchronized clock is difficult to implement in sensor networks and requires additional overhead, again consuming battery life. The third drawback in this approach is that it does not consider the normal packet loss in a network. Any wireless network has normal packet loss due to poor link qualities. This loss must be quantified and taken into account when deciding if a packet is intentionally dropped. As such quantification is not made, the accuracy of detection decreases rapidly with increase in channel error rate.

Our approach in detecting selective forwarding attacks also uses an acknowledgment based scheme for detecting attackers. We try to reduce the communication overhead in the network by decreasing the required number of acknowledgment messages and localizing them in the network. There is no communication with the source or the base station. Missing packets in a network increase the maliciousness indicator of node maintained by its neighbors. The intentionally dropped packets and normal loss of packets are differentiated using the correlation between LQI and PRR. This will help improve the accuracy of detection. Our approach also decreases the overhead involved by eliminating the use of a synchronized clock.

CHAPTER III

METHODOLOGY

3.1. Network Architecture

We consider a wireless sensor network divided into clusters. Each cluster has a cluster head which has greater computational and communicational capabilities. These cluster heads can be Stargate gateways [13]. Normal sensor nodes in the network can be any motes like micaZ [14] and TelosB [8], which follow 802.15.4 [9] and Zigbee [10] specifications. A central base station collects all the data and controls the sensor network. This base station can be a server class computer. Figure 1 shows the network architecture. The base station is denoted by BS. Dotted rectangles represent clusters of normal sensor nodes with a cluster head denoted by CH.

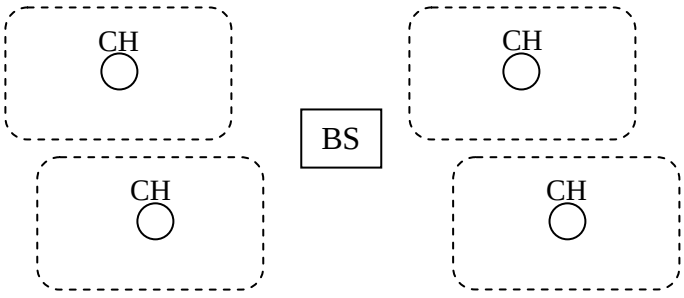

Figure 1: Network architecture

Normal motes sense the information from their environment and report it to the base station using a multi-hop routing protocol. Any multi-hop routing protocol like [15] [16] developed for sensor networks can be used for this purpose. Apart from transmitting data packets, normal nodes detect attackers in the network and use report the attacker id to the cluster heads.

A cluster head uses multi-hop routing protocol to communicate with other cluster heads and the bases station. Any communication between among cluster heads and the base station is to be secure for obvious reasons. Due to the greater computation power available [13] complex encryption algorithms which cannot be used by normal nodes can be used by the cluster heads and the base station.

3.2. Assumptions

We assume a static network (no node movement). The communication among the cluster heads and the base station is assumed to be secure due to the greater computational power at their disposal. We assume that there is no attack taking place within the first few minutes when the network is deployed. Any attack can be avoided by properly planning the deployment of the network.

3.3. Proposed Approach

Our approach involves the use of an acknowledgment based scheme to detect selective forwarding. This approach is similar to the one used in [7]. To improve efficiency we use a localized cumulative acknowledgement based scheme. The accuracy of detection will also be increased with the help of a probability based metric to decrease the probability of false alarm.

3.3.1. Relay of Acknowledgements

Consider the part of the network presented in figure 2. A, X and B are nodes on the path taken by the data towards the base station. A forwards the data it receives to X, and X forwards it to B. To detect if X is an attacker we need to send acknowledgements from B to A for the packets that are successfully forwarded by X. In a network that is sufficiently dense we will be able to find two nodes M and N that are in communication range with both A and B. These nodes act as the intermediaries for transmitting the acknowledgements from B to A. Hence the behavior of a node is reported by other independent nodes who cannot communicate with each other. Using these two intermediate nodes, we are able to localize the transmission of acknowledgements and ensure low communication. If an intermediate node is compromised, the other can relay back fair acknowledgements. Using two intermediate nodes reduces the risk involved; as probability of two nodes being compromised is less than probability that one node is compromised. And the two intermediate nodes should not be able to communicate with

each other. Doing so would mean that they can collude and tamper the packets in a similar way so as to make it difficult to detect.

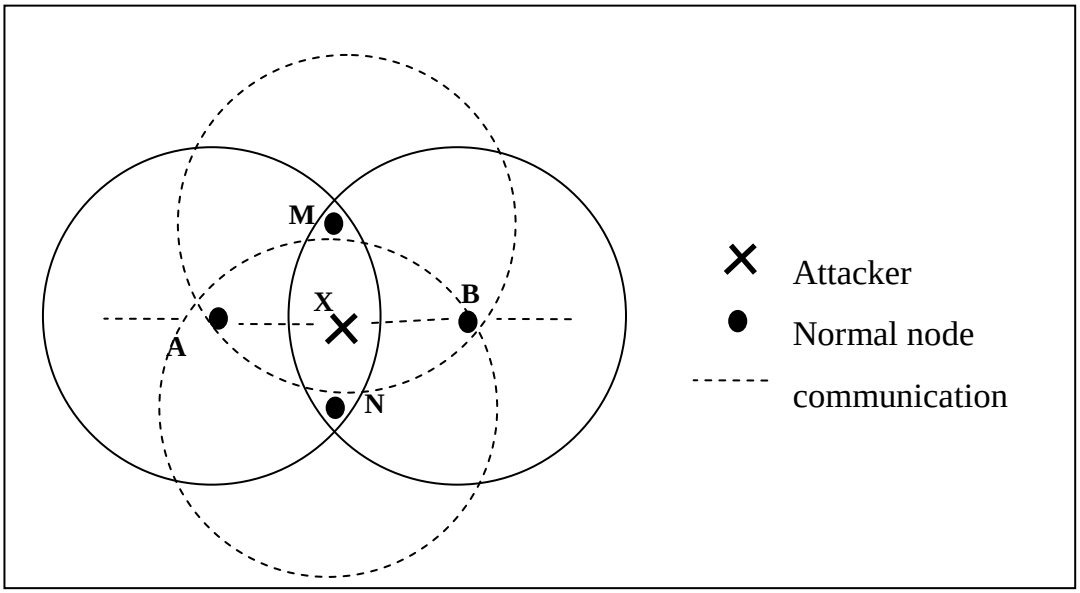

Figure 2: Communication model

Considering a unit disc radio model of radius r, for our approach to work the following constraints must be satisfied:

AX ≤ r – node X should be within communication range of node A.

BX ≤ r – node B should be within communication range of node X.

AN ≤ r – node N should be within communication range of node A.

BN ≤ r – node B should be within communication range of node N.

AM ≤ r – node M should be within communication range of node A.

BM ≤ r – node B should be within communication range of node M.

MN > r – node M should not be within communication range of node N.

AB > r – node A should not be within communication range of node B.

For our approach the key notion is of node density. Inside a fixed region, one cannot have too dense a region or the above constraints will be violated. We propose to define the densities and regions needed for our approach. This will be derived during the remainder of the project.

We can reduce the number of acknowledgements that need to be transmitted by using cumulative acknowledgements. Instead of sending an acknowledgement for every packet received, one acknowledgement can be sent for every *n* seconds. This message would contain the number of packets received in the last *n* seconds. Here, *n* is an integer dependent on the data rate in the network. The higher the data rate, the higher *n* should be to minimize energy consumption. For example, a sensor network that monitors temperature would have a low data rate compared to a network that senses light intensity, as temperature cannot change as rapidly as light intensity. Hence, the network monitoring light intensity would have a higher *n* compared to the network monitoring temperature.

3.4. Packet Reception Rate Vs. LQI

Due to the nature of the medium involved for communication, the efficiency of communication in a sensor network is not perfect. Environmental conditions and the distance between the nodes also limit the efficiency of the communication. We need to differentiate between the normal packet loss occurring in a network and the intentional dropping of packets. To achieve this goal, we use the link quality indicator (*LQI*) [9]

computed by the CC2420 radio [10]. It has been shown that when sufficiently large number of *LQI* values are collected and averaged, there is a strong correlation between the packet reception rate (*PRR*) and *LQI* [12]. Figure 3 shows the second order trend line that has been obtained by performing regression analysis of the data we obtained from the authors of [12] and also from our experimental observations.

