A Novel Game Theoretic Framework for Security in Wireless Sensor Networks

by

AFRAND AGAH

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

To my husband

and

my parents.

ACKNOWLEDGEMENTS

I am deeply indebted to my supervising professor Dr. Sajal K. Das whose expertise, encouragement and guidance was integral throughout my studies in the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). His expertise, professionalism and devotion to the research and his students made it possible for me to finish my Ph.D. degree. Without his continuous encouragement, I would not have reached this far.

I would also like to thank Prof. Kalyan Basu, Dr. Manfred Huber, Dr. Behrooz Shirazi, Dr. Ramesh Yerraballi and Dr. Gergely Zaruba for serving in my Ph.D. dissertation committee. I am especially grateful to Prof. Basu and Prof. Huber for the helpful discussions and invaluable comments in improving the quality of my dissertation.

I thank all my colleagues in the CReWMaN Lab for their support, friendship and all the valuable discussions. In addition, I'd like to thank the department of Computer Science and Engineering at UTA and the National Science Foundation (Grant # IIS-0326505) for the financial support, without which I would not be able to finish my degree. Also I am thankful to Dr. Frank Lewis and the Automation and Robotics Research Institute (ARRI) at UTA for providing me the opportunity for conducting experiments on the sensor network testbed.

Last but definitely not the least, I express my deep gratitude to my husband, Mehran Asadi who immensely helped and inspired me during my undergraduate and graduate studies. I am also extremely grateful to my parents for their sacrifice and patience.

November 14, 2005

ABSTRACT

A Novel Game Theoretic Framework for Security in Wireless Sensor Networks

Publication No. _____

Afrand Agah, Ph.D.

The University of Texas at Arlington, 2005

Supervising Professor: Sajal K. Das

Due to severe resource limitations and often lack of centralized infrastructure, providing security in wireless sensor networks is a great challenge. Misbehavior due to malicious or faulty nodes can significantly degrade the performance of such networks. Therefore, countermeasures against denial of service (DoS) attacks and node misbehavior are essential requirements. We argue that the conventional view of security based on cryptography techniques is not sufficient for securing wireless sensor networks. In this dissertation, we investigate a novel framework by proposing three approaches for security enforcement in such networks that range from prevention of DoS attacks to secure routing. Prevention of DoS attacks focuses on the formal assessment of the properties of cooperation enforcement mechanisms used to detect and prevent malicious behavior of sensor nodes.

Our first proposed approach is called Utility based Dynamic Source Routing (UDSR). It is based on non-cooperative game theory, where players of the game are sensor nodes. Players can occasionally misbehave. In this game, we demonstrate that in order to reach equilibrium, where no rational player has any incentive to deviate and to maximize the

profit for the network (i.e., the least amount of false detections), a sensor network shall isolate those nodes that act maliciously. These nodes have the minimum amount of utility in the game. This approach provides an automatic method for the social mechanisms of reputation and cooperation.

Our second proposed approach is called Secure Auction based Routing (SAR). The assumption is that rational players always plan to maximize their profit over time. Here the key to solve this problem is when a node uses other nodes in the network to forward its own packets, it has to contribute to the network life (by forwarding other nodes packets) in order to be entitled to use them in the future. To enable such networks to keep functioning despite the presence of misbehaving nodes, we propose a mechanism such that nodes prefer to gain reputation in the network. Nodes willing to do so must compete against each other, where the competition is based on auction theory. A node's truthful bidding remains a dominant strategy and to have a secure routing protocol, malicious nodes who do not bid truthfully shall be isolated.

Our third proposed approach deals with detection of malicious nodes, based on repeated games. The benefit of this approach is the impact of a large group of players in the sense that the strategy chosen by a player does not only depend on one malicious node's perception of the game, but also on the group policy for all players. The strategy of a sensor node is to decide whether to cooperate with other nodes. This approach identifies non participating nodes and isolates them. We show that infinite repetition can be the key for obtaining equilibrium behavior, which could not be reached if the game were played once or for a known finite number of times. Implementation results on a sensor network testbed indicate that the repeated game based approach, conditioned on past histories of players, detects the malicious nodes more accurately.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Recent advances in micro-electro-mechanical systems (MEMS) technology and wireless communications have enabled the development of wireless sensor nodes that are small in size and able to communicate over short distances. Each sensor node consists of sensing, data processing and communication components [6].

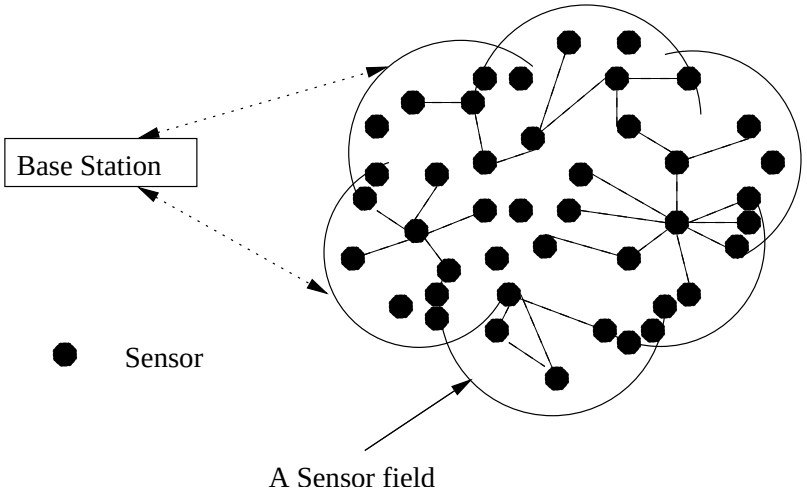

Figure 1.1. An architecture of Wireless Sensor Network.

A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the monitored environment or very close to it, as depicted in Figure 1.1. A sensor node in a sensor field detects the environmental data and the data is routed hop by hop through the nodes until it reaches the base station. Sensor nodes have the ability to locally carry out simple computations and transmit only the required and

partially processed data. Sensor nodes carry limited power sources (battery), generally irreplaceable, and hence sensor protocols must focus on power conservation.

Although many protocols and algorithms have been proposed for traditional wireless ad hoc networks, they are not well suited for the unique features and application requirements of wireless sensor networks. The differences between wireless sensor networks and ad hoc networks are outlined below [29]:

- The number of sensor nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad hoc network.
- Sensor nodes are densely deployed.
- Sensor nodes are prone to failure.
- The topology of a sensor network changes very frequently.
- Sensor nodes mainly use a broadcast communication paradigm whereas most ad hoc networks are based on point-to-point communications.
- Sensor nodes are limited in power, computational capacities and memory.
- Sensor nodes may not have global identification because of the large amount of overhead and large number of sensors.

1.1 Application Characteristics

Typically, sensor networks are meant to be deployed and then left physically unmaintained for a longer or shorter period, ranging from years to days. Depending on the scenario, reliability/fault tolerance requirements, remote maintainability, and lifetime of the sensor network may be different. Lifetime of individual sensors determines the lifetime of the entire network.

The requirement to be able to update the software running on the sensor networks is probably quite prevalent, but heavily influenced by the purpose of the sensor network and the environment it is deployed in. An application may cover an area where all nodes

can communicate with all other nodes using single hop communication. If this is not the case, the nodes need to employ multi-hop communication. This obviously influences the topologies that are needed. Related to this is the node density. If the phenomenon to be observed requires a high node density, the choice of network protocol need to take into account the increased risk of collisions, etc.

An important question for any sensor network application is whether the nodes are mobile, and expected to move during the lifetime of the network. Stationary topologies mean that the network can perform route investigations during initialization. Obviously, this does not mean that the application can then forget about route changes; because the electromagnetic field may change (e.g., as a consequence of external noise) and the application will still have to handle changes in the link reliabilities and performance. Finally, in the extreme case, a static routing scheme can be planned before deploying the nodes, but this must be considered an unwise approach unless in a very controlled environment. Changing topologies in most cases require that the application performs node discovery or at least is able to determine that a node is no longer able to send/receive data. Related to mobility is localization. Some applications require localization to work efficiently, which again can influence which platforms are suitable for the application.

In general, nodes will fail or communication will make it appear that a node has failed. The real question about node failure is to what extent the application can tolerate it. Some applications may accept node failures and simply try to work with as many nodes as possible. For other applications, it may be essential that certain key nodes never fail. Node discovery can mean rediscovering a node that was thought to have failed. Node discovery could also be the case for networks where some nodes leave the network.

For some monitoring applications, we may wish to collect data very frequently. For example, a timestamped notice each time a cars passes a point on a busy highway. In other cases, we may only wish for the number of cars that passed during an hour,

collected each hour. For some applications we may collect data very infrequently, but rather get noticed when, e.g., a person enters a room, because we want to take action based on this information.

1.2 Platform Characteristics

A common feature for wireless sensor nodes is that they are battery operated (sometimes solar or otherwise powered) and communicate wirelessly using radio, infrared lights, or other communication techniques. Having enough energy (battery power) is a primary concern in the design of a sensor network. The energy is spent on the sensor when it measures sensed data, on the micro controller unit (MCU) when it is running, on the radio when it is listening, receiving or sending, etc. If each device is expected to deliver a number of samples of sensed data, store them, and participate in communicating both its own samples and also samples from other nodes, the energy initially present in the battery is expected to be relatively high. In general, successfully transmitting a bit to another node is orders of magnitude more expensive than doing a calculation on the MCU [23]. Power control, that is controlling what the device does and only enabling the absolutely needed components, is an important task for the software (or hardware) running on the energy constrained sensor nodes. However, redundancy may be used instead of reliability and fault tolerance for some deployments. In other words, deploying a lot of nodes may simply be cheaper than improving the reliability of the hardware and software. Because the cost of nodes influences the number of nodes that can be deployed, keeping the nodes low cost is an objective for designing a sensor node.

1.2.1 Mica Nodes

Probably the most prevalent platform is the Mica family of platforms from UC Berkeley [23]. The Mica family originally developed from the COTS (Common Off The

Figure 1.2. Mica2 Mote.

Shel) prototype that was meant to demonstrate the communication capabilities in large scale of a node built with a radio. The Mica platforms include the Mica platform and the Mica2 platform. They are both based on 8 bit CPUs from the company Atmel, and include analog to digital converters (ADC), serial ports, an EEPROM, some flash RAM, a small amount of RAM, an expansion connector. Figure 1.2 depicts a Mica2 node with no external antenna.

1.3 Motivation of This Dissertation

As mentioned wireless sensor networks have very limited battery power, memory, computation and communication capacity, high node density, and easy node failure. These characteristics pose significant challenges in terms of the security of WSNs, and also render ineffective the applicability of existing methodologies of securing (ad hoc) networks.

A security breach can happen in a wireless sensor network not only while generating information but also while relaying to the end-user. A sensor network must be able to securely sense the physical environment, collectively process the sensed data, and communicate among nodes. Performing all of these relies on the trust among the nodes, which can be abused by adversaries to carry out security breaches. As sensor nodes are envisioned to be low-cost, it is infeasible for manufactures to make them tamper-resistant,

Table 1.1. DoS attacks in sensor networks and defense strategies [29]

DoS attacks	Defense strategy
Radio interference	Use spread-spectrum
Physical tampering	make nodes tamper-resistant
Denying channel	Use error correction code
Black holes	Multiple routing paths
Misdirection	Source authorization
Flooding	Limit the connections

so an adversary can insert faulty data into the network. Since internal adversarial nodes have access to valid cryptographic keys, cryptographic and authentication mechanisms alone can not be used to enforce security [29].

Nodes of a sensor network can not be trusted for the correct execution of critical network functions. Misbehavior of nodes may range from simple selfishness or lack of collaboration due to the need for power saving, to active attacks aiming at Denial of Service (DoS) and subversion of traffic. A sensor network without sufficient protection from DoS attacks may not be deployable in many areas. There are two types of DoS attacks:

- *Passive attacks*: selfish nodes use the network but do not cooperate, saving battery life for their own communications; they do not intend to directly damage other nodes.
- *Active attacks*: malicious nodes damage other nodes by causing network outage by partitioning, while saving battery life is not a priority.

DoS attacks can happen in multiple sensor network protocol layers. Table 1.1 depicts the typical DoS attacks and the corresponding defense strategies [29].

Clearly, it is crucial that the security of sensor networks be monitored and diagnosed, if necessary, to ensure correct behavior. This is challenging in an environment where the network is designed to be flexible. A malicious node can misrepresent its

identity in the network and issue route error messages to misdirect the path or drop incoming packets. As normal usage and communication patterns needed for anomaly-based intrusion detection are typically not known in advance in sensor networks [29], and also the presence of intruders can make it difficult to determine these values, we believe that a sensor network should be more intrusion-tolerant than intrusion detective. Being intrusion-tolerant means that a malicious node can only compromise a very small number of nodes in its vicinity, rather than causing widespread damage in sensor networks.

1.4 Contributions of This Dissertation

In this work we discuss security of sensor networks. Our objective in this work is to define a new set of security requirement with appropriate mechanisms that can be adapted to an infrastructure-less environment.Game theory is a branch of mathematics that studies the interactions of multiple independent decision makers that try to fulfill their own objectives. In recent years, game theoretic research on networks has emerged. In sensor networks, the maliciousness of nodes has more drastic consequences than in traditional networks because the network relies on the cooperation of the nodes. Game theory provides a good theoretical framework to analyze this issue. We model the interaction between a sensor node and the rest of the network as a game. Game theory deals with multi-person decision making situations. The basic assumption is that the decision makers pursue some well defined objectives and take into account their knowledge or expectations of other decision makers' behavior.

Cooperation of a node is its willingness to perform networking functions for the benefit of other nodes. However, non cooperation creates an energetic cost that can lead to a selfish behavior, especially in a battery powered environment such as wireless sensor networks. Indeed, there is no reason to assume that nodes will participate in the network operation. The main objective of this work is to define a framework that (i) encourages

sensor nodes to participate in network operations, (ii) identifies non participating nodes, and (iii) isolates such nodes. We propose three different approaches based on game theory to prevent DoS attacks. In a DoS attack, the attacker's objective is to make target destinations inaccessible by legitimate users. There is very little work done on the prevention of DoS attacks for sensor networks. Attempts to add DoS resistance to existing protocols often focus on cryptographic authentication mechanism. Aside from the limited resources that make digital signature schemes impractical, authentication in sensor networks poses serious complications. It is difficult to establish trust and identity in large-scale sensor network deployments. Adding security afterward often fails in typical sensor networks. Thus, design-time consideration of security offers the most effective defense against DoS attacks [29].

To enforce of security in wireless sensor networks, we will use different non-cooperative game theoretical frameworks, such as two person, strategic and repeated games as depicted in Figure 1.3. In the first approach, which is called Secure Auction based Routing (SAR), we propose a secure sensor network routing protocol based on an auction theory, which isolates malicious nodes. Nodes willing to participate in forwarding incoming packets and gaining reputation in the network, must compete against each other by participating in an auction. The amount of bid each node offers is its utility value; and the price that a winner of a bid pays is a reduction of its original battery power. Node's truthful bidding remains a dominant strategy and so to have a secure routing protocol, malicious nodes who do not bid truthfully, must be isolated over time.

In the second approach, which is called Utility based Dynamic Source Routing (UDSR), a game theory based mechanism to prevent DoS attacks is introduced, where a game is between an attacker and the sensor network. In this game each player maximizes its own payoff. forwarding messages, or (ii) issuing route error messages to a normal node, thus misdirecting the path. The network also intends to detect attacks correctly. The

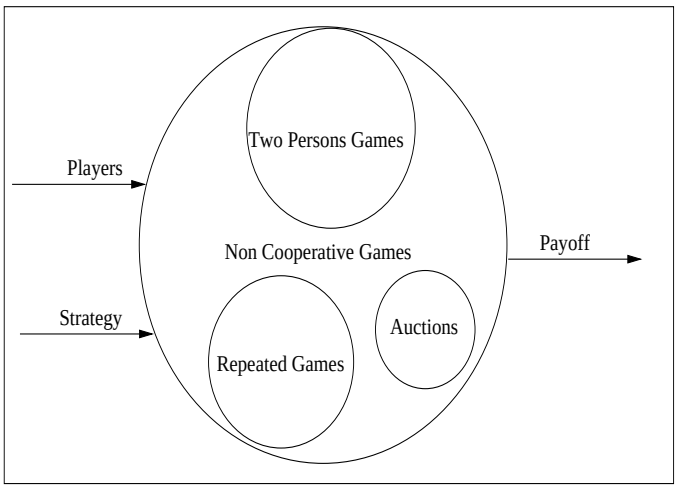

Figure 1.3. Game Theory Framework.

fewer miss-detection it has, the more payoff it gains. Since the sensor network must defend nodes from such intrusions, we formulate the attack-defense game as a non-cooperative, non zero-sum, two-player game. This game achieves Nash equilibrium, thus leading to a defense strategy for the network. In order to choose the most reliable route, two different schemes are proposed. The first scheme includes the total utility of each route in data packets, while the second scheme incorporates a watch-list, where misbehavior results in bad reputation and propagates to other nodes too. Simulation results indicate that the proposed approach keeps the number of dropped packets constant irrespective of the network size. Also, after recognizing and labeling some nodes as malicious ones, as bad behavior will propagate throughout the network, other nodes in the network can ignore these malicious nodes for their future packet forwarding requests.

The third approach focuses on the prevention of passive DoS attacks at the routing layer in wireless sensor networks and is formulated as a repeated game between an intrusion detector and nodes of a sensor network, where some of these nodes act normally and some act maliciously. We propose a scheme to enforce cooperation among nodes and punishment for non cooperative behavior. We assume that the rational users

optimize their profits over time. Intrusion detector residing at the sink keeps track of other nodes' collaboration by monitoring them. If performances are lower than some trigger thresholds, it probably means that some nodes act maliciously by deviation. An intrusion detector rates other nodes, which is known as subjective reputation and the positive rating accumulates for each node as it gets rewarded.

1.5 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we present the security requirements that are specific to wireless sensor networks. We show different kinds of attacks. Further, we introduce secure routing and different mechanism for preventing denial of service attacks. We also illustrate the building blocks for sensor network security available in the literature and discuss their properties. Chapter 3 provides a basic introduction to game theory as relevant to this work. In Chapter 4, we propose our game theoretic approach which is based on auction theory. In Chapter 5, we propose an approach, which is based on a non-cooperative non-zero sum game. In Chapter 6, we develop an approach based on repeated game theory that is used to model the network and the interaction between the users. In Chapter 7, we provide both numerical and analytical validation of the proposed approaches. A simulation-based evaluation is carried out to provide significant insight into the basic properties of our schemes. Chapter 8 concludes the dissertation, and discusses the future research directions to extend this work.

CHAPTER 2

RELATED WORK

The unique characteristics of sensor networks limit the applicability of traditional security measures. Since sensor nodes have limited power sources, limited local memory and calculation capacity, they are not able to store long-sized keys or run complex cryptology algorithms. Usually sensor nodes are densely deployed, and may not have a global identification number because of the large amount of overhead. They may also fail due to lack of power or physical damage. The dynamic nature of sensor networks' topology is typically due to node failure or node insertion instead of node mobility since most sensor networks applications do not assume a highly mobile characteristic [29].

In this chapter, we will discuss the existing work in the following areas: In Section 2.1, goals and challenges for security are presented. In Section 2.2, work on secure routing are presented. Section 2.3, talks about existing approaches for prevention of Denial of Service attack.

2.1 Security Goals

When dealing with security, one is faced with achieving some or all of the following goals [29]:

- *Availability:* network assets are available to authorized parties when needed and the sensor network should ensure the survivability of network services despite denial of a service (DoS) attack. To ensure the availability of message protection, the sensor network should also protect its resources to minimize energy consumption [29].

- *Authenticity:* an adversary might easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originates from a trusted source.
- *Confidentiality:* a confidential message is resistant to revealing its meaning to an eavesdropper. Confidentiality should be provided by keys with as small a scope as possible, to discourage a single break from compromising a large portion of the network.
- *Freshness:* it implies that the data is recent and ensures no adversary replayed old messages.
- *Data Integrity:* it ensures that the received data is not altered in transit by an adversary.
- *Scalability:* sensor networks can not utilize a keying scheme that has poor scaling properties in terms of energy cost or latency. In general, the number of neighbors and the distance or power required to send messages from one node to another will not be known in advance.

There exists a conflicting interest between minimizing resource consumption of sensor nodes and maximizing security performance. The resource in this context includes energy as well as computational resources like memory. The capabilities and constraints of sensor nodes will seriously influence the type of security mechanisms that can be hosted on a sensor node platform. Energy is perhaps the greatest constraint on sensor nodes' capabilities. The extra power which will be used by sensor nodes can be due to several security functions, such as encryption, decryption, signing data or key storage. Tamper protection adds costs to each node. When designing the sensor network security architecture, we should assume that one or more sensor nodes within the network may be compromised. Due to the lack of tamper protection available to sensor nodes, a sufficiently capable adversary can extract compromising cryptographic information from

a sensor node. Tamper detection technologies can provide indication that tampering has occurred but have limited value in long-term unattended operation [6].

The ad hoc networking topology renders a sensor network susceptible to link attacks ranging from passive eavesdropping to active interfering. Unlike fixed hardwired networks with physical defense at firewalls and gateways, attacks on sensor networks can come from all directions and target any node. Since it is difficult to track down a particular mobile node in a large-scale sensor network, attacks from a compromised node are more dangerous and much harder to detect. All this indicates that any node must be prepared to operate in a mode that trusts no peer. New nodes may be added or current nodes may die, thus a sensor network has a dynamic routing structure. Frequent routing changes can mean that the intermediate nodes processing data for an end-to-end session can change. Also, since many security services will be provided on a hop-by-hop basis, cryptographic key establishment will occur with local neighbors in the routing topology. If the routing changes, the set of local neighbors may change and thus cryptographic key establishment may need to occur again. Considering a large number of nodes in a typical sensor network, it is not practical to adopt centralized security measures. Instead, distributed security algorithms should be adopted. Introducing any central entity into security solution may cause fatal attack on the entire network once the centralized entity is compromised. Generally, decision making in the sensor network is decentralized and many security algorithms rely on the cooperation of all nodes or partial nodes. The nature of ad hoc networking requires limited pre-configuration in order to support a flexible and easily deployable network. This constraint limits the amount and type of cryptographic schemes that should be necessary to deploy a secure sensor network. The sensor nodes may be unattended for long periods of time. For example, remote reconnaissance missions behind enemy lines may not have any physical contact with friendly forces once deployed. Although they may be managed remotely, in general sensor nodes are not in

physical contact with ground troops once deployed. This makes it impossible for physical detection of tampering (i.e., through tamper seals) and physical maintenance (e.g., battery replacement). Other maintenance functions are possible (e.g., software updates, key updates) but must be done remotely. The amount of time that a sensor is left unattended increases the likelihood that an adversary has compromised its key material [29].

2.2 Secure Routing

There are many new routing protocols proposed for general ad hoc networks. Among those routing protocols, the Ad hoc On-demand Distance Vector protocol (AODV) and Dynamic Source Routing (DSR) have demonstrated good performances [32]. But most of these schemes are not well suited to the typical sensor network model due to their assumption of high power, large memory and small scale.

Traditional ad hoc routing schemes have serious security limitations when used for sensor networks. There are six main types of routing attacks [29]: (i) *falsify route sequence numbers*: meaning AODV maintains routes by assigning monotonically increasing sequence numbers to each packet. When those numbers are changed, packets can be mistakenly considered to be misrouted or lost, (ii) *modify hop count field of the header*: AODV uses the hop count field in the route discovery message to determine a shortest path. A malicious node can attract routes towards itself by resetting the hop count field to zero, (iii) *modify the source route field*: DSR is a routing protocol which explicitly states routes in data packets. These routes lack integrity checks and hence a simple DoS attack can be launched in DSR by altering the source routes in packet headers, (iv) *spoofing*: spoofing occurs when a node misrepresents its identity in the network, such as by altering its IP address in outgoing packets, it can produce loop paths, (v) *falsify route error message*: an attacker can issue route error messages to a normal node to indicate

a broken link and thus misdirect the path, (vi) *corrupting routing table*: an attacker can delete, alter or inject the information in routing tables so that the path is messed up.

There are two kinds of attacks toward sensor networks routing protocols [40]:

- *External attacks:* replaying old routing information, injecting erroneous routing information are some examples of this kind of attack. Attackers can partition a network or introduce excessive traffic load into the network.
- *Internal attacks:* internal compromised nodes may send malicious routing information to other nodes.

So every sensor network routing security protocol must satisfy the following requirements [29]:

- Unauthorized nodes should be isolated during route discovery procedure.
- The network topology should not be revealed to adversaries.
- Paths should be immune to being misdirected from the shortest path by an attacker.
- Routing discovery messages can not be spoofed.
- Fabricated routing messages should be identified.
- Routing messages can not be altered in transmit by unauthenticated nodes.

Perrig, et al.[47], addressed secure communication in resource-constrained sensor networks by introducing two low-level secure building blocks. The Security Protocols for Sensor Networks (SPINS), consists of Sensor Network Encryption Protocol (SNEP) and μTESLA. The SNEP protocol has low communication overhead (only 8 extra bytes per message), providing baseline security primitives like data confidentiality, two-party data authentication, reply protection and message freshness. It achieves semantic security, i.e., the same message is encrypted differently each time, thus preventing eavesdroppers from inferring the content from the encrypted message. The μTESLA protocol [46] uses a symmetric key mechanism. To generate a one-way key chain, the sender chooses the last key randomly and generates the remaining keys by successively applying a one-

way function. The protocol discloses the key once per time interval (rather than one key per packet), and restricts the number of authenticated senders. To bootstrap, each receiver needs one authentication key of the one-way function key chain. The periodic key disclosure of μTESLA ensures that compromising a single sensor does not reveal the keys of all the sensors in the network. Consequently, the routing model is fairly limited: route discovery depends on the detection of authenticated beacons broadcast by the base station. Also, node to node communication necessitates authentication via the base station. The advantages of the SPINS protocol are: (i) feasibility of security with very limited computing resources and (ii) symmetric cryptography. The disadvantages are: route discovery depends on the detection of authenticated beacons by the base station, and requires node to node authentication via the base station.

The INtrusion tolerant routing protocol for wireless SEnsor NetworkS (INSENS) proposed in [21] does not rely on detecting intrusions, but rather tolerates intrusions by bypassing the malicious nodes. An important property of this protocol is that while a malicious node may be able to compromise a small number of nodes in its vicinity. It constructs forwarding tables at each node, minimizes computation, communication, storage and bandwidth requirements at the sensor node at the expense of increased computation, communication, storage and bandwidth requirements at the base station. The advantage of INSENS is that a malicious node can not cause widespread damage in the network. And the disadvantages are: (i) multiple disjoint paths are built to bypass a failed node, and (ii) energy consumption and packet collisions are increased because data are sent along multiple paths, irrespective of whether there is a node failure or not.

The Pebblenet protocol [8] adopts a cluster-based sensor network architecture for data forwarding. It uses a global unique key and a hash function to generate session keys in each update round. One of the cluster heads with higher power is chosen to become the key distribution center in each re-keying phase. The advantage of this protocol is that

network wide keys for encryption information are very good in terms of storage requirements and ease of use. The disadvantage is that once a node is compromised, forward secrecy is broken and therefore tamper-resistance becomes crucial. The key management server stores not only its own key pair, but also the public keys of all the nodes in the network. The difficulties include the storage requirement exerted on the servers which must potentially be specialized nodes in the network, and the overhead in signing and verifying routing messages in terms of both computation and communication. One can summarize the disadvantages as follows: (i) compromise of a single node undermines the security of the entire network, (ii) key management is a bottleneck, and (iii) there is too much communication and computation overhead.

The Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks (CONFIDANT) protocol [13, 12], aims at detecting and isolating misbehaving node and thus making it unattractive to deny cooperation. It consists of four components: (i) monitor, (ii) reputation system, (iii) path manager, and (iv) trust manager. In this protocol, a neighborhood watch is proposed and nodes look for any malicious behavior. As a component within each node, the monitor registers these deviations from normal behavior. The reputation system provides a quality rating of participants of transactions. The path manager ranks paths according to a security metric. A trust manager manages trust between nodes to determine the trustworthiness of paths. It is obvious that this approach is not applicable to sensor networks due to their limited memory. The CONFIDANT protocol assures routing security and fairness targeted to mobile ad hoc networks. However, it suffers from DoS attacks performed using the security mechanism itself. Indeed, malicious nodes are not prevented from distributing bogus information about other nodes' behavior. Even though it minimizes the effect of misbehaving or selfish nodes on routing through punishment and reporting, the approach is vulnerable to blackmailers [33]. The advantages of CONFIDANT protocol are: (i) nodes that do not forward will be punished, and (ii) it

Table 2.1. Classification of Secure Routing protocols for Sensor Networks

Protocol	Advantages	Disadvantages
SPINS	(i) Feasibility of security with very limited computing resources. (ii) Symmetric cryptography.	(i) Route discovery depends on detection of authenticated beacons by the base station. (ii) Requires node to node authentication via the base station.
INSENS	A malicious node can not cause widespread damage in the network.	(i) Multiple disjoint paths are built to bypass a failed node. (ii) Increase in energy consumption and packet collisions, because data is sent along multiple paths, irrespective of node failure.
Pebblenet	Having network wide keys for encryption information is very good in terms of storage requirements and ease of use.	(i) Suffers from obvious security disadvantage that compromise of a single node undermines the security of the entire networks. (ii) Key management is a bottleneck. iii) Too much communication and computation overhead.
CONFIDANT	(i) Nodes that do not forward will be punished. (ii) Avoids possible bad routes.	(i) Eavesdropping not addressed. (ii) Nodes in black list are ignored. (iii) Friend making is not well established.

avoids possible bad routes. The disadvantages are: (a) eavesdropping is not addressed, (b) nodes in black list are ignored, and (c) friend making is not well established.

Table 2.1 summarizes the classification of the described protocols.

We should also use a localized trust model instead of centralized security management. Centralized trust management is difficult and expensive. Besides, a sensor node typically cares about the trustworthiness of their immediate neighbors most due to the broadcast nature and the inherent local interactions of wireless transmissions. The node has to rely on its neighboring nodes for packet forwarding, routing and other network resource access. Therefore, a localized trust model is more appropriate for sensor network

routing protocol design. In the localized trust model, a locally trusted entity is globally accepted and a locally distrusted entity is regarded untrustworthy anywhere. The research presented in this dissertation is based on game theory. Here multiple players are considered, and the strategy selection phase is driven by node perception of the game. The three approaches that we are proposing in this work have several advantages. There is no need to key establishment, or storing long sized keys. There is also no need to include a trustworthy third party in every node to node communication. The economical modeling of security enforcement of the proposed approaches does not require a monitor and rating system at each individual sensor node, which again saves its memory and battery power. The benefit from using game theory lies in the ability of this method to seize the dynamics of a large group of players, and that the strategy chosen by a player does not only depend on a self-interested perception of the game but also takes into account a group policy of all the players. The proposed framework describes strategy of a self-interested node that has to make the decision whether to cooperate with the rest of the network. By adopting a more realistic assumption that takes into account observation of the node's behavior, the game theoretic framework proves the superiority of this framework.

2.3 Prevention of Denial of Service Attack

One simple form of DoS attacks is vulnerability by arbitrarily neglecting to route some messages. A subverted or malicious node can still participate in lower-level protocols, and may even acknowledge reception of data to the sender, but it drops messages on a random or arbitrary basis. Such a node is neglectful. The dynamic source routing (DSR) protocol is susceptible to this attack. Because the network caches routes, communications from a region may all use the same route to a destination. And a malicious node can degrade or block traffic from a region to a base station [59].

A necessary operation to overcome DoS attacks is to identify and circumvent the misbehaving nodes [37]. This watchdog scheme attempts to achieve this purpose through the use of watchdog and path-rate concepts. Every node implements a watchdog that constantly monitors the packet forwarding activities of its neighbors and a path-rater rates the transmission reliability of all alternative routes to a particular destination node. The disadvantages of this scheme are that it is only practical for source routing protocols instead of any general routing protocol, and collusion between malicious nodes remains an unsolved problem [29].

In the Rating scheme [38, 40, 41], the neighbors of any single node collaborate in rating the node, according to how well the node executes the functions requested from it. It strikes a resonant chord on the importance of making selfishness pay. Selfishness is different from maliciousness in the sense that selfishness only aims at saving resources for the node itself by refusing to perform any function requested by the others, such as packet forwarding, and not at disrupting the flow of information in the network by intension. The disadvantages of this approach are: (i) how an evaluating node is able to evaluate the result of a function executed by the evaluated node, (ii) an evaluated node may be able to cheat easily, and (iii) the result of the function may require significant overhead to be communicated to the evaluating node.

The Virtual currency scheme [9, 15] introduces a type of selfish nodes that are called *nuglets*. To insulate a node's nuglets from illegal manipulation, a tamper-resistant security module storing all the relevant IDs, nuglet counter and cryptographic materials is compulsory. Each packet is loaded with nuglets by the source and each forwarding host takes out nuglets for its forwarding services. The disadvantages of this schemes are that : (i) malicious flooding of the network can not be prevented, (ii) intermediate nodes are able to take out more nuglets than they are supposed to, and (iii) overhead.

Route DoS Prevention attempts to prevent DoS in the routing layer by cooperation of multiple nodes [12]. It incorporates a mechanism to assure routing security, fairness and robustness targeted to mobile ad hoc networks. The disadvantage of this approach is that misbehaving nodes are not prevented from distributing bogus information about other nodes' behavior and legitimate nodes can be classified as misbehaving nodes [29].

Since a large fraction of cooperation enforcement schemes are based on principles akin to decision making and economic modeling, game theory offers a natural tool that emerged to be suitable for validating such mechanisms.

2.4 Summary

Sensor networks will play a key role in sensing, collecting and dissemination of information about environmental phenomena. Therefore, the security issues become a central concern. Traditional approaches for securing general wireless networks are not suitable for sensor networks, as those protocols do not consider the energy consumption as the first concern. A wireless sensor network without sufficient protection from DoS attacks may not be deployable in many areas. Adding security afterward often fails in typical wireless sensor networks. Thus design-time consideration of security offers the most effective defense against DoS attacks. DoS attacks can happen in each layer of the sensor network protocol. A necessary operation to defeat DoS attacks is to identify the misbehaving nodes.

CHAPTER 3

AN OVERVIEW OF GAME THEORY

Game theory is a branch of applied mathematics that deals with multi person decision making situations. The basic assumption is that the decision makers pursue some well defined objectives and take into account their knowledge or expectations of the other decision makers' behavior. There are two main ways to capitalize game theory. It can be used to analyze existing systems or can be used as a tool when designing new systems. Existing systems can be modeled as games. The models can be used to study the properties of the systems. For example, it is possible to analyze the effect of different kinds of users on the system. The other approach is implementation theory, which is used when designing a new system. Instead of fixing a game and analyzing its outcome, the desired outcome is fixed and a game ending in that outcome is looked for. When a suitable game is discovered, a system fulfilling the properties of the game can be implemented [50].

In Section 3.1 we discuss the formal definition of games. Section 3.2 presents the non-cooperative and cooperative games. Section 3.3 discuses the strategic and extensive games. And in Section 3.4 we present the equilibrium of the games.

3.1 Formal Definitions

A game $G =< N, (A_i), (u_i) >$ consists of a finite set N of players, a nonempty set A_i (of actions for each player $i \in N$), and a von Neumann-Morgenstern [44] utility function $u_i : A_i \rightarrow \Re$, which represents consequences of the actions. The players are decision makers who choose how they act. The actions of the players result in a consequence

or outcome. The players try to ensure the best possible consequence according to their preferences. The preferences of a player can be expressed in terms of a utility function, which maps every consequence to a real number. With mild assumptions, a utility function can be constructed if the preference relations of a player are known.

The most fundamental assumption in game theory is rationality. Rational players are assumed to maximize their payoff. The idea of maximizing the expected payoff was justified by the seminal work of von Neumann and Morgenstern [7]. The maximizing of one's payoff is often referred to as selfishness. This is true in the sense that all the players try to gain the highest possible utility. However, a high utility does not necessarily mean that the player acts selfishly. Any kind of behavior can be modeled with a suitable utility function. For example, a preference model not only pays attention to the benefit to the player, but also the benefit relative to the other players. In many occasions, an Equity Reciprocity and Competition (ERC) model fits experimental data better than simpler models where the players only try to maximize their own benefit [10]. It is also assumed that the players are intelligent, which means that they know everything that we know about the game and they can make the same deductions about the situation that we can make.