Considering the resource constrained nature of the sensor nodes, we used a second order curve to perform the regression analysis. The resulting equation is

$$PRR = -0.06 * (LQI)^2 + (11.42 * LQI) - 486.1$$

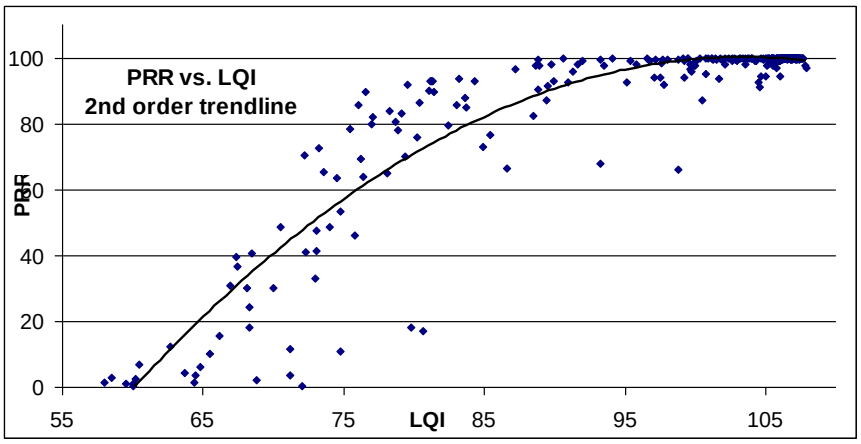

Figure 3: Packet reception rate vs. LQI

The *PRR* that is computed is adjusted to cope for the error induced by regression analysis. If the *PRR* of a link is found to be higher than the value that is computed through the second order equation, the expected *PRR* is adjusted accordingly. This process is described in detail in the following sections.

3.5. Proposed Algorithm

The proposed acknowledgement based scheme to detect selective forwarding attacks in a sensor network can be divided into two phases. The discovery and learning phase and the attack detection phase. The actual working of these phases is presented in the following subsections.

3.5.1. Learning and Discovery Phase

As discussed earlier, when a wireless sensor network is deployed, we assume that there is no attack taking place in the network. Typically the routes towards the base station are established in this phase and the various tables like the neighbor table and routing table are initialized and populated.

For our purpose we increase the amount of work done in this phase to include the discovery of intermediate nodes and also the discovery of routes between intermediate nodes to share their observations with each other.

3.5.1.1. Discovering Intermediate Nodes

In many routing algorithms a beacon packet is broadcast by each node in the network. This packet helps in the discovery of a node's neighbors. Once the neighbors are discovered, the routes towards the base station are established.

We now describe a notation for simplifying the discussion about the intermediate discovery process. Let X, A be the ID's of nodes in the network. We define,

$Parent(X)$ as the ID of the node to which the packets arriving at X are forwarded.

$Neighbors(X)$ as the set of all the neighbors of X.

$Intermediate(X, A)$ as the set of intermediate nodes between X and A. Intermediate nodes between X and A can be defined as the nodes that can communicate with both X and A.

We introduce a new packet called the intermediate node discovery packet (INT_DISC). The purpose of this packet as the name implies is to help in the discovery of intermediate nodes. The structure of the packet is shown in figure 4. The first field in the packet contains the ID of the sender. The second field called the parent contains the value $Parent(Sender)$. The last field contains the set $Neighbors(Sender)$.

Sender	*Parent (Sender)*	*Neighbors (Sender)*

Figure 4: Intermediate node discovery (INT_DISC) packet structure

After all the neighbors of a node are discovered and routes towards the base station are established, the intermediate node discovery packet is transmitted. The routes towards the base station and the neighbor list might change due the discovery of new neighbors in a network. To counter this scenario, the INT_DISC packet is retransmitted at regular intervals.

We also change the neighbor table at each node as follows.

Neighbor ID	*Parent(Neighbor)*	*Neighbors(Neighbor)*

Figure 5: Neighbor Table

When a node, say with ID 'receiver', receives the INT_DISC packet, it checks the sender ID to see if it is equal to *Parent (Receiver)*. If the values are not equal the neighbor table is updated with the new information received from 'sender'. If the two values are found to be equal the process of discovering intermediate nodes begins. The neighbor table of 'receiver' is scanned to see if the *Parent (Sender)* occurs in it. If it occurs, the neighbor ID from that row in the neighbor table is obtained and added to the *Intermediate (Receiver, Parent (Sender))* set. Thus, the intermediate nodes set can be obtained. The algorithm for this process is given below. In the algorithm we denote P->Q as the value of the field Q in table P.

```
AT Receiver
IF (INT_DISC packet received)
        IF (Parent (Receiver) = Sender)
            FOR (each Neighbor in Neighbor Table)
                    IF (Parent (Sender) IN Neighbors (Neighbor))
                            Intermediate (Receiver, Parent (Sender)) += Neighbor
        ELSE
            Neighbor_Table (Sender)->Neighbors = Neighbors (Sender)
```

Figure 6: Algorithm for Intermediate Node Discovery

3.5.1.2. Learning Link Statistics

In the discovery and learning phase, adjustments to the relation between link quality and packet reception rate is performed. We know that the equation between LQI and PRR obtained by regression analysis is not perfect. There is an error in the equation that needs to be addressed.

During the initial few minutes, after the routes towards the base station have been established, the acknowledgment scheme is implemented. This presents the opportunity to tune the PRR value that is expected on a link. If the expected PRR on a link is less than or more than the actual PRR, the expected PRR is to be changed to reflect the actual PRR. This adjustment can be performed as we assume that there is no attacker within the first few minutes after the network is deployed.

All the observed PRR values during the first few minutes are accumulated during the learning and discovery phase of the network. These values are averaged and this averaged value becomes the new expected PRR of the link under observation. After this phase, there is no learning taking place in the network. This is due to the fact that the actual discrepancy in PRR might be due to an attacker.

3.5.2. Attack Detection Phase

This is the second phase of network operation. The actual attack detection takes place in this phase. This phase does not end and continues as long as the network is operational.

3.5.2.1. Identifying Packet Losses

Consider figure 2 shown above. When node A transmits data downstream towards the base station, X receives the data and forwards it to node B. There are two sets of packet losses occurring in this data transfer. One loss occurs when A sends data to X and the other occurs when X sends data to B. The loss occurring when A sends data to X cannot be considered in detecting the number of packets dropped by X. Only the loss occurring between X and B may be considered.

We need to quantify the number of packets lost on the link from A to X. This can be done if we obtain the LQI value of the link from A to X. Using the LQI value we can predict the packet reception rate of the link and deduct the loss from X. If symmetric links are

considered, LQI value of the link from A to X will be equal to the LQI value of the link from X to A. This LQI value can be used to compute the packet loss over the link.

If symmetric links are not considered, we need to identify the LQI value using other approaches. One approach would be to obtain the value reported by X to the intermediate nodes between X and *parent(A)*. X reports the LQI value of the link A to X to *parent(A)* as part of the detection process. Hence, using the value provided by its parent, A can predict the number of packets dropped on the link from A to X.

The value reported by *parent(A)* to A can be the average of the previous three LQI values. The average of the last three LQI values means that A has sampled a lot of packets and computed the LQI value. Each LQI value means that the number of messages was as equal to number of messages transmitted in *n* seconds in the network. Such average over a large sampling of packets can ensure the accuracy in the prediction of data loss from A to X.

Let l_1 be the number of packets lost on the link from A to X. We can obtain the value of l_1 by using the number of packets transmitted by A and, using regression analysis to get the expected PRR as shown in Section 3.4.

3.5.2.2. Processing Acknowledgment Messages

As previously described acknowledgements should be transmitted every *n* seconds. We call these messages ACK packets. ACK packets have the structure shown in figure 7.

Number of messages received (N_{ACK})	Expected packet reception rate (PRR)	Average link quality observed (LQI)

Figure 7: ACK message structure

Here N_{ACK} is the number of messages observed by the sender and the PRR field corresponds to the expected packet reception rate at the sender.

Consider the scenario from figure 2. ACK packets are sent by node B every *n* seconds to both M and N which are the intermediate nodes. The intermediate nodes relay these packets to the node A upstream (towards the source). We define the following notation. Let:

N_{ACK} be the number of messages that have been acknowledged by node B (obtained from ACK message).

N_{TX} be the actual number of messages that have been transmitted by node A.

L_T be the total number of packets lost.

l_1 be the number of packets lost on the link from A to X.

l_2 be the number of packets lost on the link from X to B.

a_l be the acceptable loss on the link X to B computed by A using the PRR received from node B.

u_l be the unacceptable loss on the link X to B computed by A.

The total number of packets lost, L_T, can be defined by the following equation.

$$L_T = N_{TX} - N_{ACK}$$

The actual number of packets lost on the link X to B is L_2 given by,

$$l_2 = N_{TX} - N_{ACK} - l_1$$

Acceptable loss of packets based on the PRR obtained from the ACK message can be computed using the equation,

$$a_l = (N_{TX} - l_1) * (1 - \frac{PRR}{100})$$

Unacceptable loss of packets can be computed using the equation,

$$u_l = l_2 - a_l$$

Here u_l number of packets are unaccounted for and are considered to be an act of selective forwarding by the node supposed to forward the packets.

The algorithm for processing ACK packets at the sender and the receiver is shown in figure 8.

AT Sender

EVENT: Packet Received *{*
$N_{ACK} = N_{ACK} + 1$
}

EVENT: *n* seconds elapsed *{*
ACK_MSG-> $N_{ACK} = N_{ACK}$
ACK_MSG->*PRR* = Expected PRR
Send (ACK_MSG)
}

AT Receiver

EVENT: Packet Sent *{*
$N_{TX} = N_{TX} + 1$
}

EVENT: ACK_MSG received *{*
N_{ACK} = ACK_MSG-> N_{ACK}
PRR = ACK_MSG->*PRR*

$l_2 = N_{TX} - N_{ACK} - l_1$

$a_l = (N_{TX} - l_1) * (1 - \frac{PRR}{100})$

$u_l = l_2 - a_l$
}

Figure 8: Algorithm for processing ACK packets

3.5.2.3. Probability Model

From the analysis of figure 3, we observe that the regression analysis does not predict the packet reception rate for a given link quality indicator value with 100% accuracy. This introduces the possibility of a normal node being falsely identified as an attacker performing selective forwarding. For example, the equation might give us an expected packet reception rate of 85% for an LQI value of 75. The actual packet reception rate

occurring in this particular case might be 80%. The node under consideration would falsely be categorized as malicious. To avoid this, we need a maliciousness indicator which provides some sort of leniency.

To achieve a model, we performed experiments on 2 TelosB motes and came up with the following distribution for the packet reception rate at various LQI value ranges. A mote was programmed to send 200 packets at 1 packet per second towards the other mote. Once all the transmissions are complete, we obtained the average LQI and PRR over the test run. More than a hundred tests were conducted using the scenario.

For LQI values between 80 and 90, we observed the distribution of packet reception rates as shown in figure 9. From the figure, we can observe that the highest concentration of packet reception rate is within a range of -1 to 1 times the standard deviation.

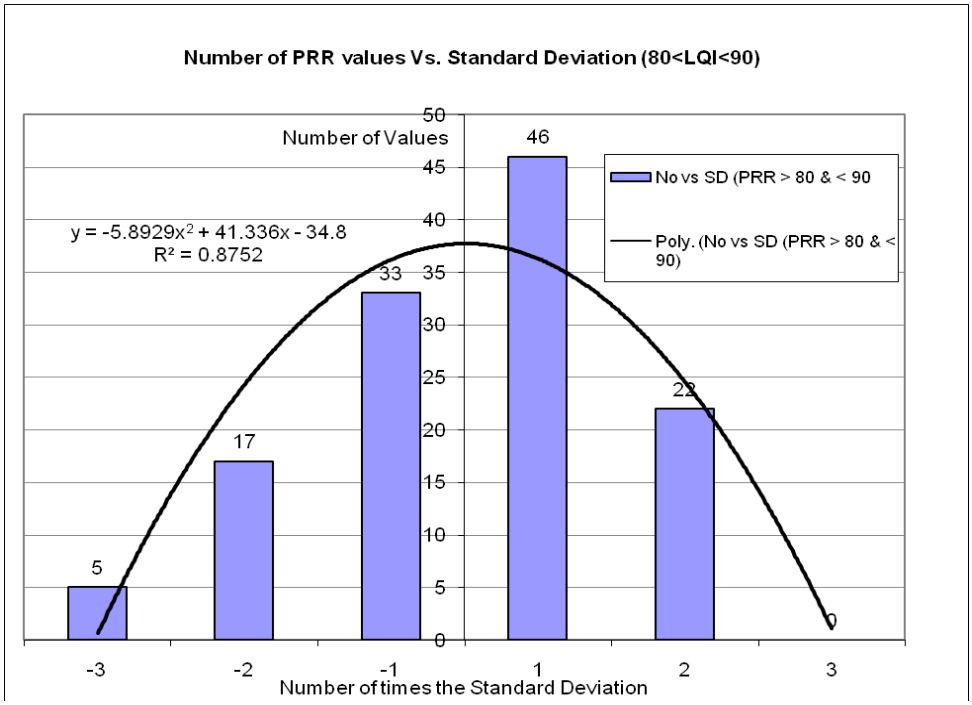

Figure 9: Distribution of PRR values from the mean

The number of times a PRR value greater than or less than the standard deviation was observed is very slim (as shown from the graph above). Using this information, we can approximate the probability that the observed PRR value is within one times the standard deviation and similarly, within two times the standard deviation. This probability value will be used to probabilistically increment the maliciousness indicator of the node.

Probability that observed PRR value is, $-1*\sigma < PRR < 1*\sigma$, where σ represents the standard deviation, can be given by the following equation.

Pr = Area under the curve from -1 to 1 / Total area under the curve

This can be rewritten using integrals to compute the area as,

$$\Pr_1 = \frac{\int_{-1}^{+1} f(x)dx}{\int_{-3}^{=3} f(x)dx}$$

In the above equation the function f(x) represents the distribution of packet reception rates in the network. Using regression analysis on the data provided in figure 9, we obtain the function,

$$f(x) = -5.892x^2 + 41.33x - 34.8$$

Similarly the probability that the observed PRR differs from the expected PRR by more than one times the standard deviation and less than two times the standard deviation can be computed by the following equation as Pr_2.

$$Pr_2 = \frac{\int_{+1}^{+2} f(x)dx + \int_{-1}^{-2} f(x)dx}{\int_{=3}^{-3} f(x)dx}$$

Similar experiments can be performed by varying the LQI values and a distribution of PRR values can be found. Using this distribution the probability that a node is malicious can be computed.

3.5.2.4. Maliciousness Indicator

As show in Figure 6, the unacceptable loss u_l can be computed. This unacceptable loss just gives us the raw packets that we think are maliciously dropped. In the previous section we showed why such raw numbers can lead to false positives and proposed a probability model to counter such scenarios.

We need an indicator that can say how probable it is that a node we are observing is a malicious node. This indicator is the maliciousness indicator. Each node receiving acknowledgements probabilistically computes the maliciousness indicator based on the

PRR values reported in the acknowledgement messages and the distribution of PRR calculated as shown in section 3.5.2.3.

Once the unacceptable loss u_l is computed we need to determine the amount by which the reported packet reception rate differs from the expected packet reception rate. Based on the difference computed we probabilistically increase the maliciousness indicator of each node.

Let τ_A^X be the maliciousness indicator of node X as computed by node A. Once the unacceptable loss and the probability that the observed PRR is correct are computed, we can use the following equation to compute the new maliciousness indicator of X as computed by A (τ_A^X).

$$\tau_A^X = \tau_A^X \left\{1 + \frac{u_l}{N_{TX}}\right\} * (1 + (1 - \Pr))$$

In the above equation Pr is the probability that the observed PRR is correct. For example let us consider that the difference between the observed PRR at B on the link XB differ by more than one times the standard deviation and less than two times the standard deviation. This information can be used to compute the value of Pr_2 as described in the previous section. Based on LQI ranges, mean PRR and standard deviation (see table 1), the respective probabilities will be added to determine the probability that the received PRR is an expected value for that LQI range. This value will be used to compute the new maliciousness indicator.

Using this probabilistic approach we can ensure that the nodes that report more deviant values of PRR will be punished more and those differing just a little will be given a chance to recover and thereby decreasing the overall false positives.

3.5.2.5. Maliciousness Reduction

Consider a scenario where a node drops a large number of packets at once due to unforeseen collisions in the network. If there are no acknowledgements for packets in the networks, this loss of packets will result in the increase of maliciousness indicator of a node. This indicator will then stay at the current value and likely cause the node to be confirmed as an attacker. Such scenarios must be reduced in the network.

To achieve the above goal, we introduce a maliciousness reduction constant. This constant will be used to reduce the maliciousness indicator of a node if it behaves normally. A node is said to behave normally if it does not drop any packets intentionally. The same condition can be expressed in terms of unacceptable loss, as having zero unacceptable loss of packets. A reduction constant will help reduce the number of false positives in the network. The amount by which maliciousness of a node is reduced is governed by following equation,

$$\tau_A^X = \tau_A^X * (1 - \rho)$$

Where, ρ is the reduction constant used in the network. Figure 10, describes the steps followed when an acknowledgement message is received.