In game theory, a solution of a game is a set of possible outcomes. A game describes what actions the players can take and what the consequences of the actions are. The solution of a game is a description of outcomes that may emerge in the game if the players act rationally and intelligently. Generally, a solution is an outcome from which no player wants to deviate unilaterally. When a player makes a decision, he can use either a pure or a mixed strategy. If the actions of the player are deterministic, he is said to use a pure strategy. If different probability distributions are defined to describe the actions of the player, a mixed strategy is used [44].

Games can be classified into different categories according to their properties. The terminology used in game theory is inconsistent, thus different terms can be used for the same concept in different sources.

3.2 Non-cooperative and Cooperative Games

Games can be divided according to their payoff structures. A game is called a *zero-sum* game, if the sum of the utilities is constant in every outcome. Whatever is gained by one player, is lost by the other players. Gambling is a typical zero-sum game. Such games are also called strictly competitive games [44].

Games can be divided into *non-cooperative* and *cooperative* games according to their focus. Cooperative games are also called coalition games. In non-cooperative games, the actions of the single players are considered. Correspondingly, in coalition games the joint actions of groups are analyzed, i.e., what is the outcome if a group of players cooperate. The interest is in what kind of coalitions form. In telecommunications, most game theoretic research has been conducted using non-cooperative games, but there are also approaches using coalition games [39]. Coalition games can be used to analyze heterogeneous ad hoc networks. If the network consists of nodes with various levels of selfishness, it may be beneficial to exclude too selfish nodes from the network if the remaining nodes get better quality of service that way.

Suppose in a game, payoff matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ ($\forall i = 1, ..., n$ and $\forall j = 1, ..., m$), are defined such that a_{ij} denotes player p_1's payoff when player p_1 chooses strategy i and player p_2 chooses strategy j; and b_{ij} denotes player p_2's payoff when player p_1 chooses strategy i and player p_2 chooses strategy j. If $a_{ij} + b_{ij} = 0$,$\forall i = 1, ..., n$ and $\forall j = 1, ..., m$, then it is a zero-sum game, otherwise it is a non-zero sum game. Note that the games in which $a_{ij} + b_{ij}$ is a constant are also categorized as zero-sum as they can be

easily converted to zero-sum games. In zero-sum games, only one player's payoff matrix is enough to define the game and the payoff matrix of the other player can be derived.

3.3 Strategic and Extensive Games

In *strategic* or *static* games, the players make their decisions simultaneously at the beginning of the game. While the game may last long and there can be probabilistic events, the players can not react to the events during the game. On the other hand, the model of an *extensive* game defines the possible orders of the events. The players can make decisions during the game and they can react to other players' decisions. Extensive games can be finite or infinite. A class of extensive games is *repeated* games, in which a game is played numerous times and the players can observe the outcome of the previous game before attending the next repetition [44].

3.3.1 Auctions

In strategic games, players first make their decision and subsequently the outcome of the game is determined. The outcome can be either deterministic or contain uncertainties. The decisions are made without knowledge of other player's decisions [7]. The strategic game called (bidding) consists of

- a finite set of N players
- for each node $i \in \{1, ..., N\}$ a nonempty set A_i (set of actions available to each player i).
- for each node $i \in \{1, ..., N\}$ a von Neumann-Morgenstern utility function $u_i : A_i \rightarrow \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers.

A sealed-bid auction is a typical strategic game with incomplete information [36]. A player knows its own valuation of the packet but does not know the valuation of other bidders. Every strategic game with a finite number of players each with a finite set of

actions has an equilibrium [42, 50]. This Nash equilibrium is a point from which no single player wants to deviate unilaterally.

3.4 Equilibrium

A *Nash equilibrium* of a strategic game $< N, (A_i), (u_i) >$ is an n-tuple $a^* = (a_1^*, ..., a_N^*)$ of actions with the property that for every player $i \in N$, player i is using strategy $a_i \in A_i$:

$$u_i(a^*) \geq u_i(a_1^*, ..., a_{i-1}^*, a_i, a_{i+1}^*, ..., a_N^*) \; for \; all \; a_i \in A_i,$$

A pair of strategies (i^*, j^*) is said to constitute a Nash equilibrium solution to the game if the following pair of inequalities is satisfied: $a_{i^*j^*} \leq a_{ij^*}$ and $b_{i^*j^*} \leq b_{i^*j}$, $\forall i = 1, ..., n$ and $\forall j = 1, ..., m$.

It is shown in [7] that a two-player, non zero-sum game may or may not have a pure strategy Nash equilibrium. Equilibrium can also be established with another form of strategy called *mixed strategy*, which means the equilibrium is established when at least one of the players has a "mix" of possible strategies. A mixed strategy for a player is a probability distribution on the set of his pure strategies. For example, if p_1 has two strategies numbered as s_1 and s_2, a mixed strategy for p_1 means that he has a probability of π to choose strategy s_1, and a probability of $1 - \pi$ to choose strategy s_2. Nash proved that every bimatrix game has at least one equilibrium solution in mixed strategies [44].

The intuition behind the equilibrium is that, when a game reaches an equilibrium no player has any positive reason for changing his strategy, assuming that the other player is not going to change strategies.

CHAPTER 4

SAR: SECURE AUCTION-BASED ROUTING FOR SENSOR NETWORKS

In this chapter we propose a secure routing protocol in sensor networks which is based on the concept of sealed auctioning. In first-price sealed auction [36], always the bidder with the highest bid wins and reaches equilibrium and thereafter the truth bidding is a dominant strategy for sensors. With suitably designed rules, auctions can achieve efficient allocations with minimal a priori information. One of the essential reasons to use auction is to speed up the sale and ignite competition between buyers, which is the main reason to adopt the first-price sealed auction mechanism in the approach presented in this chapter.

The absence of pre-existing infrastructure in sensor networks means that most of the nodes will serve as routers for through traffic. Sensor nodes, either malicious or truthful, compete against each other in order to forward incoming packets and by doing so each node improves its reputation among other nodes. Bidding is done to gain a better reputation in the network and instead of paying money, the winner of the bid disinherits some of its initial energy power. Participation in an auction is a decision that is completely up to the sensor node, whereas a malicious node tries its best to win the bid and then drops the packets and corrupts the network.

In Section 4.1 we propose the auction theory framework, Section 4.2 discusses the bid and Section 4.3 states the equilibrium of the strategies. In Section 4.4 we show the payoff of the game for each player. Section 4.5 depicts the performance evaluation of the proposed approach and Section 4.6 summarizes the chapter.

4.1 Protocol Description

In the proposed Secure Auction-based routing (SAR) protocol, a node sends out a *Route_request* message. All nodes receiving this message place themselves into the source route and forward it to the their neighbors, unless they have received the same request before. If a receiving node is the destination, or has a route to the destination, it does not forward the request, but sends a *Reply* message containing the full source route and the bid price that it is willing to pay. After receiving one or several routes, the source selects the best one having the highest bid; stores it and sends messages along that path. In the SAR protocol, the path is chosen by picking the path from the cache of available paths to the packet's destination with the highest bid, as depicted in Figure 4.1. Once a route request reaches its destination, the path that this route request has taken is reversed and sent back to the sender. This protocol proposes an auction on routes to ensure a view on which nodes will provide likely service due to their commitment. Note that a malicious node could agree to the auction and still subvert the route, so a watch-list facilitates recognizing such faulty nodes. How can we determine if a node is acting maliciously? Destination nodes can send back messages, and when one destination node gets notified of the winning path, it sets a timeout timer. In order to implement the timeout at the receiving node, once the auction ends, the sending node sends a *Winning_route* packet to the destination node, which stores this route and the source. Once the destination node gets a packet from the source (that is not a control message), it removes the source from its list of pending links. If the pending link times out, then the destination node sends a *Bad_route* packet to the base station, which updates its list with the nodes in the route (excluding the source and destination). If a node is placed on the watch-list more often than a pre-defined threshold, the base station sends out a *Watch list_ignore* broadcast, and all of the nodes add that node into their ignore lists. The threshold is high enough

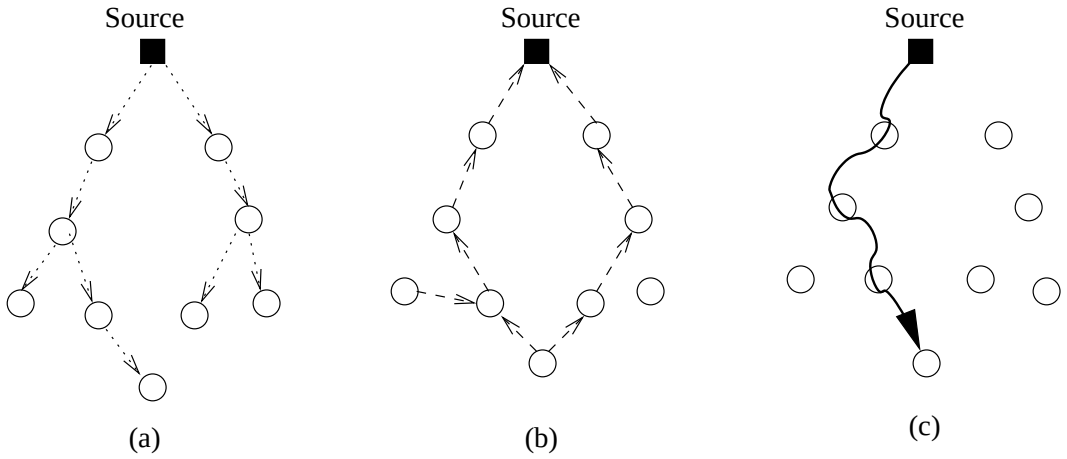

Figure 4.1. (a) ROUTE REQUEST, (b) ROUTE REPLY, (c) Establishing the path.

to distinguish deliberate malicious behavior from simple selfishness of a node. All nodes prefer not to communicate with a node in their ignore list.

4.2 The Bid

In strategic games, players first make their decision and subsequently the outcome of the game is determined. The outcome can be either deterministic or contain uncertainties. The decisions are made without knowledge of other player's decisions [7]. In a sensor network consisting of N sensors, where occasionally some of them act maliciously, there are N players that compete to bid against each other.

Definition 1 *The strategic game called (bidding) consists of*

- *a finite set of N sensor nodes*
- *for each node $i \in \{1, ..., N\}$ a nonempty set A_i of actions available to node i.*
- *for each node $i \in \{1, ..., N\}$ a von Neumann-Morgenstern utility function $u_i : A_i \rightarrow \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers.*

A sealed-bid auction is a typical strategic game with incomplete information [36]. A player/node knows its own valuation of the packet but does not know the valuation of

other bidders. The solution of a strategic game is a Nash equilibrium. Every strategic game with a finite number of players, each with a finite set of actions has an equilibrium [42, 50]. This Nash equilibrium is a point from which no single player wants to deviate unilaterally.

Definition 2 *A Nash equilibrium of a strategic game* $< N, (A_i), (u_i) >$ *is a profile* $a^* = (a_1^*, ..., a_N^*)$ *of actions with the property that for every player* $i \in \{1, ..., N\}$*:*

$$u_i(a^*) \geq u_i(a_1^*, ..., a_{i-1}^*, a_i, a_{i+1}^*, ..., a_N^*) \quad \forall a_i \in A_i,$$

when a game is played, the rationality assumption will force the game into a Nash equilibrium outcome.

If the outcome is not a Nash equilibrium, at least one player would gain a higher payoff by choosing another action. If there are multiple equilibria, more information about the behavior of the players is needed to determine the outcome of the game. We present the model of our proposed framework for secure routing as an auction game. Players of this game are the nodes of the network who bid against each other in order to obtain better reputations in the network.

Each bidder submits a sealed bid of b_i. A winner i will be charged the price Ω_i, which is its battery power loss. Only one path wins a bid, and nodes on this path are assumed to be cooperative. If the nodes on the winning path cooperate, then their reputation will be raised, otherwise another path will be chosen. In order to model the player's strategy, their perception of the bid needs to be presented. Our analysis relies on the fact that players, along with their own absolute profit, are also motivated by the relative profit, which indicates how their standing compares to the profit of other players. The utility of a node is not solely based on the absolute payoff but also on the relative payoff compared to the overall payoff of all nodes. We use the theory of Equity, Reciprocity and Competition (ERC) model presented in [10] as follows: $v_i = \alpha_i u(y_i) +$

$\beta_i r(\sigma_i)$, where α_i, β_i are positive constants and $u(.)$ is differentiable, strictly increasing and concave, and $r(.)$ is differentiable, concave and has its maximum at $\sigma_i = 1/N$. Here v_i is the ERC global utility function, y_i is the absolute profit and $u(y_i)$ is the absolute utility function for player i. The total amount of bid that each node is willing to pay is v_i. The absolute profit value of a node depends on the node's battery power (Ω_i) and its reputation (Φ_i) which will be discussed in section 4.4.

4.3 The Equilibrium Strategy

Each of the $N > 1$ potential bidders knows how much it is willing to pay (v_i). Each node's decision problem can be viewed as of choosing a bid $b(v_i)$ and probability ρ of winning. Suppose b^* is the equilibrium bid strategy, one can show that b^* is monotonically increasing in v, which guarantees that the bidder with the highest evaluation would win the auction [58].

Proposition 1 *For the given utility structure of the bidding game, there is always a Nash equilibrium at* $b^*(v_i) = (1 - \frac{1}{N})v_i$.

Proof: If bidder i bids the amount $b = b^*(v_i)$, he wins with probability: $\rho(b) = Pr\{b^*(v_1, ..., v_{i-1}, v_{i+1}, ..., v_N) < b\} = Pr\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_N\} < \delta(b)\}$, where $\delta(b)$ is the inverse of $b^*(v_i)$. This indicates the valuation that leads to bidding the amount b if strategy b^* is played. In equilibrium, the bid b must be a best-response to bids. Therefore, b must be a maximizer of the expected gain: $\rho(b)(v_i - b)$, leading to the condition: $\rho'(b)(v-b) - \rho(b) = 0$, where ρ' is the derivative of ρ. By using the fact that $v = \delta(b^*(v))$, the differential equation will have the solution of $\delta(b) = (\frac{N}{N-1})b$. Finally solving this for b, the equilibrium bid strategy is given by: $b^*(v) = (1 - \frac{1}{N})v$.□

One can also prove that this equilibrium is unique. For $N = 2$, $\delta(b) = 1/2\delta^2 + \delta(0)$. Since $b(0) = 0$, one has $\delta(0) = 0$, and thus $\delta(b) = 2bv$, where v is the price that bidder

is willing to pay. Hence, the unique equilibrium strategy is $b^*(v) = 1/2v$. In order to generalize the proof to $N > 2$, one can apply a transformation of variables, from (δ, b) to (z, b), where $z = \delta/b$, and by separating variables and uniquely solving the differential equation by integration [58].

The intuition behind this proposition is that, for the defined bidding game, a Nash equilibrium exists. This is a point from which no other node wants to deviate. The maximum amount that each node is able to truthfully bid is its utility. Now that we know where the equilibrium is (which depends on the payoff values calculated at the moment), the network will take the corresponding biding acceptance based on the payoffs calculated. In other words, the equilibrium tells us about the most rational choice for each bidder in the game in certain situation and the network follows that.

4.4 The Payoff

Each node's payoff is calculated based on two parameters, namely, battery power and reputation. In order to compute the required power for each sensor node, communication and computation of sensor nodes are being considered. The communication energy usage is much higher than the computation energy usage. Communication in sensor networks is dependent on the connectivity of the network, where connectivity is defined as the ability to link between any pair of nodes. The connectivity cost is a function of the number of hops, latency, etc. [45]. In a path containing k nodes, we measure the connectivity energy usage as a function of available energy at each node and the total number of en-route hops. $\Omega_i(t)$ is defined as : $\Omega_i(t) = \frac{\sum_i Pow_i(t)}{k}$, where $Pow_i(t) = D_T[P_{i_T}(T_{on}(t) + T_{st}(t)) + P_{i_o ut}(T_{on}(t))] + D_R[P_{i_R}(R_{on}(t) + R_{st}(t))]$, where P_T is the power consumed by the transmitter, P_R is the power consumed by the receiver, P_{out} is the output power of the transmitter, T_{on} is the transmitter on time, R_{on} is the receiver on time, T_{st} is the transmitter start-up time, R_{st} is the receiver start-up time,

D_T is the number of times the transmitter is switched on per unit time, and D_R is the number of times the receiver is switched on per unit time, which depends on the medium access control scheme used [53].

A node that acts maliciously or saves its internal memory and power by not forwarding incoming packets and dropping them for its selfishness, should suffer from bad reputation and be isolated from the rest of the network. On the other hand, when a node does not act selfishly, then it must be rewarded; and the reward it gets is the good reputation. By doing a service for the network, each node will improve its reputation. As time passes, more nodes recognize a node with good reputation. Let $P_i^f(t)$ and $P_i^r(t)$ be respectively the number of packets forwarded and received at sensor node i at time t.

Definition 3 *The reputation of node i, denoted as $\Phi_i(t)$, is defined as the ratio of the number of packets forwarded to the total number of received packets at time t at node i. Thus, $\Phi_i(t)$ is a measure of throughput experienced at each node and calculated as,* $\Phi_i(t) = \frac{P_i^f(t)}{P_i^r(t)}$.

The reputation value decreases when misbehavior is detected. For any given node i, the payoff is given by $\Phi_i(t)$, which is the reputation that it gains over time. But it also must bear some additional cost $\Omega_i(t)$, which is the energy loss. Therefore, the payoff is calculated as, $y_i(t) = \alpha\Phi_i(t) - \beta\Omega_i(t)$, where α and β are weight parameters.

4.5 Performance Evaluation

The simulation of the proposed protocol SAR is implemented in ns2 [43]. Mobility of sensors follows the *Random Way point Model* [13], in which sensors move to a random destination at a speed uniformly distributed, where sensors are scattered in the field. Nodes are deployed inside a rectangular area of 1000×1000 m. The physical layer assumes that two nodes can directly communicate with each other successfully only if they are in each other's transmission range. The MAC layer protocol simulates the

Table 4.1. Parameters for Simulation

Parameters	Values
Area	$1000m \times 1000m$
Speed	uniformly 0-20 m/s
Radio Range	250 m
MAC	802.11
Sending capacity	2 Mbps
Simulation time	1000s
Auctions last	60s
Timeout at the receiving node	20s

IEEE 802.11, and Dynamic Source Routing (DSR) [32] is used as the underlying routing protocol to discover the shortest routes. Three different types of attacks to a sensor network are considered: (i) IP spoofing attack, where a malicious node misrepresents its identity in the network, (ii) the black holes attack, where a malicious node in the route aggressively drops messages that are routed through it, and (iii) falsify route error message attack, where a malicious node issues route error messages to a normal node to indicate a broken link and thus misdirects the path. We have measured the mean number of packets dropped versus time and the total number of malicious nodes present in the network. We have also computed the routing overhead. The fixed parameters for simulation experiments are listed in Table 4.1.