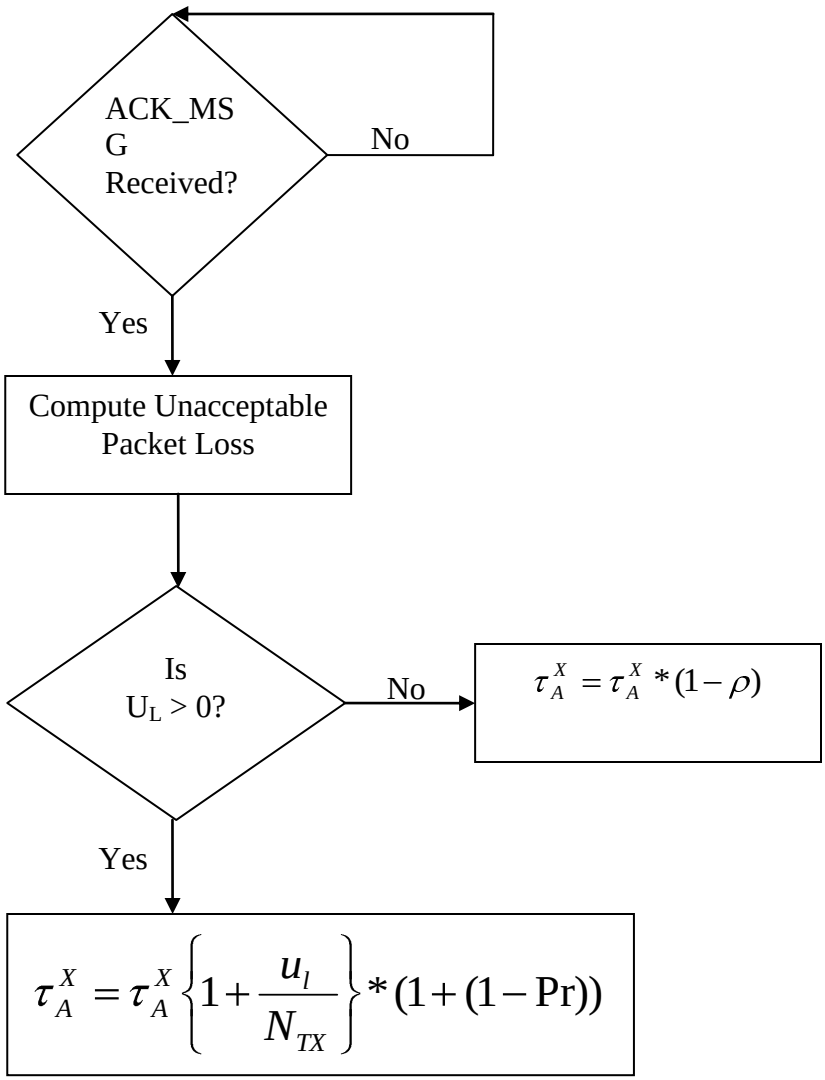

Figure 10: Flowchart showing execution flow when an ACK_MSG is received

3.5.2.6. Confirming an Attack

Each time an acknowledgement message is received; new maliciousness indicator values are computed. To confirm that an attack is actually taking place in the network, we need to define a threshold for maliciousness indicator. If the maliciousness indicator for a node

in the network crosses this threshold we say that a selective forwarding attack is taking place in the network.

Let T_a be the threshold at which we say a node is performing selective forwarding attack. In settling on an appropriate threshold we must consider the tradeoff between speed and accuracy. A low threshold might result in higher speed of detection but has the side effect of raising false alarms. On the other hand, a high threshold ensures that there are less false positives but sacrifices in speed are inevitable.

The value of T_a is also application specific. For example a network performing sensitive data gathering for the military might have a need to be more accurate to avoid loss of lives and resources. For this purpose we would have a higher threshold to decrease the number of false positives. A medical network sensing vital information about patients must be fast in detecting any attacks. In such a scenario, observing more false positives might be reasonable.

3.6. Other Issues

There are a few issues that need to be addressed to ensure that our detection approach works properly in real-time networks. We address them in this section.

3.6.1. Ensuring Acknowledgement safety

In any wireless network, there is a possibility that a message might be lost due to collisions. If an acknowledgement message is lost due to collisions, the speed of detection would be adversely affected. There is also a possibility that a malicious node is jamming the frequency to stop acknowledgement messages from reaching the nodes upstream.

As we make use of cumulative acknowledgements, the value of acknowledgement messages is increased. Therefore the tolerance for the loss of such information is drastically reduced. To address these issues each node receiving any acknowledgement messages must respond by sending an ACK message. A node sending the acknowledgement messages will retransmit the message until an ACK message is received from the nodes upstream.

Another approach to address the issue would be the use of a frequency hopping mechanism [17]. Each node can send the acknowledgement messages on a frequency channel other than the one used for normal communication. This channel will have fewer collisions as it is only used for sending acknowledgement messages.
If utmost security is needed, both the approaches can be used in conjunction. A predefined frequency hopping mechanism only known to the intermediate nodes and the node upstream can be used to relay acknowledgements on different channels.

3.6.2. Reporting an Attacker

When an attack is confirmed by a node in the network it must ensure that this information is passed on to the cluster head. The cluster heads can then pass this information back to the base station and take any actions it sees fit.

To ensure that the information concerning an attack is reported to the cluster head, the node detecting the attack floods the network with a high priority message containing data about the type of attack observed, the Id of the node performing the attack and the reporters Id (its own Id). This broadcasting might cause congestion in that network. To avoid the disruption of normal operation, a controlled flooding mechanism is used. Each node transmits to only two of its neighbors on the route towards the base station. These two nodes would append their Id to the message and forward the message.

When all such messages finally reach the cluster head, it can trace back the attacker. Apart from tracing back the attacker, the cluster head would also have a topology map by constructing a routing tree based on the alarm messages received. This topology map and trace back route to the attacker can be used to take effective counter measures.

CHAPTER IV

FINDINGS

The effectiveness of any intrusion detection algorithm can be measured based on its accuracy of detection. This means that an algorithm having lower false positives is better than one having higher false positives. In the case of wireless sensor networks, we must also consider the energy consumed by an algorithm. As energy is very scarce in sensor networks, an algorithm that consumes as little energy as possible is required. For example an intrusion detection algorithm might achieve high accuracy by consuming more energy. Such a detection mechanism would be a poor choice since wireless sensor networks should minimize power consumption.

4.1. Measuring PRR Distribution

Analyzing the effectiveness of the algorithm would require proper quantification of the distribution of packet reception rate (PRR) for a given range of link quality indicator (LQI) values. In order to obtain and analyze the distributions, we set up a network of 10 TelosB motes. Each mote was programmed using TinyOS to broadcast 200 packets of data at one second intervals to all other motes in the network.

To avoid collisions, only one mote designated as the sender, transmitted the data in the network at any particular point of time. When the other motes in the network receive this information they read the LQI value from the broadcasted packet and use this to calculate the average LQI over all the packets received. They also compute the number of packets they successfully received from the sender. After a node finished transmitting 300 packets, another node is designated as sender. This test was repeated a number of times by changing the locations of the motes to observe variation in LQI values.

When a sender completed transmitting data each mote was given a command to send the data back to the base station attached to a computer. The following results were observed from the experiments conducted. The PRR distribution for different LQI value ranges can be seen to follow Rayleigh distribution.

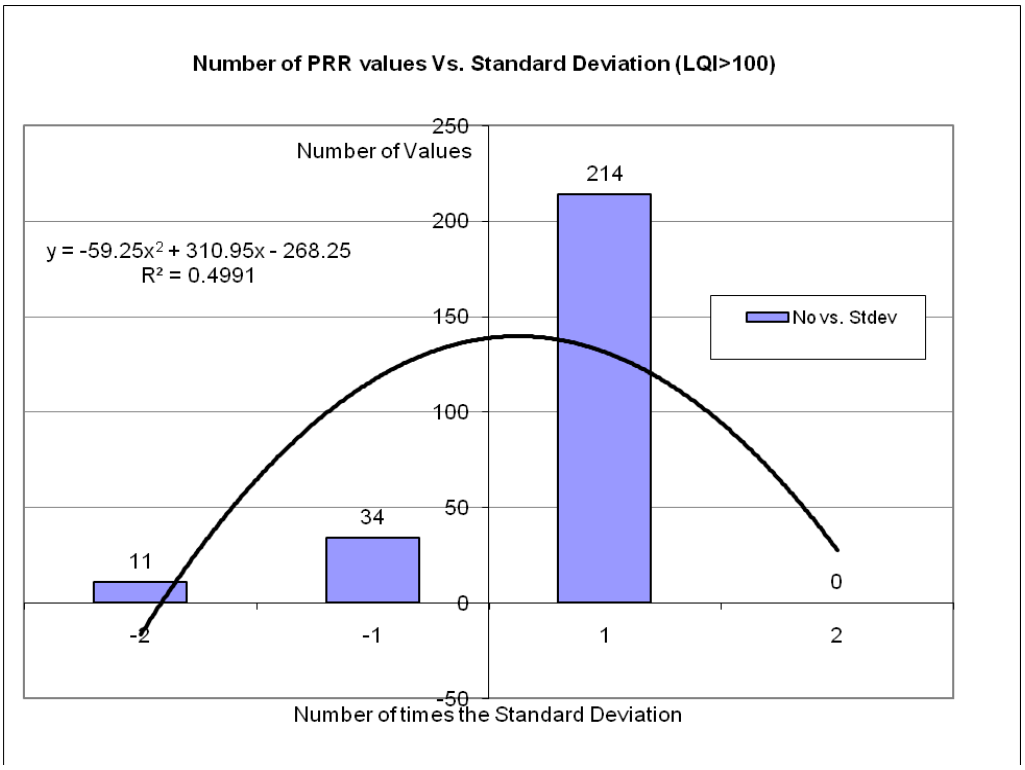

Figure 11: Distribution of PRR where LQI is greather than 100

Figure 11 above shows the distribution of packet reception rate for link quality indicator values greater than 100. When the LQI is greater than 100, the average PRR over the observed results was found to be 99.66. The standard deviation from this PRR was found to be 1.13. From these values the above graph for the distribution of PRR values was obtained.

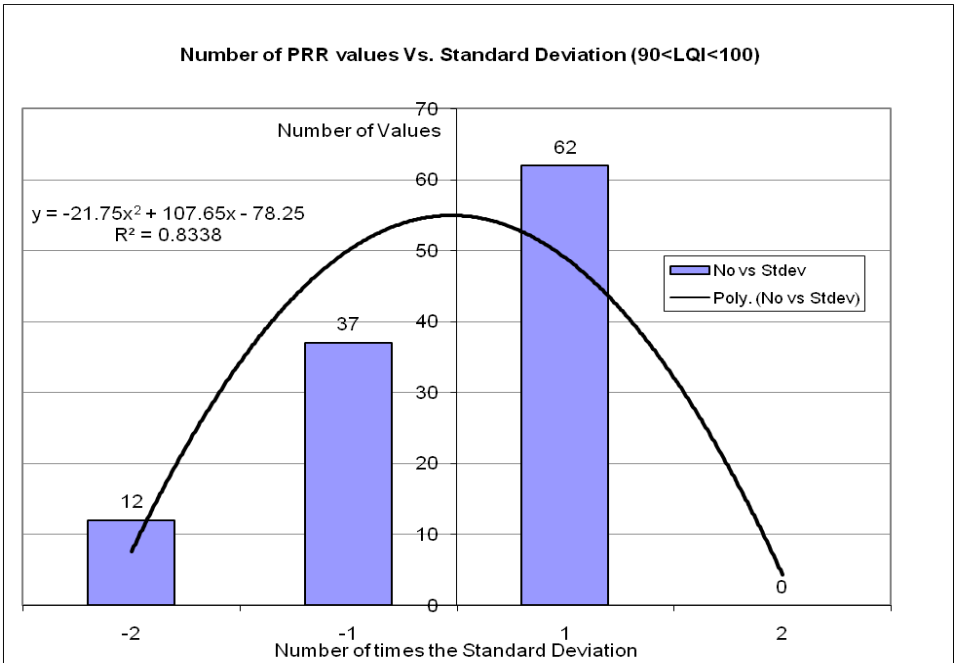

Figure 12: Distribution of PRR values where LQI is between 90 and 100

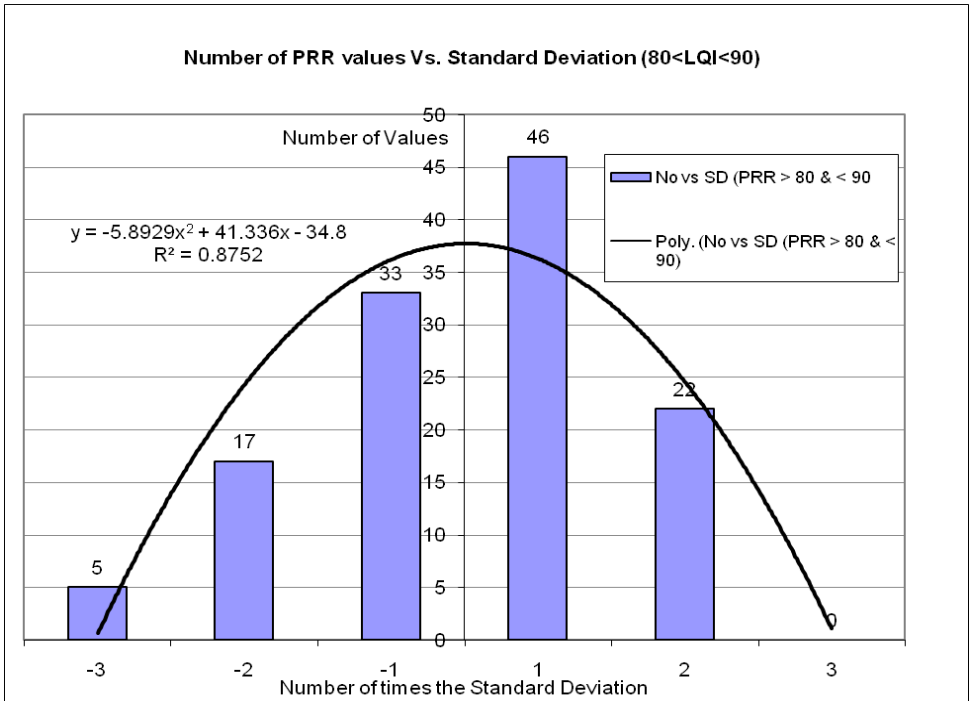

Figure 13: Distribution of PRR values where LQI is between 80 and 90

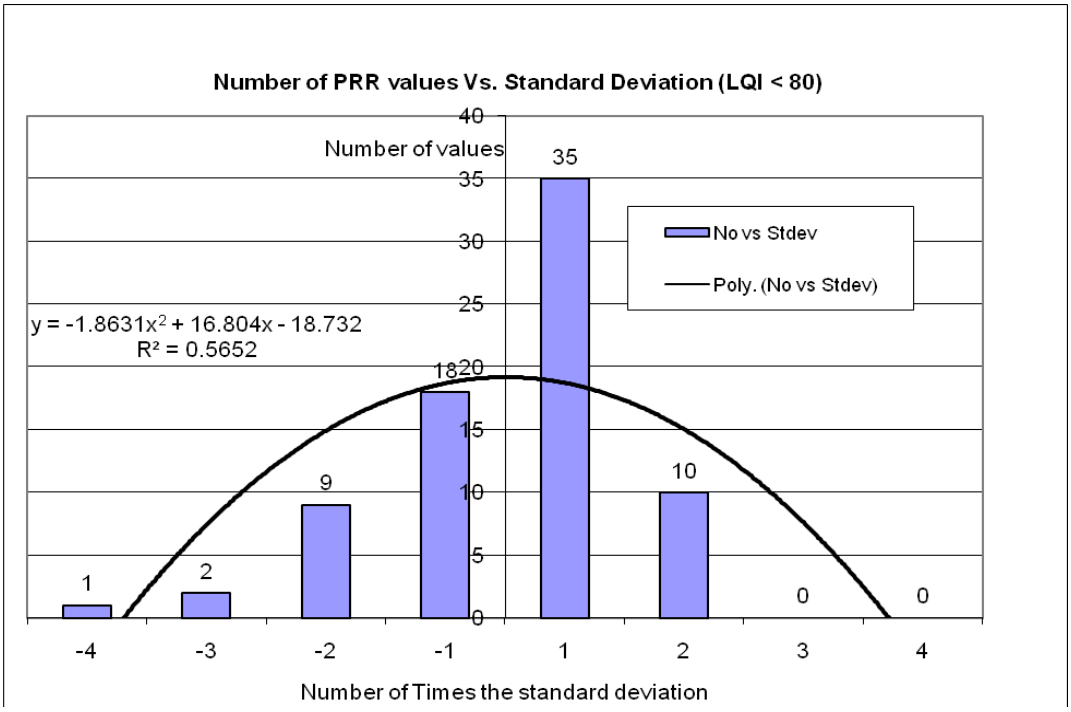

Figure 14: Distribution of PRR values where LQI is less than 80

Similar analysis was performed for LQI value ranges 90 to 100, 80 to 90 and less than 80. The resulting graphs are shown in figure 11, figure 12 and figure 13 respectively. The mean PRR values and the standard deviation of the PRR values for each range of LQI values are provided in table 1.

LQI Ranges	Mean PRR	Standard Deviation
> 100	99.66	1.14
> 90 & < 100	97.64	2.52
> 80 & < 90	93.16	3.77
< 80	78.72	9.26

Table 1: Mean PRR and Standard deviation for different LQI ranges

4.2. Experimental Setup

To analyze the effectiveness of the detection algorithm, we made use of both TOSSIM simulator and real motes. The programs were written in nesC for the TinyOS operating system. After the programs were verified to work properly in TOSSIM, the real network with TelosB motes was setup.

4.2.1. Network Topology

A network of five TelosB motes was setup to analyze the algorithm we developed. These motes were arranged as shown in figure 15. As described earlier, two motes were intermediate nodes for our algorithm. The remaining three motes were designated as the downstream node, node under investigation and the upstream node in the network. The positioning of the motes was carefully adjusted by taking into consideration, the required average link quality at each node.

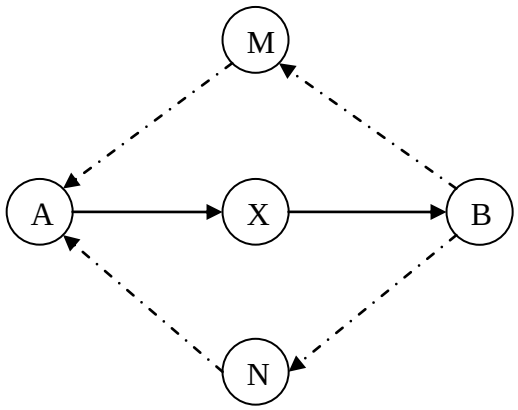

Figure 15: Network Topology

In figure 15, node A sends data to node X. Node X must forward the data to node B on path towards the base station. Here X is the node under suspicion. There are two other nodes M and N, which are placed so that they can communicate with both A and B. These are the intermediate nodes that can be used to relay acknowledgements from B to A. The solid arrows in the figure depict the path take by normal data packets. Dotted lines in the figure represent the path taken by acknowledgement messages in the network.