4.5.1 Metrics

The following metrics are considered:

Throughput: One metric is the resulting total throughput of the network with n nodes. We express this as:

$$\frac{\sum_{i=1}^{n} Packets_{Recd}}{\sum_{i=1}^{n} Packets_{Orig}}$$

Here $Packets_{Recd}$ and $Packets_{Orig}$ represent the total number of received and originated packets respectively.

Packet loss can occur due to general network conditions causing link errors or unreachable nodes, but packets can also be lost because an intermediate node intentionally drops them.

Overhead: Since the cost of internal computation in terms of energy consumption is negligible compared to the cost of a transmission, we look at the overhead caused by extra messages and define it as the total number of packets exchanged during route discovery.

4.5.2 Simulation

For simulating spoofing we incorporate a timer. At the beginning of the simulation, the timer is set. When the timer expires, a "spoof" bit is turned on in the node and the timer is reset. Once the "spoof" bit is on, the node will take on the source address of the next received route request. The node will then turn off the "spoof" bit. Figures 4.2 and 4.3 show the mean number of packets dropped, varying the pause times, but keeping the fraction of malicious nodes fixed at a third of the total number of N nodes. In the simulation, we consider $N = 50$ and 100. In the case where no security was enforced, for $N = 50$ nodes the number of packets dropped is greater than in the SAR protocol. Furthermore, SAR stabilizes and drops less packets than the CONFIDANT protocol. And for the case of $N = 100$, we observe that the average number of dropped packets in SAR stays steady and still the total number of dropped packets shows 9% improvement comapred to CONFIDANT. This is due to the fact that in SAR, the nodes with bad reputation will be ignored by the majority of nodes. It can be seen that in a network with no security enforced, even a small percentage of malicious nodes can have deep impact.

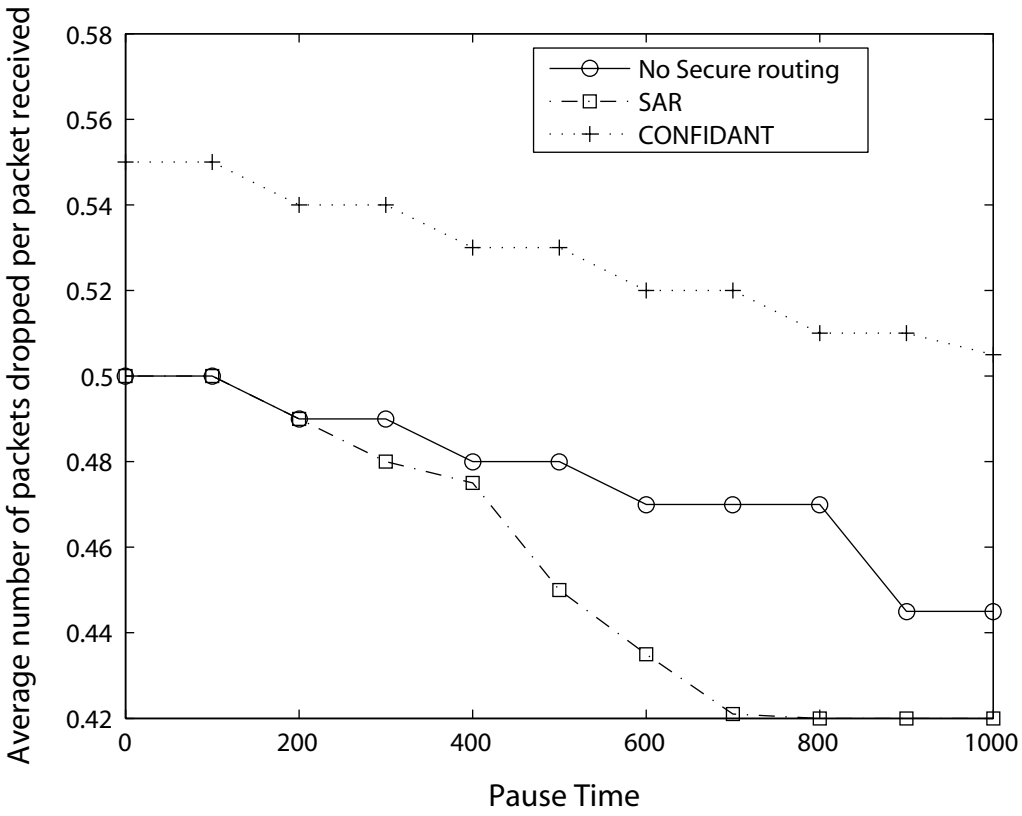

Figure 4.2. Mean number of packets dropped vs. pause time, one third is malicious, $N = 50$ node.

In Figure 4.4, the pause time is set to 0 to stress a very dynamic network. SAR still keeps the number of deliberately dropped packets low, even in a very hostile environment. SAR loses a lower fraction of packets due to malicious nodes, whereas more packets are lost for cases where no security is enforced or the CONFIDANT protocol is applied. The nodes in SAR can avoid bad experience where packets are dropped until the malicious node is avoided in the route, whereas in others the malicious nodes just keep dropping packets without initiating a reaction to the malicious behavior.

Figure 4.5 shows that when the total number of nodes increases, more packets will be dropped due to malicious acts of intermediate nodes. But in SAR after a while, the total number of dropped packets will decrease when the total number of nodes in the

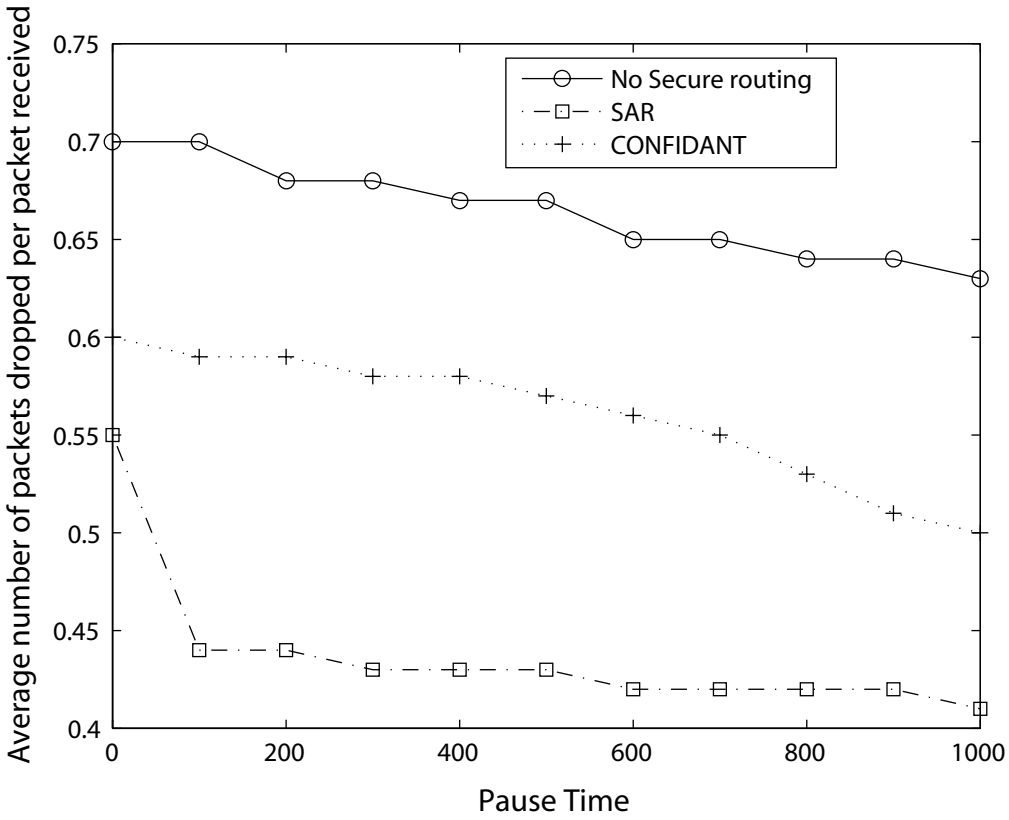

Figure 4.3. Mean number of packets dropped vs. pause time, one third is malicious, $N = 100$ nodes.

network increases. The reason is that when there are more nodes in the network, more nodes can ignore those with bad reputation; but in a smaller network one cannot ignore all the nodes and sometimes has to choose some of them to forward incoming packets.

Figure 4.6 illustrates the case of 70% malicious nodes in the network. More packets are being dropped due to malicious acts of the majority of the nodes in the network, but SAR faces less loss compared to others. This is due to the fact that in other protocols there is less reaction to the bad behavior of nodes. In SAR protocol, as bad behavior propagates throughout the network, when a node is labeled as a malicious one, other nodes in the network ignore it and it woud not be able to harm the network.

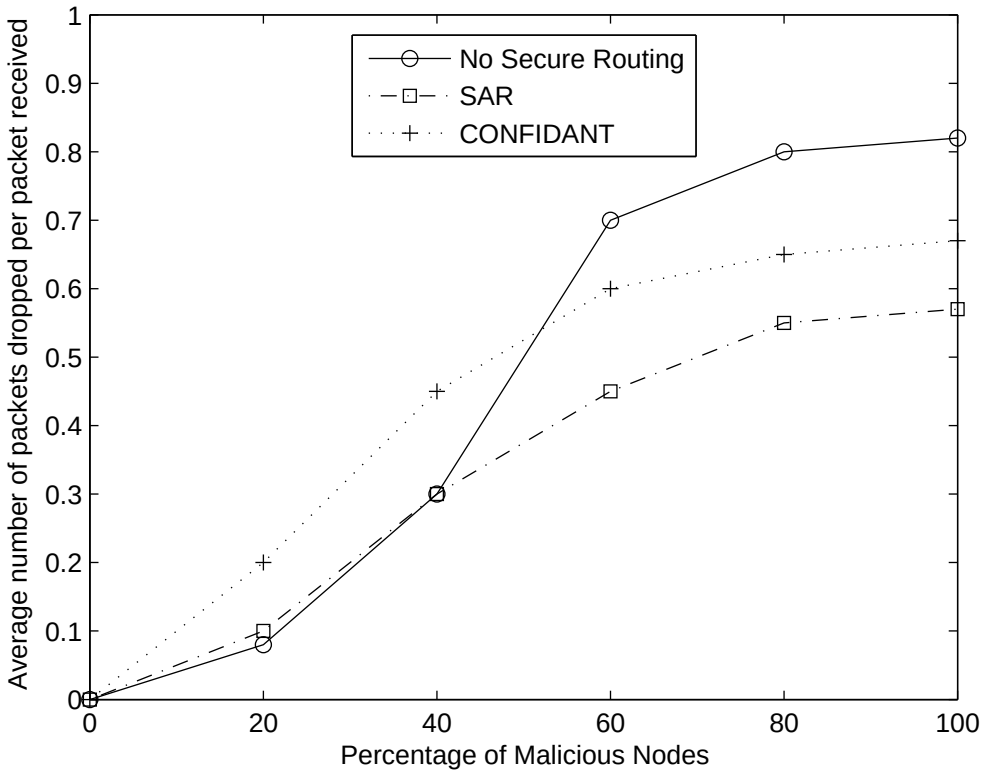

Figure 4.4. Mean number of packets dropped, varying percentage of malicious nodes, $N = 100$ nodes.

Figure 4.7 depicts the reputation of an over-bidder malicious node in a network consisting of 50 nodes where 70% of them are malicious ones. In the begining an over-bidder node gets packets but because it drops them later on, the total reputation value decreases over time. As this figure illustrates, bad behavior propagates throughout the network when a node is labeled as a malicious one. Other nodes in the network ignore it and gradually its reputation declines. Figure 4.8 depicts the reputation of a normal node.

Figure 4.9 depicts the message overhead in SAR and INSENS. This experiment measures the total number of packets exchanged during route discovery. We compare SAR with INSENS and the case when no security is enforced in the network. It is

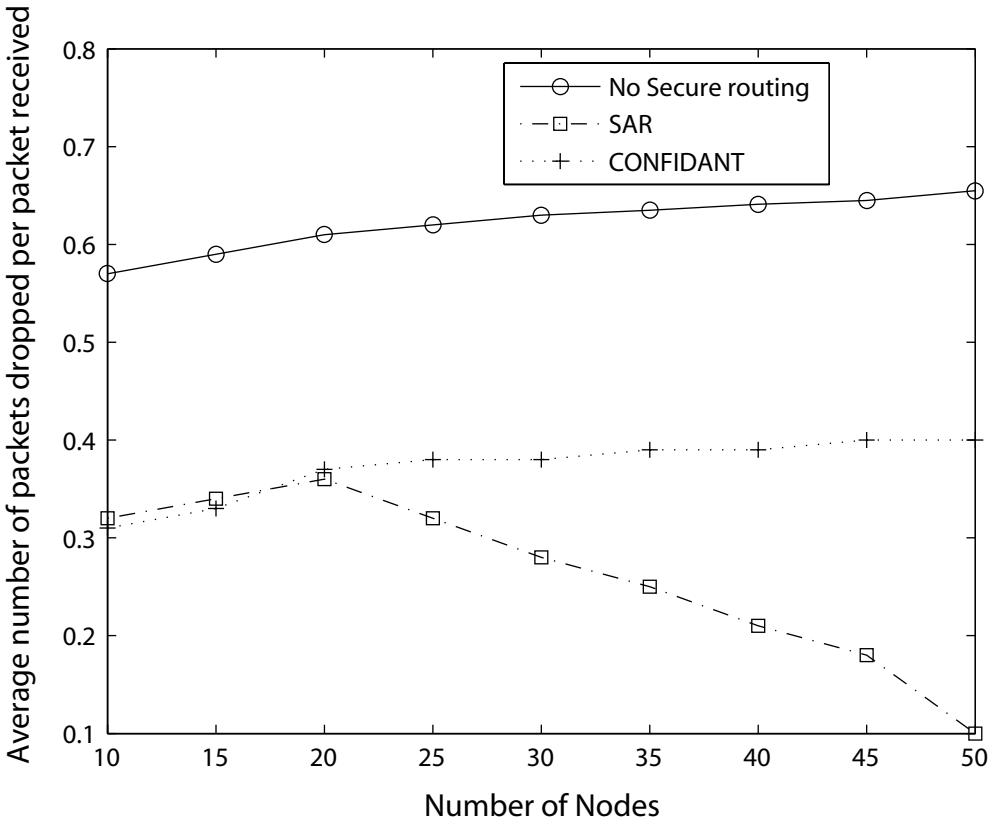

Figure 4.5. Mean number of packets dropped, one third malicious, 0 pause time, $N = 10 \ldots 50$ nodes.

clear that SAR and INSENS send more packets than the trivial one, and the difference increases with increasing numbers of nodes in the network, but compared to INSENS, the proposed protocol SAR has a lower number of message overheads.

We believe that the research we have conducted so far has given interesting results and proposes a useful basis to study the application of the game theory framework to enforce security. However, we think that it is possible to investigate characteristics of the game theory framework by modeling the interactions between decision makers (the attacker and the wireless sensor network) as a non-cooperative game.

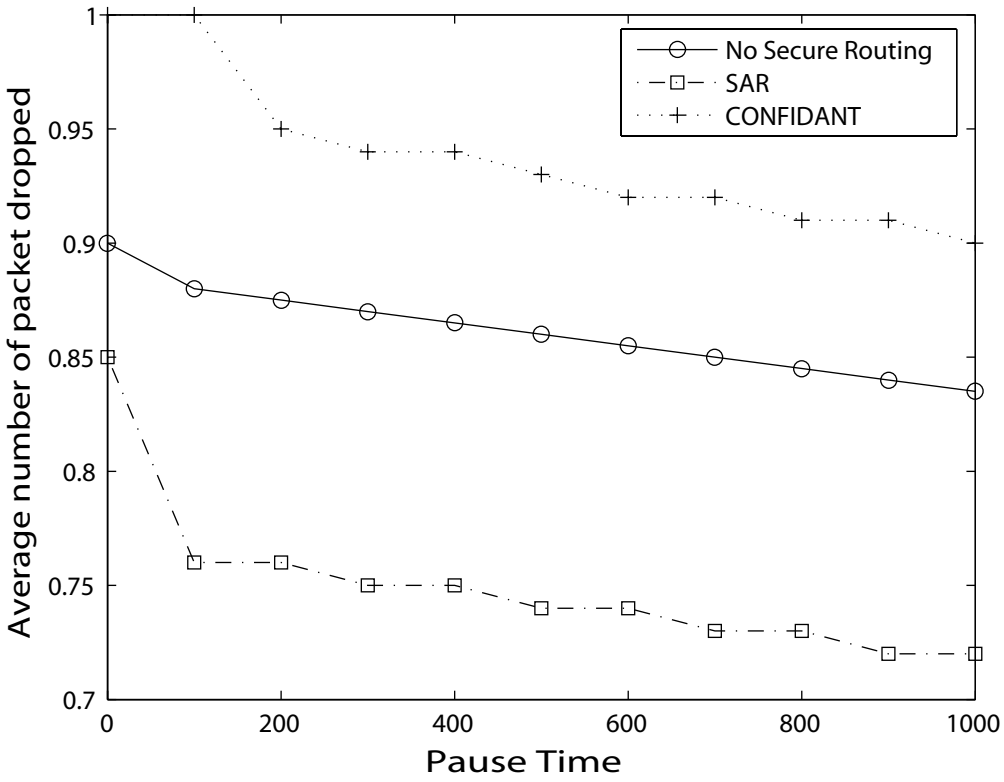

Figure 4.6. Mean number of packets dropped vs. pause time, 70% are malicious.

4.6 Summary

In this chapter, we designed the SAR protocol and studied its performance. Our objective was to measure the effectiveness of different schemes in detecting malicious behavior. The experimental results show that by using an auction based framework and incorporating the utility value of each route which is based on energy power and reputation of en-route nodes, we can guarantee more reliable delivery. And also, by defining an acceptable threshold for utility of sensor nodes, we can observe the behavior of sensor nodes and isolate suspicious ones.