4.2.2. Network and Algorithm Parameters

Node A in the network was programmed to send data at 1 packet per second towards the base station. Node X was instructed to drop packets at different rates ranging from 5% to 30% of the packets received. The attack was simulated using this scenario.
As the data was sent at 1 packet every second, the value of n (time between ACK_MSG transmissions) was set at 100 seconds. This ensures that the average LQI value would give a fair estimate of the actual link quality. Each node was programmed with an initial maliciousness indicator (τ_a) of 0.1. Hence node A would have an initial maliciousness indicator (MI) of 0.1 on node X (the suspicious node). The maliciousness reduction constant used in our tests was 0.1.

Each node in the network was programmed with the values in Table-1. These values were used to store the mean packet reception rate and standard deviation for different LQI value ranges in the nodes. This table functions as a lookup table for the probability computed by the nodes, that a reported PRR is correct for observed LQI values. Moreover, the regression equation for finding the PRR with respect to a particular LQI was programmed into the nodes.

Once the real network was started, it was allowed to continue execution until 20 minutes. All the data logged by the motes was then collected and analyzed. With the same network, an attacker was not included and the experiment was repeated for 20 minutes. For each network configuration more than 20 such tests were conducted. As discussed

previously, threshold value must be carefully chosen to avoid false positives. For the results obtained from the experiments with different network configurations, we analyzed the increase in maliciousness indicator of the attacker. Different values for the threshold were analyzed and 4.5 were found to be a good threshold for MI in our network conditions. At this value, we found that we were achieving a balance between required accuracy of detection and time taken to detect an attacker. If a node was not detected as an attacker, i.e. did not cross the threshold even after 20 minutes, we assumed that our algorithm could not detect the attacker anymore.

To ensure that ACK_MSG transmissions by the intermediate nodes are not lost due to collisions or poor link quality in the network, we made use of ACK messages. As discussed earlier, a node must acknowledge the receipt of ACK_MSG transmission by sending an ACK message. Though this introduces some extra communication, it is a necessary precaution.

4.3. Results

4.3.1. Detection Accuracy

Figure 16 shows the accuracy of detection with respect to packet drop rate at node X at different LQI value ranges. It can be observed that the accuracy of detection increases as the amount of packets maliciously dropped by the attacker increases. This behavior can be attributed to the fact that it is easier to accurately detect an attacker when the

difference between the actual loss rate in the network and the amount of maliciously dropped packets is high.

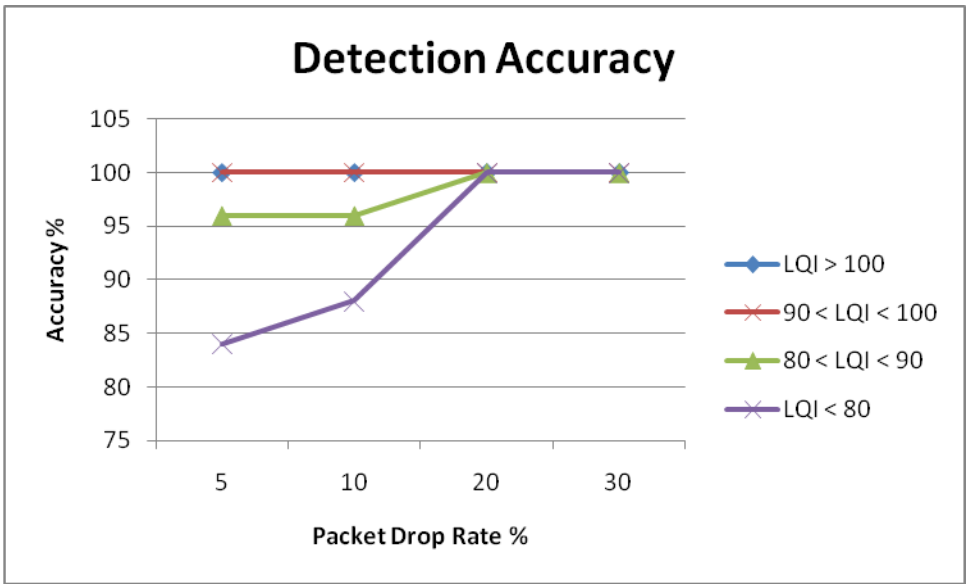

Figure 16: Graph showing the detection of accuracy for different LQI ranges

In the case where LQI values were greater than 100, the accuracy of detection was found to be 100%. Similarly for LQI values between 90 and 100, the accuracy of detection was 100% at all packet drop rates. When the LQI values observed were between 80 and 90, the accuracy of detection at low packet drop rates decreased in the network. However the accuracy reached 100% when the number of packets being dropped increased. This same behavior can be observed for LQI values below 80.

4.3.2. Undetected Rate

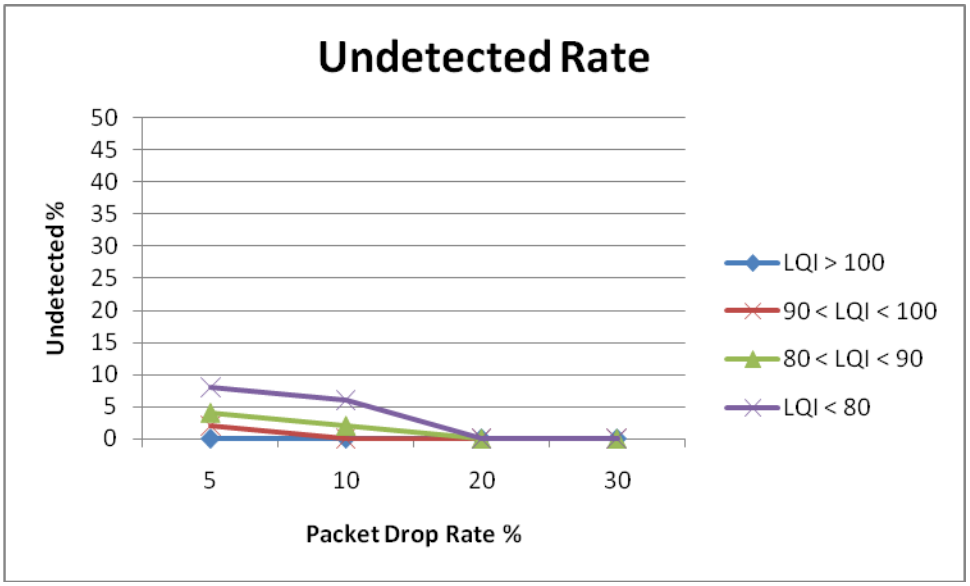

Figure 17: Graph showing the number of attackers not identified after 20 minutes

Figure 17 shows the rate of undetected attackers in the network. If a malicious node in the network was not identified within 20 minutes, it is said to have not been detected by our algorithm. Like the accuracy of detection, the rate of undetected attackers depends on the LQI values observed in the network and also the packet drop rate by the malicious node. The number of undetected attackers was found to decrease with the increase in packet drop rate. So an attacker that drops more packets has a higher chance being detected. Similarly the undetected rate in the network decreases with increase in observed LQI values. High link quality in the network aids in proper discrimination between maliciously dropped packets and normal packet loss.

From figure 17, we can observe that at high LQI values over 100, the number of undetected nodes is 0. As LQI values and the amount of packet drop rate decrease, the number of motes not detected in the network increases. For LQI values below 80, the highest amount of undetected nodes was observed at 8%. This rate decreased as the LQI values or packet drop rate increased.

4.3.3. Communication Overhead

The number of messages being transmitted in the network depends upon the data rate in the network and also the messages transmitted for enforcing various routing and cryptographic algorithms used in the network. In our case, we define communication overhead as the number of additional messages transmitted in the network for the proper functioning of our detection algorithm.

To get the relative communication overhead of our algorithm, two identical networks as described in section 4.2.2 were used. The first network was not running our detection algorithm. The number of messages transmitted in this network has been logged. Our detection algorithm was incorporated into the second network and the number of messages transmitted by each node in the network has been logged. As described earlier, to ensure proper reception of ACK_MSG sent by the intermediate nodes, we made use of ACK messages. Each node must send back a ACK message to the intermediate nodes sending the ACK_MSG. This is necessary for proper functioning of our algorithm. By

comparing the logs from the two networks, we can get an idea of the relative communication overhead due to our algorithm.

The ratio between the number of messages transmitted with and without the detection algorithm was calculated for different average LQI value ranges observed in the network. When the data rate in the network was 1 packet per second, with 100 second time interval between acknowledgement message transmissions and with average LQI value of 100, the average number of additional messages transmitted was found to be as shown in Figure 18 for each interval.