One other goal that we have in mind is to observe the functionality of the proposed protocol, when a subset of bidders gather together and agree not to outbid each other

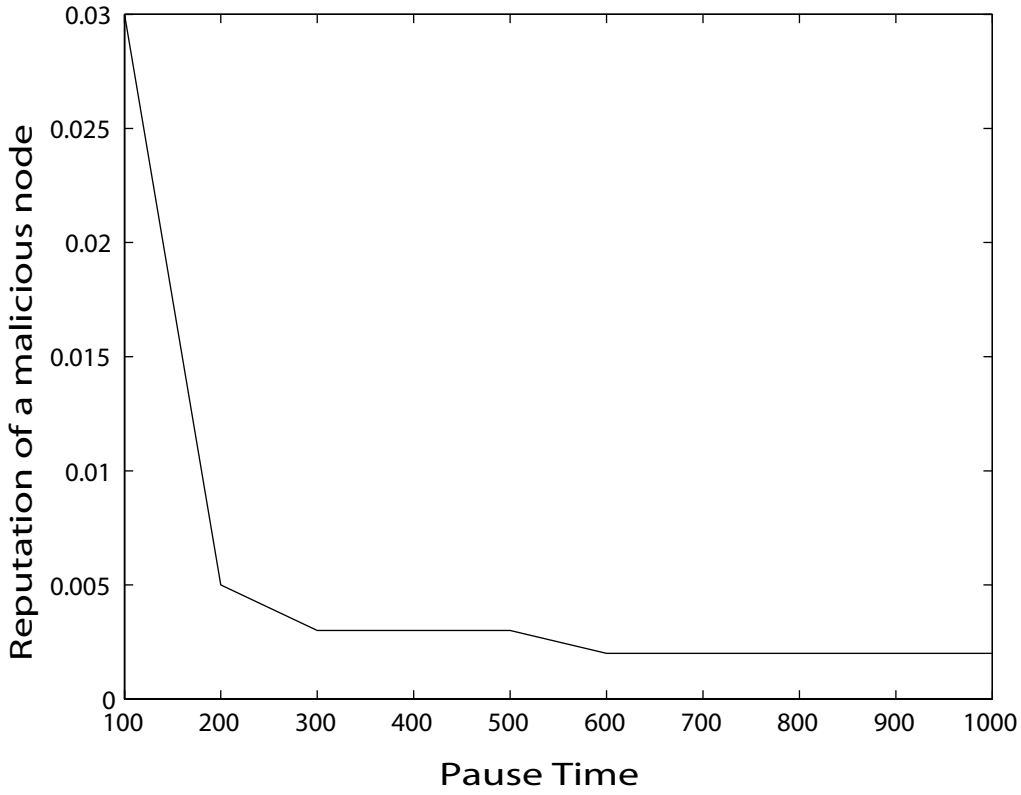

Figure 4.7. Reputation of a malicious node.

which has the overall effect of lowering the winning bid. Motivations of bidders are quite different. In essence, these bidders agree to reduce competition by not competing against each other. We would like to see how this will impact the cooperation between nodes.

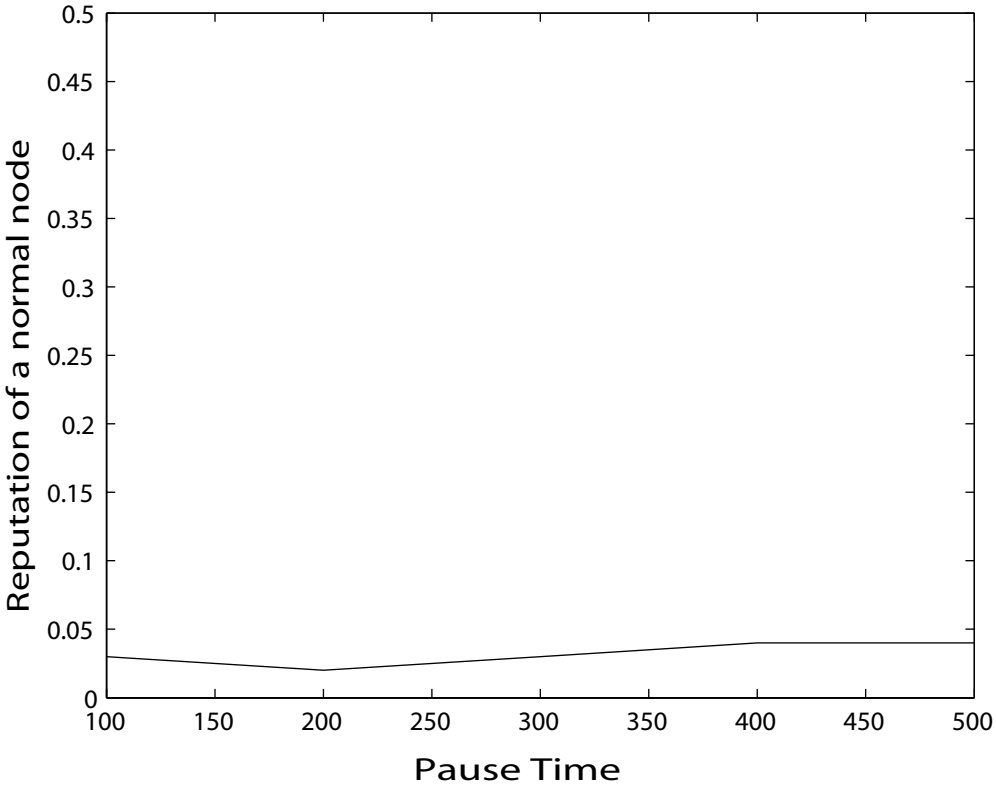

Figure 4.8. Reputation of a normal node.

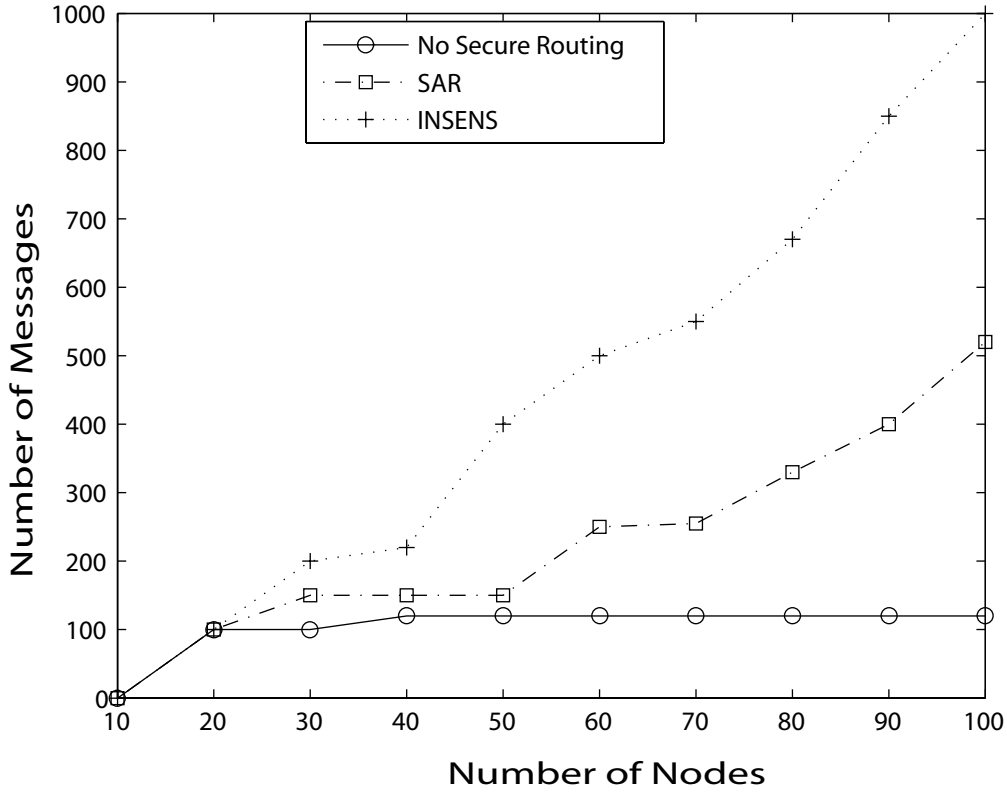

Figure 4.9. Routing overhead.

CHAPTER 5

UTILITY BASED DYNAMIC SOURCE ROUTING (UDSR)

In this approach, we propose a game theoretic framework for security in sensor networks. We define a non-cooperative game between an attacker and the network and show that the game achieves a Nash equilibrium, leading to the defense strategy for the network. We explain how and why this model is suitable to describe the decision making process that sensor nodes would undertake when participating in the security enforcing of a wireless sensor network. We also focus on the strategy that a player adopts to determine whether to cooperate or not and based on that, how to detect node misbehavior. In this chapter we assume that N sensor nodes are already clustered and each cluster has a cluster-head, and players of the game are the cluster-heads. The key to solving problems related to node misbehavior, is when a node does not act selfishly it must be rewarded. The reward is the good reputation. We claim that by doing a service for the network (forwarding incoming packets), each node will improve its reputation and as time passes, more nodes would recognize a node with good reputation. On the other hand, a node that acts maliciously or saves memory and power, by not forwarding the incoming packets and dropping them for its selfishness, should loose reputation and be isolated from the rest of the network.

Similar to other communication networks, scalability is one of the major design attributes of sensor networks. A single-tier network can cause the gateway to overload with the increase in sensor density. Such overload might cause latency in communication and inadequate tracking of events. To allow the system to cope with additional load and to be able to cover a large area of interest without degrading the service, wireless sensor

network clustering has been pursued in [28, 55]. For simplicity, we suppose that sensors are already scattered in a field and clusterized.

This chapter is organized as follows, in Section 5.1 we discuss the formulation of the game, payoff of each player and the solution for the game. In Section 5.2, we describe the performance evaluation of the proposed approach.

5.1 Game Formulation of the Proposed Protocol

In our two-player game consisting of sensor nodes, players are Intrusion Detection System (IDS), which can reside at the base station, and the attacker. With respect to one fixed node, say k, which itself is a cluster-head, the attacker node has the following two strategies as depicted in Figure 5.1:

- AS_1: attack node k,
- AS_2: attack a node different than k.

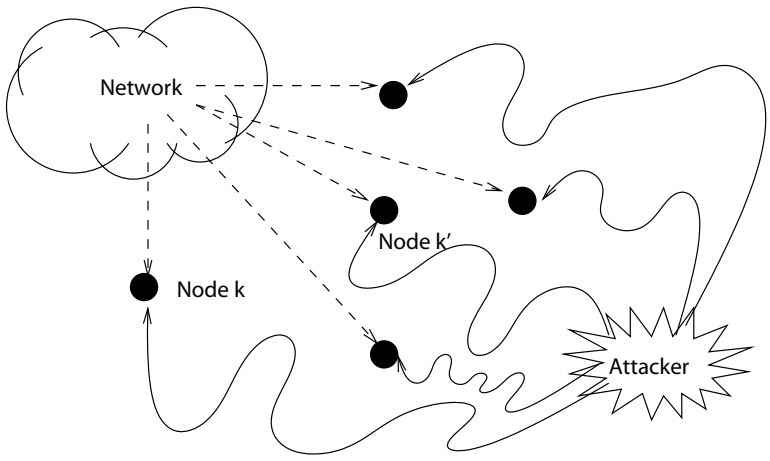

Figure 5.1. The attacker chooses a strategy to attack a node. The network predicts the attacker's strategy by finding the most vulnerable node.

The sensor network also has two strategies:

- SS_1: defend node k,

- SS_2: defend a different node.

The payoffs of these two players are expressed in the form of 2×2 matrices, A and B, where a_{ij} and b_{ij} denote the sensor network's and the attacker's payoff, respectively. Let us first define some notations:

- $U(t)$: the payoff of the network at time t.
- L_k: the cost of loosing a malicious node k.
- C_k: the cost of defending a node k.
- N_k: the number of sensor nodes in the cluster, where node k is the cluster-head.

Two payoff matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, are defined such that a_{ij} denotes player p_1's payoff when player p_1 chooses strategy i and player p_2 chooses strategy j; whereas b_{ij} denotes player p_2's payoff when player p_1 chooses strategy i while player p_2 chooses strategy j. We define the network's payoff matrix as follows:

$$A = [a_{ij}]_{2\times 2} = \begin{bmatrix} U(t) - C_k & U(t) - C_k - E[\sum_{i\neq k} L_i] \\ U(t) - C_{k'} - \sum_{i\neq k'} L_i & U(t) - C_{k'} \end{bmatrix}$$

Here a_{11} represents the payoff if the players follow the strategy pair (AS_1, SS_1), which is when the attacker chooses to attack node k and the network chooses to defend the same node k. Thus, for the network, its original utility value of $U(t)$ will be deducted by the cost of defense (C_k). The term a_{12} represents the payoff corresponding to the strategy pair (AS_2, SS_1), which is when the attacker attacks a different node k', but the network still defends node k. In this case we subtract the cost of defending one node from the original utility, as well as deducting the expected value of losing another node, which can be any other node than k. The term a_{21} represents the payoff of strategy pair (AS_1, SS_2), that is the attacker and the network choose two different nodes to attack and to defend, respectively. The term a_{22} represents the payoff of strategy tuple (AS_2, SS_2), which is when the attacker attacks a node other than k and the network defends another

node. In this case we subtract the cost of defending one node, from the original utility, as well as deducting the loss of losing another node.

We define the attacker's payoff matrix as follows:

$$B = [b_{ij}]_{2\times 2} = \begin{bmatrix} Cost_{int} & P(t) - Cost_{int} \\ P(t) - Cost_{int} & Cost_{int} \end{bmatrix}$$

where,

- $Cost_{int}$: the cost of any successful intrusion for the attacker.
- $P(t)$: the profit of each successful attack for the attacker, at time t.

Here each b_{ij} represents an attack, we subtract cost of attack from profit of concurring a node.

For the network, the cost of defense is the price it must pay to protect a node that is most likely to be under attack. We assume it is dependent on two parameters: (i) the cost of protecting a node which is more important in the network, like an aggregation point, must be higher than the cost of protecting a normal node, and (ii) the cost must be dependent on the number of nodes communicating with that node.

The cost of defending a node i for the network is given by: $C_i = \gamma_i + N_i$, where γ_i is the weight of a node i (the more important a node is, the higher is its weight), and N_i is the number of nodes in cluster i.

The profit from each attack for the attacker and the loss of losing a node for the network are dependent on the density of nodes that are communicating with that node, and the reliability $r_i(t)$ of each node. The density μ can range from few sensor nodes to few hundred sensors. Following [14], we get $\mu(R) = \frac{NR^2}{Ar}$, where N is the number of sensor nodes in region Ar, and R is the radio transmission range. The node density depends on the application in which the sensor nodes are deployed. In general, the density can be as high as 20 sensor nodes/m^3 [53].

Definition 4 *The loss of losing a node for the network is defined as:* $L_k = \mu \prod_{i=1}^{N_k} r_i(t)$.

Now in order to find an equilibrium of this game, we turn our attention to Nash's theorem [44], which proved that such games have at least one equilibrium in mixed strategies. In a given game represented by a payoff matrix $A_{m\times n}$, $x \in \Re^m, y \in \Re^n$ form a pair of mixed strategies if $x \in X = \{x \in \Re^m; \sum_{i=1}^{m} x_i = 1, x_i \geq 0\}$, and $y \in Y = \{y \in \Re^n; \sum_{i=1}^{n} y_i = 1, y_i \geq 0\}$. Let us suppose that the network can play:

$$X = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

where $x_1 + x_2 = 1$. An attacker can also play:

$$Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

where $y_1 + y_2 = 1$. Then, the payoff function can be computed as: $P(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} x_i y_j$. Therefore:

$$\begin{aligned} P(x,y) &= [U(t) - C_k]x_1y_1 + [U(t) - C_k - \sum_{i,(i\neq k)} L_i]x_1y_2 \\ &\quad + [U(t) - C_{k'} - \sum_{i,(i\neq k')} L_i]x_2y_1 + [(U(t) - C_{k'}]x_2y_2 \end{aligned}$$

and by solving the equation above, the equilibrium of the game in mixed strategies occurs at:

$$X = \begin{bmatrix} \frac{-2C_{k'} - \sum_{i\neq k'} Li}{\sum_{i\neq k'} L_i + \sum_{i\neq k} L_i} & \frac{2C_{k'} + 2\sum_{i\neq k'} L_i + \sum_{i\neq k} L_i}{\sum_{i\neq k'} L_i + \sum_{i\neq k} L_i} \end{bmatrix}$$

The intuition behind the above equilibrium is that for,

$$X = \begin{bmatrix} \alpha & 1-\alpha \end{bmatrix}$$

where α is the number of times the network should defend node k and $1-\alpha$ is the number of times it should defend a different node from node k. As any player wants to maximize its worst payoff, we need to compute the value of the game at each node, and the network

should protect the node with the highest degree of importance, which is the one with highest value of the game.

In our proposed protocol UDSR, a node sends out a *ROUTE REQUEST* message, all nodes that receive this message, calculate their utility value, put themselves and their utility value into the source route and forward it to the their neighbors, unless they have received the same request before. If a receiving node is the destination, or has a route to the destination, it does not forward the request, but sends a *REPLY* message containing the full source route and the total utility value. It sends that reply along the source route in reverse order. After receiving one or several routes, the source selects the best one having the highest average value of utility per number of hops; stores it and sends messages along that path.

Here the IDS's task is recognizing the most vulnerable node to protect, if it protects a node which is the one attacker is trying to attack then the attack is unsuccessful and if a node is part of the Route Reply message, then the requested packet will be delivered. But as IDS protects node k, only α number of times, there is possibility of protecting a different node and as a consequence the route could consist of a malicious node, this will impact the drop rate, and as a result the utility of the IDS will not be the maximum. Note that for forwarding a packet, the proposed protocol always chooses the path with the highest amount of utility, but this utility will change based on the performance of IDS in recognizing the malicious nodes.

In the UDSR protocol, the route is chosen by picking the route from the cache of available paths to the packet's destination with the highest average utility value. Once a route request reaches its destination, the path that this route request has taken is reversed and sent back to the sender. Along the way, each node adds its utility value to the route reply based on the previous node in the path. Once this route reply reaches its destination, the total utility value is divided by the number of hops in the path, and

this average utility value is used to determine the most secure route for the packet to be delivered.

In addition to the above protocol, we studied the addition of a watch list. Each node monitors the behavior of its next-hop neighbors by keeping a table which contains the calculated cooperation and reputation of each of these neighbors. Periodically, each node will send this table to the base station. Then the base station calculates the average value of reputation and cooperation of each node. If these values are less than a predefined threshold value, then that node will be considered suspicious and the base station will notify all nodes through a broadcast packet to drop and ignore any routes which contain this suspicious node.

5.2 Performance Evaluation

The first protocol we analyze is DSR, which is the underlying routing protocol to discover the shortest routes. We then introduce compromised nodes that do not cooperate with the rest of the network. The simulation of the proposed UDSR protocol is implemented in ns2. Mobility of sensors follows the *Random Way point Model*, in which sensors move at a speed uniformly distributed and they are scattered in the field. Nodes are deployed inside a rectangular area of 1000 $m \times 1000$ m. The physical layer assumes that two nodes can directly communicate with each other successfully only if they are in each other's transmission range. The MAC layer protocol simulates the IEEE 802.11. Two different types of attacks to a sensor network are considered: (i) IP spoofing attack, and (ii) the black holes attack. We have measured the mean number of packets dropped versus time and the total number of malicious nodes present in the network. Table 5.1 lists the parameters that used for simulation.

Table 5.1. Parameters for Simulation

Parameter	Values
Area	$1000m \times 1000m$
Speed	uniformly 0-20 m/s
Radio Range	250 m
MAC	802.11
Sending capacity	2 Mbps
Simulation time	1000s

5.2.1 Metrics

The following metrics are considered:

Throughput: Total throughput of the network with n nodes is expressed as:

$$\frac{\sum_{i=1}^{n} Packets_{Recd}}{\sum_{i=1}^{n} Packets_{Orig}}$$

Dropped Packets: Packet loss can occur due to general network conditions causing link errors or unreachable nodes, but packets can also be lost because an intermediate node intentionally drops them.

5.2.2 Simulation

Figure 5.2 shows the mean number of packets dropped, varying the pause times, but keeping the fraction of malicious nodes fixed at a third of the total number of nodes. In the original DSR, the number of packets dropped is up to two orders of magnitude greater than in the UDSR protocol.

In Figure 5.3, the pause time is set to 0 to stress the UDSR protocol with a very dynamic network. It can be seen that in a DSR protocol, even a small percentage of malicious nodes can have deep impact on the total number of dropped packets. UDSR still keeps the number of deliberately dropped packets low even in a very hostile environment. UDSR loses a lower fraction of packets due to malicious nodes, whereas DSR faces more

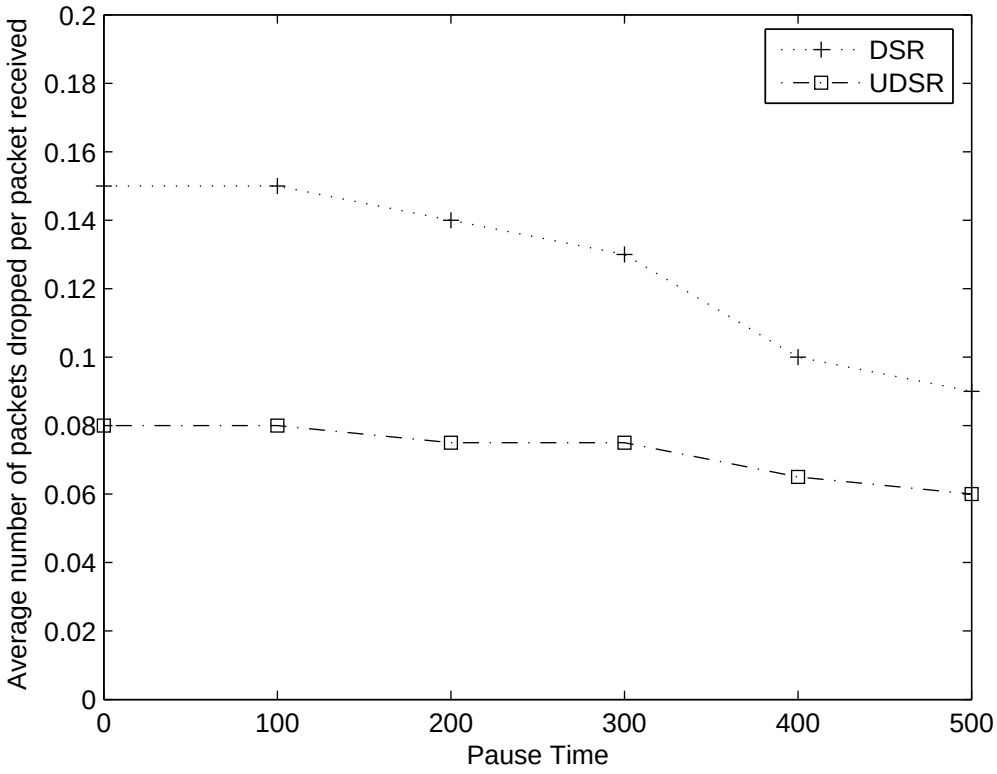

Figure 5.2. Mean number of packets dropped vs. pause time, one third is malicious, $N = 100$.

packet loss. The nodes in USDR can avoid bad experience where packets are dropped until the malicious node is avoided in the route, whereas in DSR the malicious nodes just keep dropping packets without initiating a reaction to the malicious behavior.

Figure 5.4 shows that in both original DSR and UDSR, when the total number of nodes increases, more packets are dropped due to malicious acts of intermediate nodes, but the UDSR protocol keeps the number of dropped packets fairly constant irrespective of the network size. Figure 5.5 illustrates the case of 70% malicious nodes in the network. More packets are being dropped due to malicious acts of the majority of the nodes in the network, but UDSR faces less loss compared to DSR. The reason is that in DSR there is no reaction to the bad behavior of nodes but in our proposed approach, as bad behavior

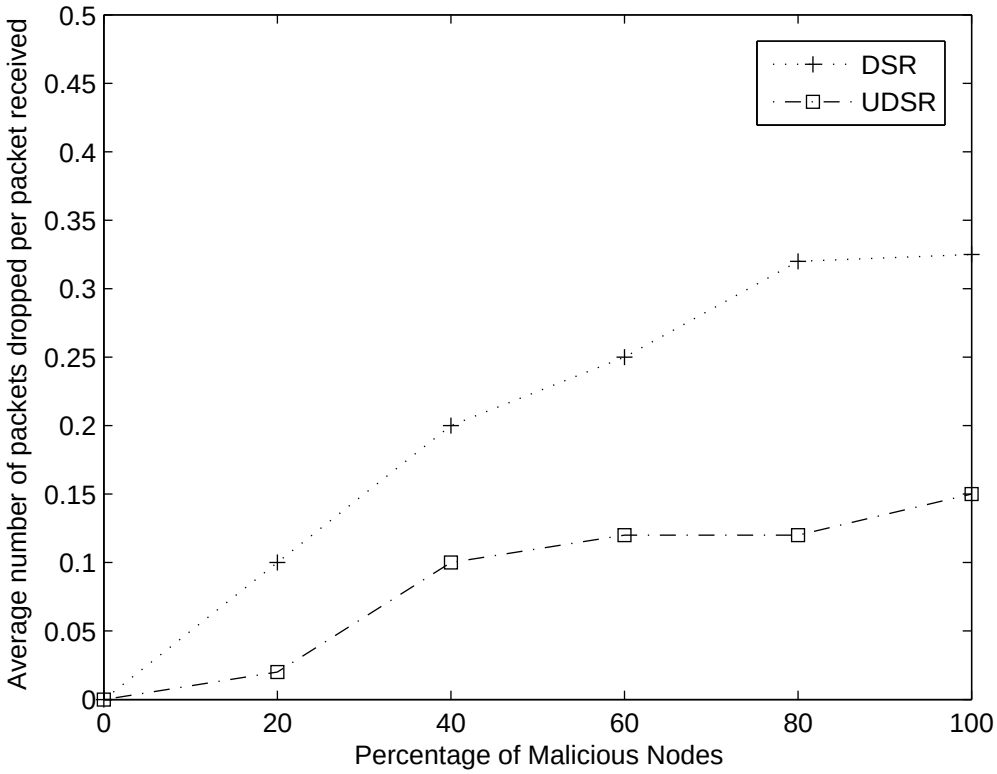

Figure 5.3. Mean number of packets dropped, varying percentage of malicious nodes, 0 pause time.

will propagate throughout the network, when a node is labeled as a malicious one, other nodes in the network ignore it and it would not harm the network.

Figure 5.6 depicts the mean number of dropped packets vs. total number of nodes in the network when 70% of them are acting malicously. It shows that when the total number of nodes increases, more packets will be dropped due to malicious acts of intermediate nodes; and the DSR protocol keeps the number of dropped packets fairly constant irrespective of the network size. But the total number of dropped packets in the UDSR protocol is even less than that in the case when we add the watch-list. The reason behind better performance of UDSR is that, when there are more nodes in the network that act maliciously, the watch-list approach needs to do more updates of neighbor rating. This

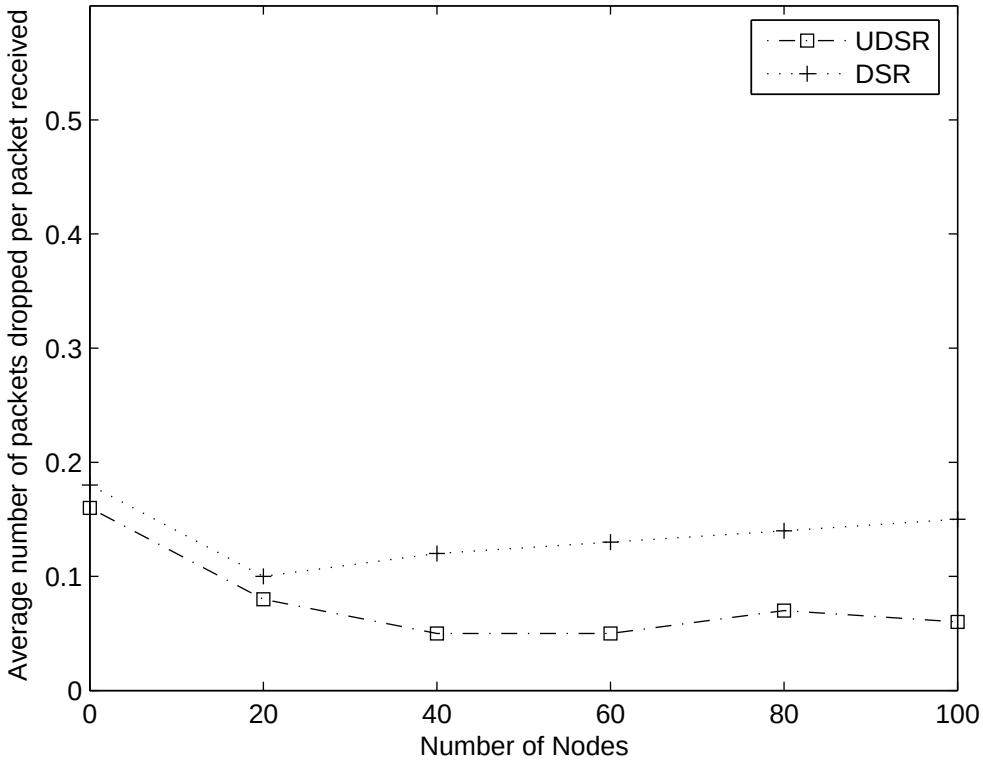

Figure 5.4. Mean number of packets dropped, one third malicious, 0 pause time.

takes more time than UDSR and so more packets will be dropped until the base station informs other nodes.

Figure 5.7 shows the mean number of packets dropped, varying the pause times, but keeping the fraction of malicious nodes at 70% of the total number of nodes. In all three approaches, the total number of droppd packets is increasing, which can be explained by the increased probability of meeting a previously unkown malicious node when nodes move around more. With the help of a watch-list, nodes in the UDSR protocol can avoid the initial bad experience (packets are dropped until the malicious node is avoided in the route), whereas in the DSR the malicious nodes just keep dropping packets without provoking a reaction.

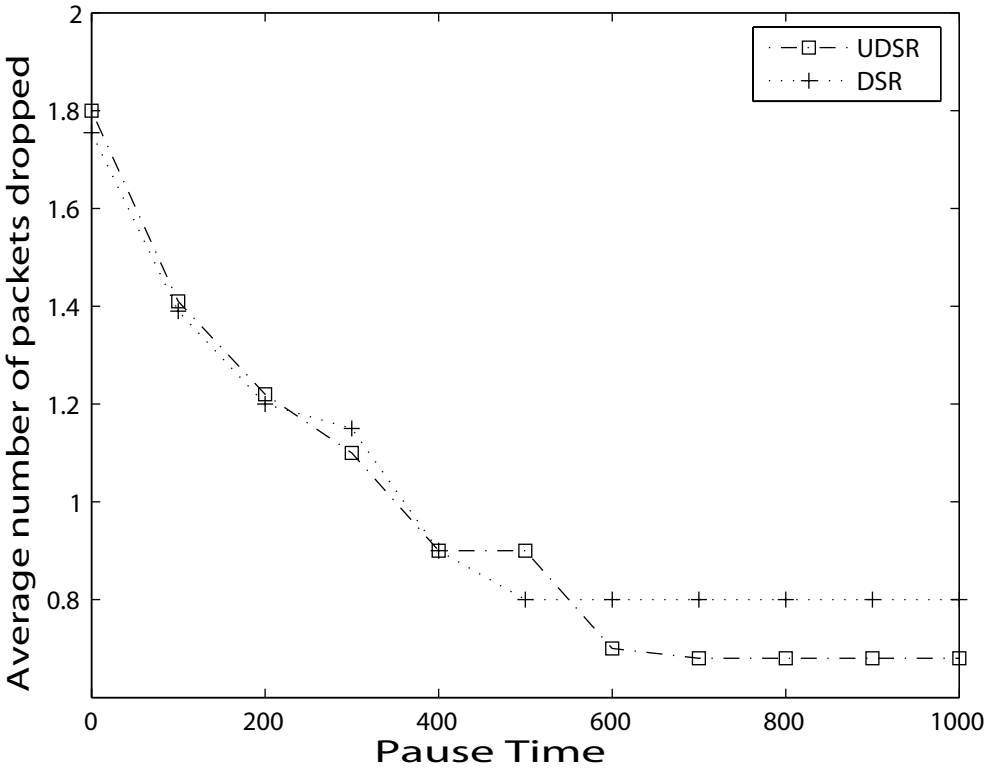

Figure 5.5. Mean number of packets dropped vs. pause time, 70% are malicious.

5.3 Summary

In this chapter, we studied the performance of two protocols: UDSR and watch-list protocol in wireless sensor networks under a game theoretic framework. Our objective was to measure the effectiveness of these schemes in detecting malicious behavior. In order to maximize their own benefits, malicious nodes have no incentive to be cooperative. Therefore, when designing cooperation strategies, optimality criteria such as Nash equilibrium must be taken into consideration. We studied a two-player packet forwarding game, and described how non-cooperation can be translated into a strategy for a player. The experimental results show that by including the utility value of each route which is

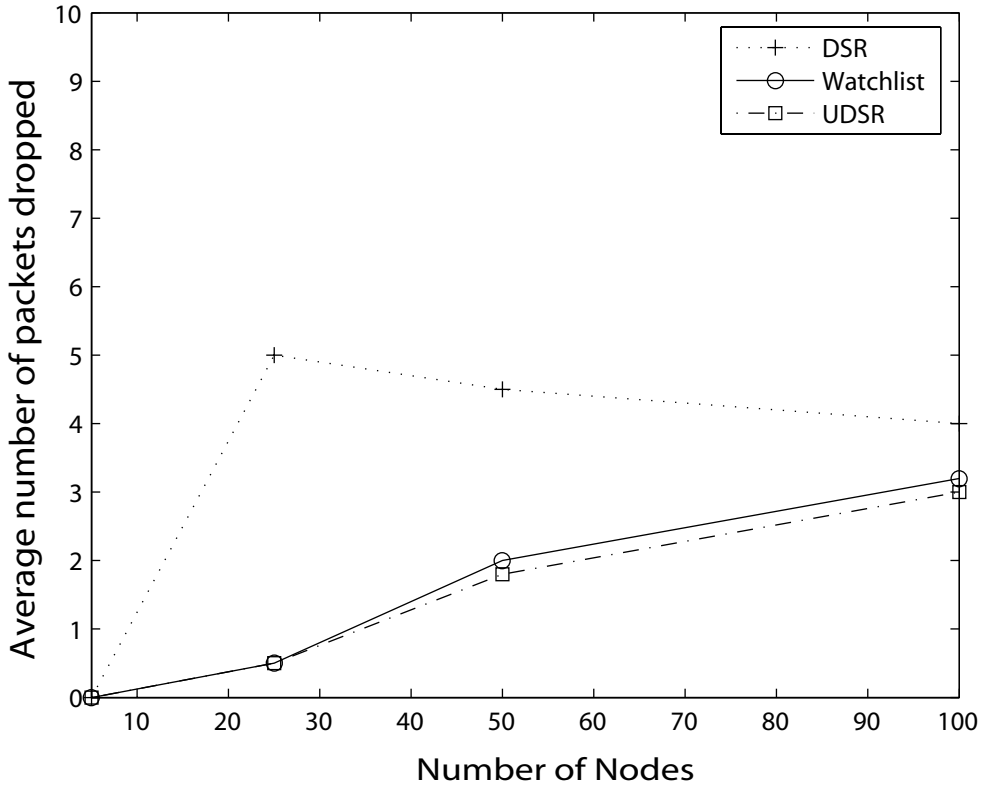

Figure 5.6. Mean number of dropped packets vs. number of nodes.

based on cooperation and reputation of en-route nodes, we can guarantee a more reliable delivery.