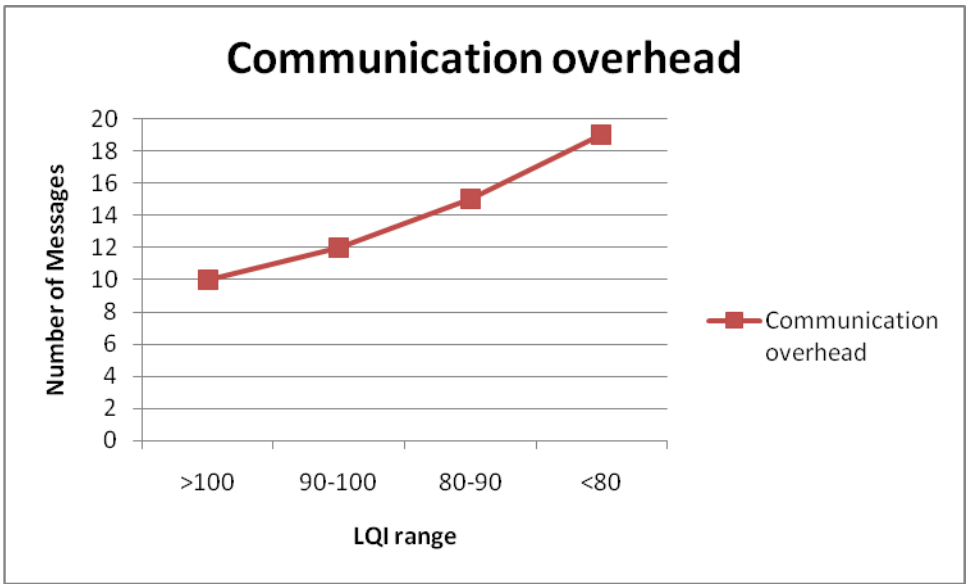

Figure 18: Communication overhead with respect to different LQI ranges

The number of additional messages for each interval of 100 seconds was computed and the average number of additional messages for a given LQI range was computed. Average number of messages represented in figure 18 is the ceil values of the average

computed. As there can't be any partial communications, we made use of the floor values.

From the values observed, we can make the following inferences. The number of messages transmitted increased with a decrease in link quality in the network. Increase in number of messages transmitted is due to the fact that ACK messages were used. When a node does not successfully acknowledge an ACK_MSG transmission, retransmissions occur and number of retransmissions in the network increases with decrease in poor link quality.

If there was no acknowledgement of ACK_MSG reception, we would see a constant communication overhead in the network. For a network transmitting at low data rates and having good average link quality among its nodes, we can eliminate the use of ACK messages. The probability that an ACK_MSG packet will be lost in such a network is very low due to these conditions. Hence elimination of the ACK messages results in constant communication overhead of 4 packets per *n* seconds, due to ACK_MSG transmissions, with no decrease in detection rate.

4.4. Comparison With Other Approaches

To show the merits of our algorithm we validate it with other selective forwarding attack detection algorithms. In this section we compare our algorithm to one described in [7].

The comparison is done on three fronts; accuracy, undetected rate and communication overhead. We also need to compare both the algorithms at different link qualities.

4.4.1. Accuracy of Detection

In [7], the authors have provided us the accuracy of detection for their algorithm. As they use alarm packets to detect maliciously dropped packets, we can use the alarm reliability they provided to get the accuracy of detection. The two approaches differ in detecting selective forwarding attacks. While [7] tries to detect an attack by identifying each packet being maliciously dropped, our approach tries to detect a selective forwarding attack by observing the data transmissions in an interval of time.

The authors of [7] have obtained their results by setting up a network of 400 motes. They use simulation at different channel error rates to obtain the accuracy of detection. They also make use of a transport layer retransmission policy with a default retry value of 5. This means that a packet will be retransmitted 5 times if the delivery fails. They mention that their retransmission algorithm is similar to PSFQ [7].

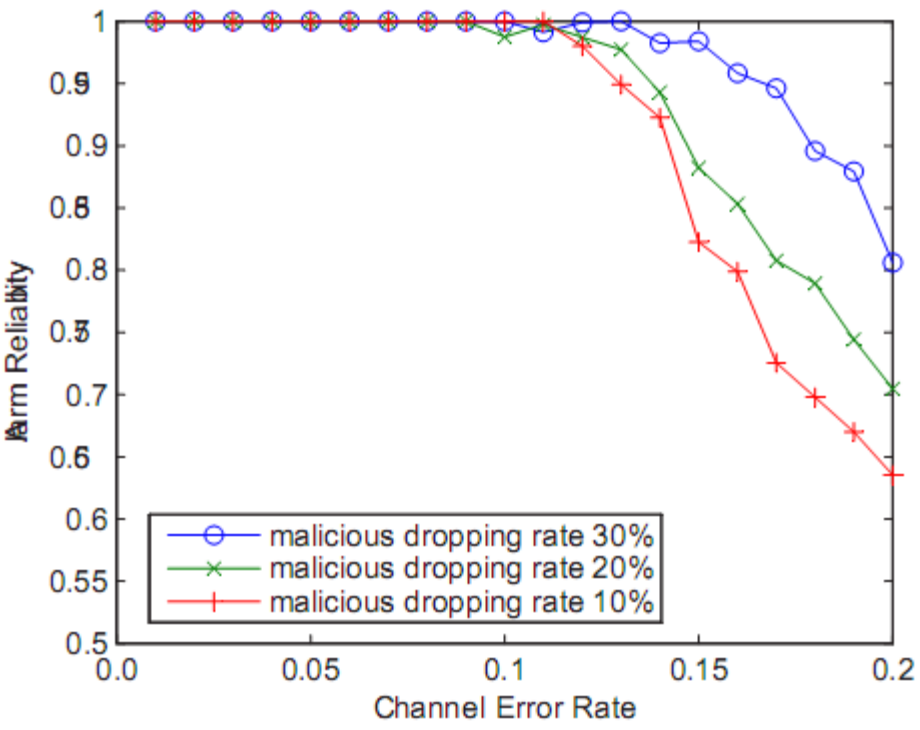

Figure 19: Alarm reliability figures from [7]

The results shown in figure 19 have been provided in [7]. From these results we can observe that the accuracy of detection decreases with respect to channel error rate in the network. Moreover, the accuracy decreases with decrease in maliciously dropped packets. The results for a malicious drop rate of 10%, 20% and 30% have been provided.

Comparing our results with those in figure 16, we can observe that the accuracy of detection is similar in both the algorithms. When channel error rate falls as low as 20% only 65% of the malicious nodes are detected in the network using the algorithm from [7]. This can correspond to the LQI range of less than 80. We can see that our algorithm works better when the channel error rate is high. Also, from the two figures, we can see

that detection accuracy is slightly higher in our approach when the number of maliciously dropped packets is lower.

4.4.2. Undetected Rate

The other metric we consider in comparing both the algorithms is the undetected rate. The results provided by the authors of [7] are shown in figure 17. A similar simulated network setup used to estimate the accuracy of detection was used to find the undetected attacker rate in the network.

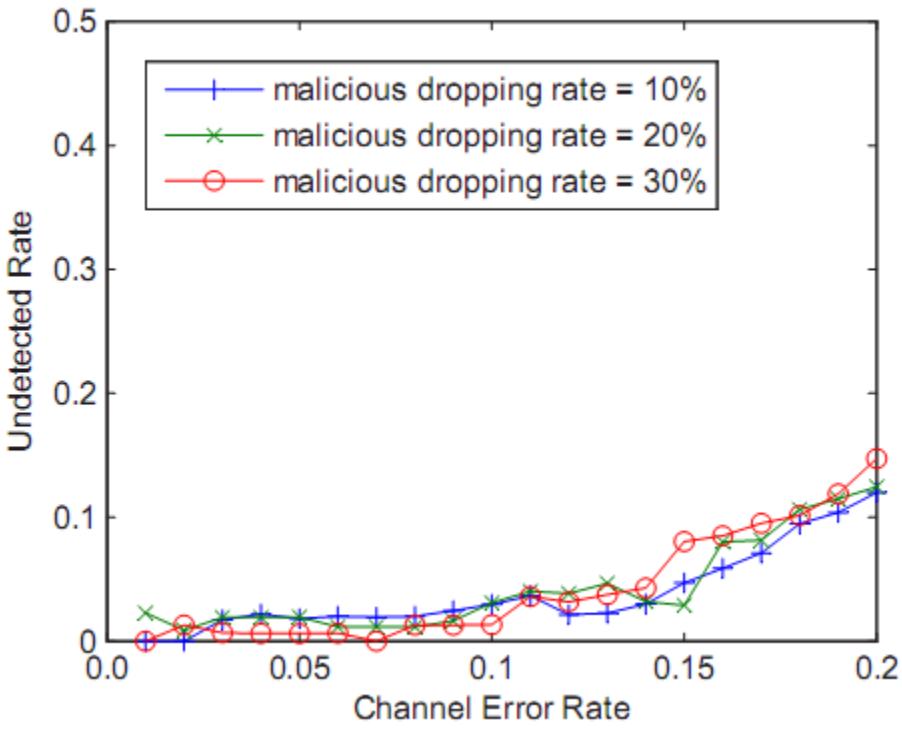

Figure 20: Undetected rate as provided in [7]

Figure 17 shows the undetected rate observed in the network setup of real motes using our algorithm. The undetected rate in the network is similar using both the algorithms. The only difference is when the link quality is very poor. Our algorithm is shown to perform better if the link quality is poor. As described earlier, a link quality of less than 80 can be seen as corresponding to 20 percent channel error rate shown in figure 20. By observing the values of undetected rate from the two figures, we observe that our algorithm has an edge when link quality is poor.