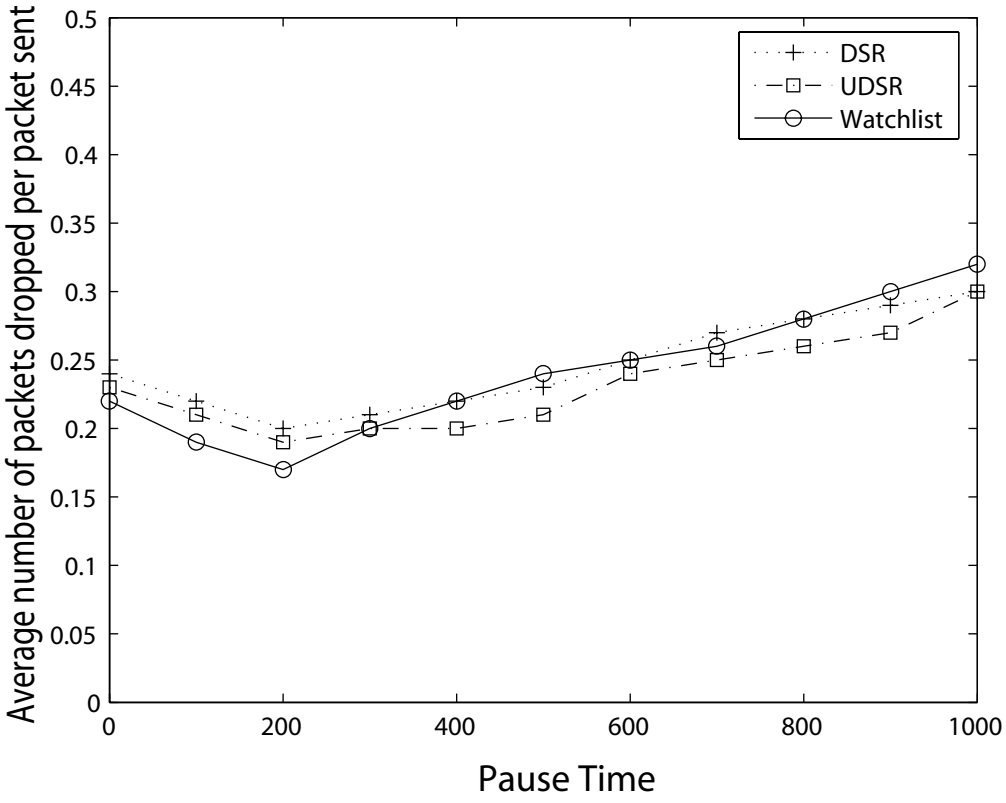

Figure 5.7. Mean number of dropped packets per received packets vs. pause time.

CHAPTER 6

PREVENTION OF DoS ATTACK USING REPEATED GAMES

This chapter formulates the prevention of passive denial of service (DoS) attacks in wireless sensor networks as a repeated game between an intrusion detector and nodes of a sensor network, where some of these nodes act normally and some act maliciously. Intrusion detection systems (IDSs) extend the information security paradigm beyond traditional protective network security. They monitor the events in the system and analyze them for any sign of a security problem [13]. Considering current intrusion detection systems, there is definitely a need for a framework to address attack modeling and response actions. Game theory addresses problems where multiple players with different objectives compete and interact with each other in the same system; such a mathematical abstraction is useful for generalization of the problem. In order to prevent DoS, we capture the interaction between a normal and a malicious node in forwarding incoming packets, as a non-cooperative N player game [44]. The intrusion detector residing at the base station keeps track of nodes' collaboration by monitoring them. If performances are lower than some trigger thresholds, it means that some nodes act maliciously by deviation. The IDS rates all the nodes, which is known as subjective reputation [38], and the positive rating accumulates for each node as it gets rewarded.

Our proposed framework enforces cooperation among nodes and provides punishment for non-cooperative behavior. We assume that the rational users optimize their profits over time. The key to solve this problem is when nodes of a network use resources, they have to contribute to the network life in order to be entitled to use resources

in the future. The intrusion detector keeps track of other nodes behavior, and as nodes contribute to common network operation their reputation increases.

To understand the concept of repeated games, let us start with an example, which is known as the Prisoner's Dilemma [50], in which two criminals are arrested and charged with a crime. The police do not have enough evidence to convict the suspects, unless at least one confesses. The criminals are in separate cells, thus they are not able to communicate during the process. If neither confesses, they will be convicted of a minor crime and sentenced for one month. The police offers both the criminals a deal. If one confesses and the other does not, the confessor one will be released and the other will be sentenced for 9 months. If both confess, both will be sentenced for six months. This game has a unique Nash equilibrium in which each player chooses to cooperate in a single-shot setting.

However, in a more realistic scenario a particular one shot game can be played more than once, in fact a realistic game could even be a correlated series of one shot games. So what a player does early on can affect what others choose to do later on. In particular, one can strive to explain how cooperative behavior can be established as a result of rational behavior. This does not mean that the game never ends; we will see that this framework is appropriate for modeling a situation when the game eventually ends but players are uncertain about exactly when the last period is.

Now in the prisoner's dilemma, suppose that one of the players adopts the following long-term strategy: (i) choose to cooperate as long as the other player chooses to cooperate, (ii) if in any period the other player chooses to defect, then choose to defect in every subsequent period. What should the other player do in response to this strategy? This kind of games is known as repeated games with sequences of history-dependent game strategies.

This chapter is organized as follows. Section 6.1 formulates the game while in Section 6.2 we discuss the equilibrium. Payoff of the game is discussed in Section 6.3. Section 6.4 evaluates the performance of proposed protocol.

6.1 Game Formulation of the Proposed Protocol

We model the interaction between nodes (normal or malicious) and IDS in a sensor network as a repeated game. N players play a non-cooperative game at each stage of the game, where players of the game are an IDS residing at the base-station and N sensor nodes. We first define the stage game, then define the uncertainty that players have about the game. Finally, we define what strategies the players can have in the repeated game.

Consider a game G, which will be called the stage game. Let the players/nodes set to be $I = \{1, ..., N\}$, and refer to a node's stage game choices as *actions*. So each node has an action space A_i. If it is a malicious node then sometimes its action is dropping of the incoming packets.

Let a_i^t refer to the action of the stage game G which node i executes in period t. The action profile played in period t is just the n-tuple of individuals' stage game actions $a^t = (a_1^t, ..., a_n^t)$. We want to be able to condition the nodes' stage game action choices in later periods upon actions taken earlier by other nodes. To do so, we need the concept of *history* which is a description of all the actions taken up through the previous periods. We define the history at time t as $h^t = (a^0, a^1, ..., a^{t-1})$. In other words, the history at time t specifies which stage game action profile was played in each previous period. So we write node i's period-t stage game as the function s_i^t, where $a_i^t = s_i^t(h^t)$ is the stage game action it would play in period t if the previous play had followed the history h^t. When the game starts, there is no past play, every node executes its a_i^0 stage game. This zero-th period play generates the history $h^1 = (a^0)$, which will be recorded at the sink,

where $a^0 = (a_1^0, ..., a_n^0)$. This history is then revealed to the IDS so that it can condition its period-1 play upon the period-0 play. It means that if a node is acting maliciously, by keeping history of the game, the IDS is able to notify neighboring cluster-heads of a malicious one. Each node chooses its $t = 1$ stage game, strategy $s_i^1(h^1)$. Consequently, in the $t = 1$ stage game the stage game strategy profile $a^1 = s^1(h^t) = (s_1^1(h^1), ..., s_n^1(h^1))$ is played.

Each node i has a von Neumann-Morgenstern utility function defined over the outcomes of the stage game G, as $u_i : A \rightarrow \Re$, where A is the space of action profiles. Let G be played several times and let us award each node a payoff which is the sum of the payoffs it got in each period from playing G. Then this sequence of stage games is itself a game, called a *repeated game*. Here,

$$u_i^t = \alpha r_i^t - \beta c_i^t$$

where r_i^t is the gain of node i's reputation, c_i^t is the cost of forwarding a packet for the node, and α and β are weight parameters. We assume that measurement data can be included in a single message that we call a packet. Packets all have the same size. The transmission cost for a single packet is a function of the transmission distance. In particular, we assume $c_i^t = c'.d^{\mu}$, where c' is a constant that includes antenna characteristics, d is the distance of the transmission and μ is the path loss exponent [49].

By assuming that in each period the same stage game is being played, two statements are implicit:

- For each node, the set of actions available to it in any period in the game G is the same regardless of which period it is and regardless of what actions have taken place in past.

- The payoffs to the nodes from the stage game in any period depend only on the action profile for G which was played in that period, and this stage game payoff to a node for a given action profile for G is independent of which period it is played.

We now define the players' payoff functions for the repeated game. When studying repeated games, we are concerned about a player who receives a payoff in each of many periods. In order to represent the performance over various payoff streams, we want to meaningfully summarize the desirability of such a sequence of payoffs by a single number. A common assumption is that the player wants to maximize a weighted sum of its per-period payoffs, where it weights later periods less than earlier periods. For simplicity this assumption often takes the particular form that the sequence of weights forms a geometric series for some fixed $\delta \in (0, 1)$, each weighting factor is δ times the previous weight. δ is called discount factor. If in each period t, player i receives the payoff u_i^t, then we could summarize the desirability of the payoff stream $u_i^0, u_i^1, ...$ by the number:

$$(1 - \delta) \sum_{t=0}^{\infty} \delta^t u_i^t$$

Such a preference structure has the desirable property that the sum of the weighted payoffs will be finite. It is often convenient to compute the average discounted value of an infinite payoff stream in terms of a leading finite sum and the sum of a trailing infinite stream. For example, suppose that the payoffs v_i^t a player receives are some constant payoff v_i' for the first t periods, and thereafter it receives a different constant payoff v_i'' in each period. The average discounted value of this payoff stream is:

$$
\begin{aligned}
(1-\delta)\sum_{\tau=0}^{\infty}\delta^{\tau}v_i^{\tau} &= (1-\delta)(\sum_{\tau=0}^{t-1}\delta^{\tau}v_i^{\tau}+\sum_{\tau=t}^{\infty}\delta^{\tau}v_i^{\tau}) \\
&= (1-\delta)v_i'\sum_{\tau=0}^{t-1}\delta^{\tau}+(1-\delta)v_i''\frac{\delta^t}{1-\delta} \\
&= (1-\delta)v_i'\frac{1-\delta^t}{1-\delta}+\delta^t v_i'' \\
&= (1-\delta^t)v_i'+\delta^t v_i''
\end{aligned}
$$

Now we need to specify the strategies for each of these players. Each node makes the decision whether to (i) accept a packet and forward it to improve its own reputation in the network, we call this action "Normal"; or (ii) do not cooperate and save battery life and stay selfish, we call this action "Malicious". On the other hand, IDS always wants to catch a malicious node but it depends on how well it can detect an intrusion. Thus the output of IDS actions are either (i) "Catch" a node as malicious, or (ii) "Miss" it. As depicted in Figure 6.1, in cases of false positives and false negatives, payoff of one player is the maximum when it is the minimum for the other player. The most important case (rewarding for IDS) is when a node acts maliciously and IDS is able to catch it. IDS has different utility values based on which case happens and how we would like to give different weights to false positives and false negatives detections. For simplicity, we assume $U(Miss, Normal) = v'$, $U(Catch, Normal) = v''$, $U(Miss, Malicious) = v'''$, and $U(Catch, Malicious) = v''''$.

At each stage game, the IDS concurrently plays an N-person game with N different nodes and several possible strategies can be described for it. We want a strategy that punishes it even for its own past deviations (false negatives). We define the utility of IDS as: $U_{IDS} = \gamma_1 v'''' - \gamma_2 v''' - \gamma_3 v''$, where each γ_i represents the number of occurrences of case i. We consider the following retaliation strategy for IDS: in the initial period every node plays cooperatively and so IDS does not catch anyone; in later periods, IDS does

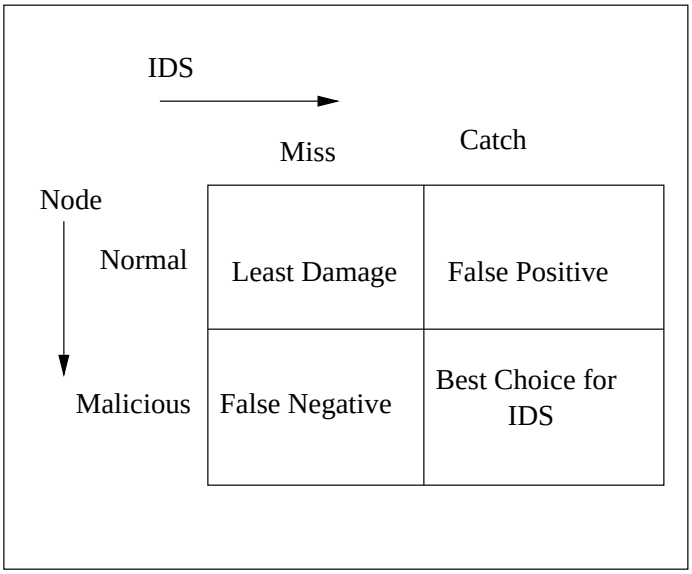

Figure 6.1. Possible cases of interaction between IDS and a node.

not catch if the node has always played normal. However, if a node acts maliciously, then the IDS catches it for the remainder of the game. More formally, the IDS has the following strategy:

$$s_{IDS}(h^t) = \begin{cases} Miss & \text{if } t = 0 \\ Miss & \text{if } a_i^{t-1} = Normal \\ Catch & \text{otherwise} \end{cases}$$

Each node in the initial period plays normally and so IDS does not catch anyone, in later periods, a node does not act maliciously if the IDS has missed it. However, if the IDS catches a node, then the node acts maliciously for the remainder of the game. More formally for a node, we have the following strategy:

$$s_i(h^t) = \begin{cases} Normal & \text{if } t = 0 \\ Normal & \text{if } a_i^{t-1} = Miss \\ Malicious & \text{otherwise} \end{cases}$$

6.2 Equilibrium

First, we show that the above strategies reach to Nash-equilibrium of the repeated game. Both players (sensor nodes and IDS) play cooperatively at $t = 0$. Therefore at $t = 1$, the history is $h^1 = (Miss, Normal)$; so they both play cooperatively again. Therefore at $t = 2$, the history is $h^2 = ((Miss, Normal), (Miss, Normal))$, and so on. The repeated game payoff to each player corresponding to this path is trivial to calculate.

Can IDS gain from deviating from the repeated game strategy given that a sensor node is faithfully following it? Let t be the period in which IDS first deviates. It receives a payoff of v' in the first t periods and in period t, IDS plays "Catch" while sensor node played "Normal", yielding IDS a payoff of v'' in that period. This defection by IDS triggers "Malicious" always response from node. The best response of IDS to this strategy is to "Catch" in every period itself. Thus it receives v'''' in every period $t + 1, t + 2, ...$

To calculate the average discounted value of this payoff stream, we see that the player receives v_i' for the first t periods, then receives v_i'' only in period t and receives v_i'''' every period thereafter. Therefore, the average discounted value of this stream is:

$$(1 - \delta^t)v_i' + \delta^t[(1 - \delta)v_i'' + \delta v_i'''']$$

By solving the above inequality for δ and calculating the average discount value of this payoff, while substituting $v'''' > v'' > v' > v'''$, one possible discount factor necessary to sustain cooperation is $\delta \geq 1/2$. In other words, for $\delta \geq 1/2$, the deviation is not profitable. This means that if IDS is sufficiently patient (i.e., if $\delta \geq 1/2$) then the strategy of retaliation is a Nash equilibrium of the infinitely repeated game. We see that with this strategy the optimal response for IDS is to cooperate and not deviate. In other words, in any stage game reached by some player having "defected" in the past,

each player chooses the strategy "defect always". Therefore, the repeated game strategy profile is a sequence of Nash-equilibria.

6.3 Payoff and Reputation

The problem of generating reliable information in sensor networks can be reduced to one basic question: How do sensor nodes trust each other? Embedded in every social network is a web of trust with a link representing the amount of trust between two individuals. Here IDS monitors the behavior of other nodes, based on which it builds up their reputation over time. It uses this reputation to evaluate their trustworthiness and in predicting their future behavior. At the time of collaboration, a node only cooperates with those nodes that it trusts. Here the objective is to generate a group of trustworthy sensor nodes.

In order to compute the values of a node's gain, we turn our attention to the work proposed in [38]. In this work the authors proposed the concept of subjective reputation, which reflects the reputation calculated directly from the subject's observation. In order to compute each node's gain at time t, we use the following formula:

$$r_i^t = \sum_{k=1}^{t-1} \rho_i(k)$$

where $\rho_i(k)$ represents the ratings that the IDS has given to node i, and $\rho_i \in [-1, 1]$. If the number of observations collected since time t is not sufficient, the final value of the subjective reputation takes the value 0. IDS increments the ratings of nodes on all actively used paths at periodic intervals. An actively used path is one on which the node has sent a packet within the previous rate increment interval. Recall that reputation is the perception that a person has of another's intentions. When facing uncertainty, individuals tend to trust those who have a reputation for being trustworthy. Since reputation is not a physical quantity and only a belief, it can be used to statistically predict the future

Table 6.1. Parameters and Notations

Cost of forwarding packet at node i	c_i
History at node i	h_i
Rating of node i	ρ_i
Reputation at node i	r_i
Utility at node i	u_i
Weight Parameters	α_i, β_i

behavior of other nodes and can not define deterministically the actual action performed by them. Table 6.1 depicts the notations that were used throughout this chapter.

6.3.1 Protocol Description

In the proposed protocol, a node sends out a *Route_request* message. All nodes receiving this message compute their utility based on their local reputation and cost, place themselves into the source route and forward it to their neighbors, unless they have received the same request before. If a receiving node is the destination, or has a route to the destination, it does not forward the request, but sends a *Reply* message containing the full source route with the total utility. After receiving one or several routes, the source selects the best one having the highest utility, which means this route consists of the most reputed possible nodes; stores it and sends messages along that path. Once a route request reaches its destination, the path that this route request has taken is reversed and sent back to the sender. As the destination notifies the base station of the receipt of the packet, the base station gives a higher reputation value to every node on the route, and broadcasts the new reputation values to nodes. As each node is aware of its neighboring node (in its transmission range), it will update the reputation table. This protocol ensures a view on which nodes will provide likely service due to their commitment, as they want to increase their reputation in the network. IDS also

wants to recognize the malicious nodes and isolates them from participating in network functions, but it would prefer not to risk it and have the least amount of false detections, to increase its own utility. The benefit of using a framework based on repeated games is that, the base station has a history of the previous games and when a node is malicious it gets a negative reputation when the total reputation accumulates, a path consisting of less number of malicious nodes is chosen to be the wining path. This results in isolation of malicious nodes.

6.4 Performance Evaluation

For simplicity we assume the following: (i) sensors are scattered in a field, (ii) in the beginning each battery has the same maximum energy, (iii) two sensors are able to communicate with each other if they are within transmission range, (iv) sensors perform a measurement task and periodically report to a sink, and (v) IDS is present at the sink and constantly monitors all nodes for any sign of maliciousness. The sensor network consists of some malicious nodes which occasionally launch DoS attacks.

6.4.1 Metrics

Number of hops for received packets: Malicious behavior affects performance in a number of ways. We consider different topologies, and see the effect of starving multi-hop flows and giving all the capacity to one-hop flows.