4.4.3. Communication Overhead

The main difference between our algorithm and the one presented in [7] is the communication overhead involved in enforcing the algorithms. The communication overhead for our algorithm has been provided in figure 18. From that figure we can observe that the number of additional messages transmitted in the network enforcing our algorithm depends on the link quality in the network. The higher the link quality the lower the number of additional messages transmitted in the network.

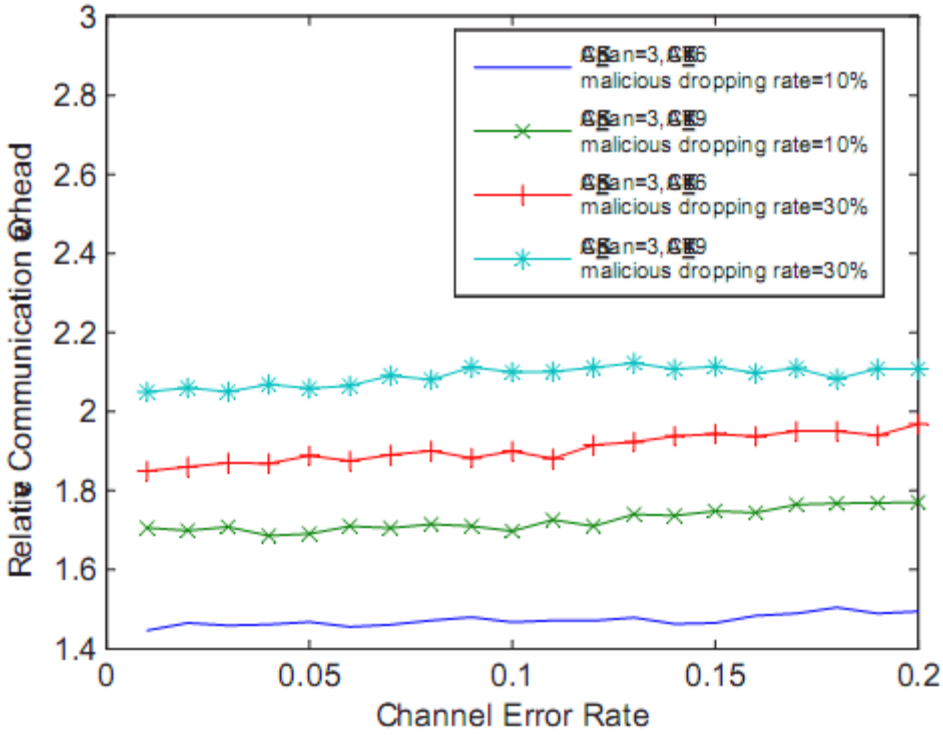

Figure 21: Relative communication provided in [7]

The communication overhead for the other algorithm can be seen in figure 21. Relative communication overhead is presented instead of the number of extra messages transmitted in the network. Relative communication overhead can be computed by comparing the number of messages transmitted for a given time in a normal network with the network enforcing the detection algorithm. The algorithm presented in [7] had the following parameters; ACK_TTL value of 3 hops and ACK_span of 3 hops. This means that the acknowledgement messages travel for 3 hops. ACK_span controls the number of times an acknowledgement message is generated in the network.

From figure 21 we can observe that the communication overhead in enforcing the other algorithm is at least 40% more than that observed in a normal network. For higher

malicious packet drop rates the communication overhead is even higher. It becomes more than 2 times the normal amount in some cases.

Comparing these results with our algorithm, shown in figure 18, we can observe that the communication overhead to our algorithm is considerably lower. This is due to the fact we make use of cumulative acknowledgements. Additional messages are transmitted only when the link quality is poor and ACK_MSG transmissions are lost. If we compute the relative communication overhead, our algorithm has a worst case measure of 20% of the packets. This number can be computed by taking into consideration that the data rate used in our network is 1 packet per second and time interval is 100 seconds. The number of additional messages transmitted at poor link qualities as shown in figure 18 is about 20. Hence, there is a communication overhead of 20%.

Clearly from the above results, we can observe that the communication overhead involved in our algorithm is considerably lower when compared to the other approach.

CHAPTER V

CONCLUSION

In this thesis, we presented an algorithm for effectively detecting selective forwarding attacks. We performed real network test runs using sensor motes and computed detection accuracy, undetected attacker rate and communication overhead of our detection algorithm. Finally we compared the efficiency of our algorithm with previous work done in the area. Based on our comparison we found that our detection algorithm slightly increases attacker detection accuracy in networks having low communication quality while not compromising the undetected attacker rate. The main improvement comes in the form of decrease in communication overhead. We found that our algorithm greatly decreases the communication cost involved in detecting an attacker. Other improvements include eliminating the need for a synchronized clock, which is difficult to implement in a wireless sensor network. All these improvements help to secure a wireless sensor network by not compromising longevity.

In the future, we would like to implement a Rayleigh curve fitting algorithm for the PRR distribution to make it more accurate. Moreover, we need to address the issue of time taken to detect a selective forwarding attack in the network.

REFERENCES

[1] Akyildiz I.F., Su W., Sankarasubramaniam Y., and Cayirci E., “Wireless sensor networks: a survey”, Computer Networks, Vol. 38, pages 393—422, 2002.

[2] Alan Mainwaring , David Culler , Joseph Polastre , Robert Szewczyk , John Anderson, “Wireless sensor networks for habitat monitoring”, Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, September 28, 2002, Atlanta, Georgia, USA

[3] C. Karlof and D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 03), pages 113-127, May 2003.

[4] W. Lee and S. J. Stolfo. “Data mining approaches for intrusion detection”. In Proceedings of the 7th USENIX Security Symposium, 1998.

[5] Soumya Banerjee, Crina Grosan, Ajith Abraham, "IDEAS: Intrusion Detection based on Emotional Ants for Sensors," *isda*, pp. 344-349, 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005.

[6] D.E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222-232, Feb., 1987.

[7] Bo Yu. and Bin Xiao, "Detecting selective forwarding attacks in wireless sensor networks", 20th International Parallel and Distributed Processing Symposium, 2006.

[8] TELOSB, www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf [last accessed - September 09, 2007].

[9] IEEE std. 802.15.4 - 2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs) http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf

[10] CC2420 http://www.ti.com/lit/gpn/cc2420 [last accessed - September 09, 2007].

[11] Zigbee specification, http://www.zigbee.org/en/spec_download/download_request.asp [last accessed - September 09, 2007].

[12] K. Srinivasan and P. Levis, "Rssi is under appreciated", In Proceedings of the Third ACM Workshop on Embedded Networked Sensors (EmNets 2006), May 2006.

[13] STARGATE, www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Stargate_Datasheet.pdf [last accessed - September 09, 2007].

[14] MICAz datasheet http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf [last accessed - September 09, 2007].

[15] Y. Yu, R. Govindan, D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks", Tech. Rep. UCLA/CSD-TR-01- 0023, Computer Science Department, University of California at Los Angeles, May 2001.

[16] B. Karp, H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks", Mobile Computing and Networking, 2000, pp. 243–254.

[17] A. Ephremides, J. Wieselthier and D. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling", Proceedings of the IEEE, 1987.

VITA

Venkata M. Mulpuru

Candidate for the Degree of

Master of Science

Thesis: DETECTING SELECTIVE FORWARDING ATTACKS IN WIRELESS 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 December, 2007.

Experience:
Worked as a Graduate Research Assistant under Dr. Johnson P. Thomas at Oklahoma State University, Stillwater, Oklahoma from August, 2005 to November, 2007. Worked as a Student Intern at Tata Consultancy Services (TCS), Hyderabad, India from January, 2005 to May, 2005.

Name: Venkata M. Mulpuru Date of Degree: May, 2008

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: DETECTING SELECTIVE FORWARDING ATTACKS IN WIRELESS SENSOR NETWORKS

Pages in Study: 54 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: Wireless sensor networks are being used in a wide variety of applications ranging from home automation to military surveillance. Security of a sensor network must be ensured for proper functioning of the many applications that depend on them. Due to the low computational capabilities and resource constrained nature of sensor nodes, sensor networks are prone to many attacks like selective forwarding. In this thesis we propose a novel and low power consuming approach to accurately detect selective forwarding attacks and trace back the attacker using an acknowledgement based scheme. We also make use of a probability based metric to increase the accuracy of detection and to reduce the undetected attacker rate in the sensor network.

Findings and Conclusions: We performed experiments on a network of real sensor motes and showed that our approach has higher accuracy of detection and lower communication overhead compared to previous work.

ADVISER'S APPROVAL: Dr. Johnson P. Thomas