Throughput: This measure characterizes the total number of forwarded packets over the total number of received packets.

6.4.2 Implementation

Figure 6.2 illustrates throughput as a function of the percentage of attackers. The figure indicates that without any attacking node, legitimate nodes spend 60% of their

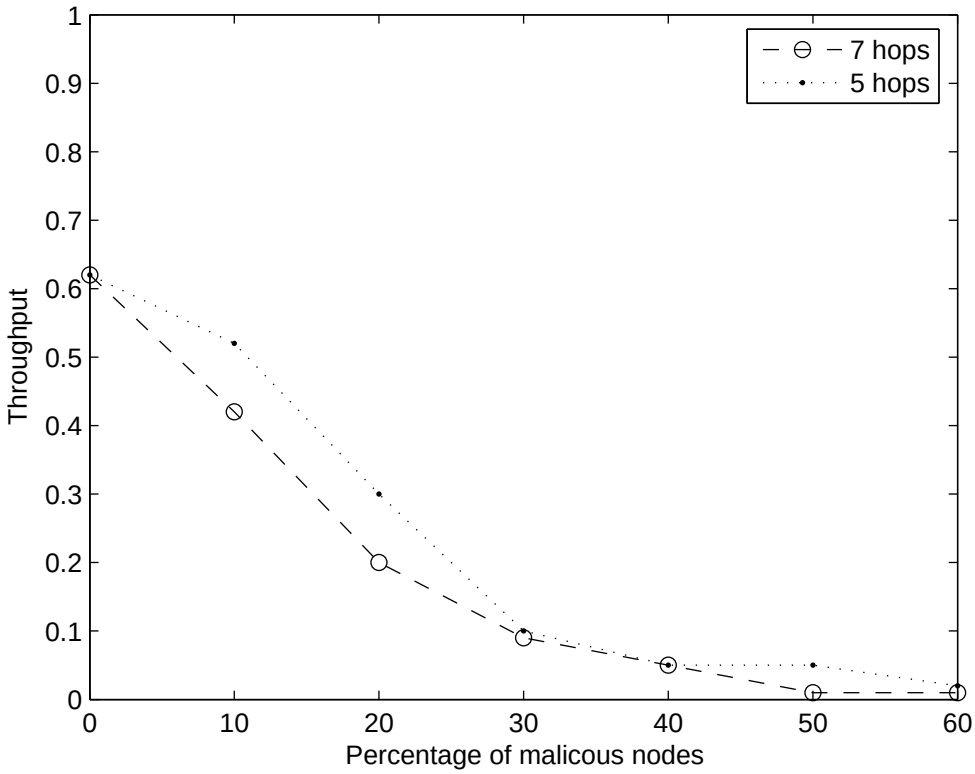

Figure 6.2. Throughput vs. number of malicious node.

time successfully transmiting, and the remaining 40% having broken routes and trying to re-establish routes due to the quality of routes. We can observe the scalability of the attack for 5 hop nodes: with 10% of attacking nodes, the throughput drops to 52%, whereas with 20% of attacking nodes, the throughput drops to 35%. We belive that the impact of the attacker is even more prominent in large-scale networks in which a longer path length is increasingly likely to include an attacking node.

Figure 6.3 depicts the average hop length for received packets. Without attack, the mean is 7 indicating that a significant number of packets are received on long routes. Yet, as the number of malicious nodes grows, the average path length for a received packet

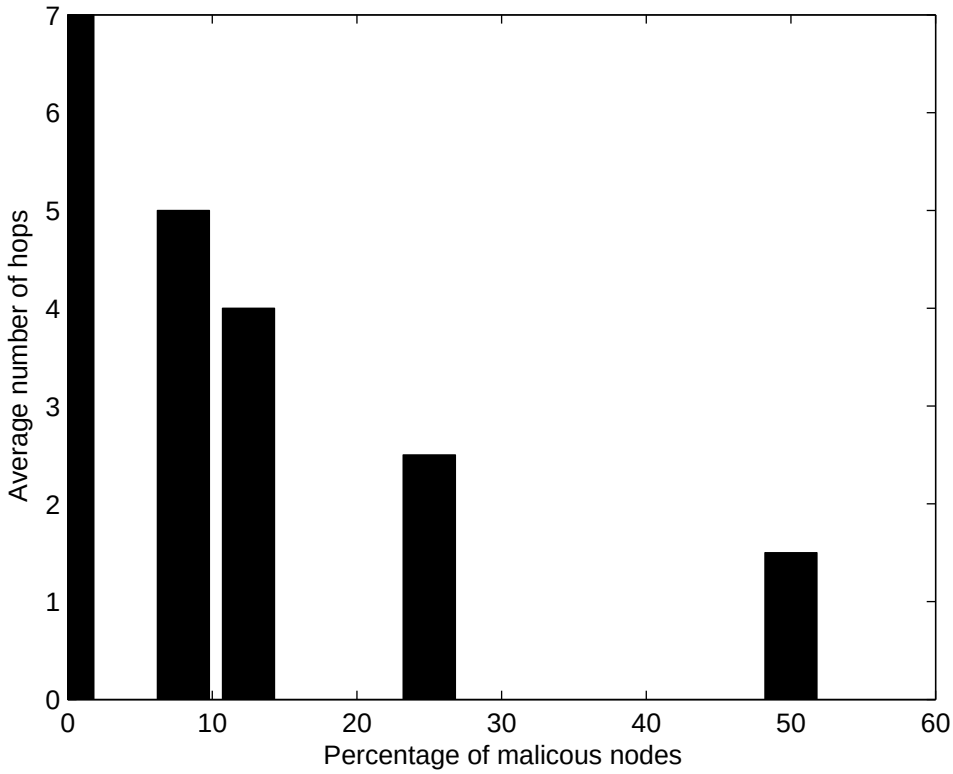

Figure 6.3. Average number of hops for received packets.

diminishes: fewer and fewer packets are able to traverse long routes leading to increased capacity for one-hop flows.

Figure 6.4 indicates the throughput of a node versus time. As the figure depicts, when a node acts maliciously its average throughput drops compared to when it acts normally. The reason behind increase in the throughput over time is that for simulating packet drop, we manually switched off the power switch on the board, and malicious nodes were turned off for shorter duration of time as we proceed with this experiment.

In the original case, we consider a $2m \times 2m$ topology with 18 nodes. Here we also consider a scenario with half the density. Figure 6.5 shows that for very low densities the average number of hops is relatively low in spite of the large dimensions of the topology.

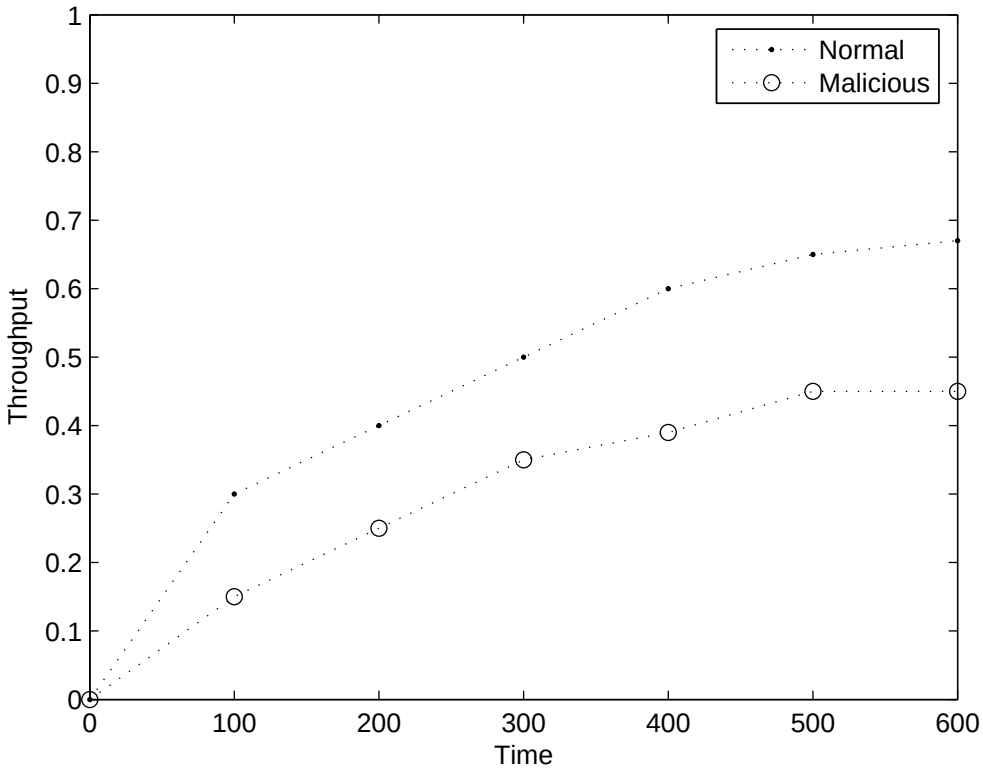

Figure 6.4. Throughput.

In fact, due to the low density, the network is not fully connected such that long-range flows are unlikely to exist.

Also, we explore the effect of system size (number of nodes) on successfully attack detection in Figure 6.6. We can observe that with the presence of 60% malicious nodes, the IDS is able to detect correctly 60% of the time, but as we have a large number of nodes present in the area the rate of success degrades.

Finally, Figure 6.7 depicts the percentage of malicious node detection by IDS. We run the experiments for 100 times for two scenarios, (i) 30% of nodes are malicious and (ii) 60% of nodes are malicious. As predicted, when we have more malicoius nodes present in the network the success rate of IDS degrades. This is due to the fact that IDS prefers

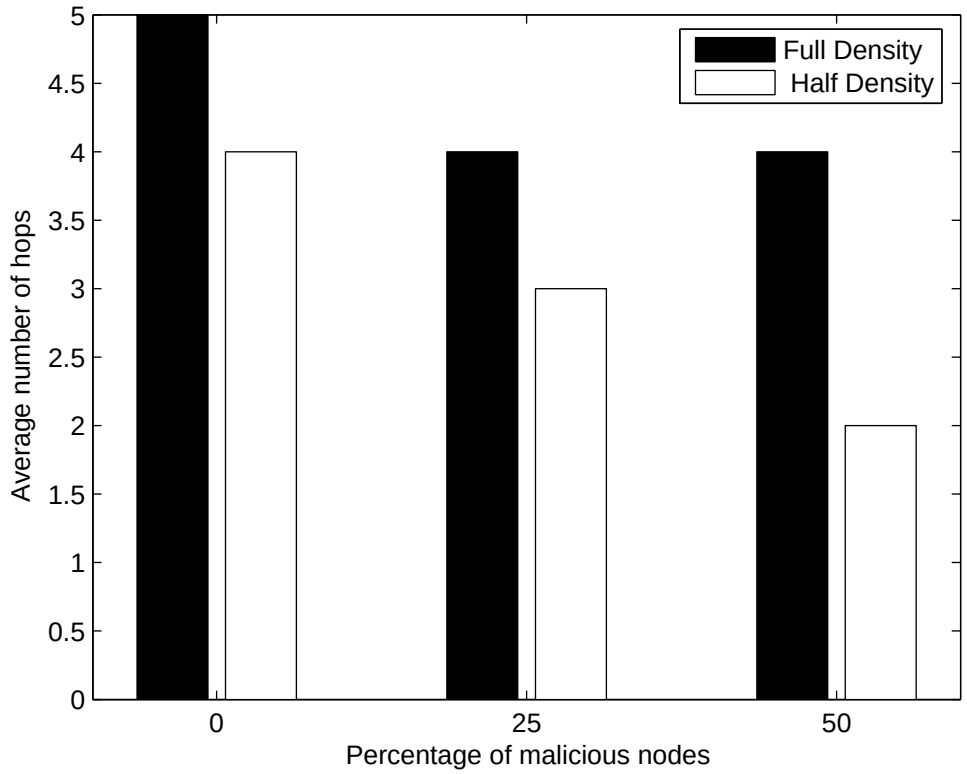

Figure 6.5. Percentage of malicious nodes vs. number of hops.

to maximize its own utility and so it has to lower the rate of false positives and flase nagatives detection and eventually it misses more malicious nodes.

6.5 Summary

Infinite repetition can be the key for obtaining behavior in the stage games which could not be equilibrium behavior if the game were played once or a known finite number of times. In the proposed protocol, IDS rates nodes through a monitoring mechanism. The observations collected by the monitoring mechanism are processed to evaluate reputation of each node. We ensure the finiteness of the repeated-game payoffs by introducing *discount* of future payoffs relative to earlier payoffs.

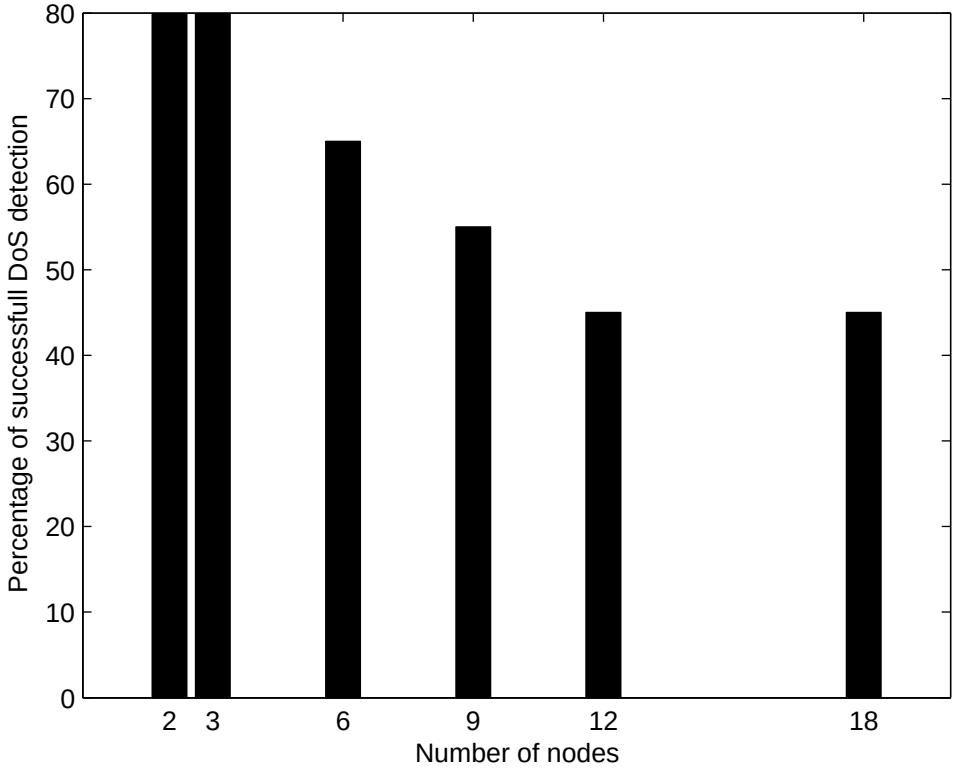

Figure 6.6. Percentage of malicious nodes vs. number of nodes.

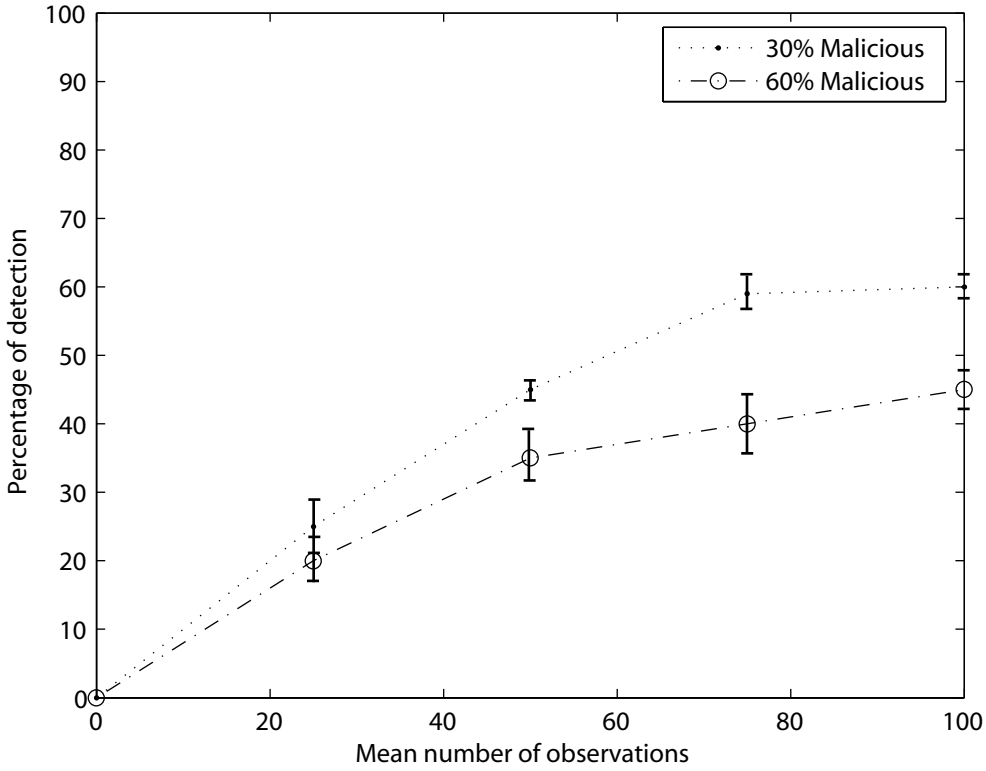

Figure 6.7. Percentage of correct detection.

CHAPTER 7

TEST-BED IMPLEMENTATION

Figure 7.1. MIB510.

We have implemented our proposed approaches for security enforcement on Mica2 [11]. We have used MTS510CA series sensors, which include a flexible sensor board with a variety of sensing modalities. These modalities can be exploited in developing sensor networks for a variety of applications, like sensing, movements, etc. MTS510CA has a light sensor as a simple photocell. In order to use the light sensor, the digital control signal PW0 must be turned on. The output of the sensor is connected to the analog-digital converter channel 7 (ADC7).

Figure 7.2. Mote.

The hardware components that were used were: (i) Programming Board (MIB510), used for programming the Motes and for housing the base station Mote for wireless communication (shown in Figure 7.1), (ii) Motes, which include a processor that runs TinyOS [56]. It is capable of radio transmission and reception. It has a 51-pin connector for housing the sensor (shown in Figure 7.2), (iii) Sensor, has the capability to sense data and transmit it using the processor/radio module, (shown in Figure 7.3).

Figure 7.3. Sensor Board.

We use a Java program for displaying the multi-hop routing topology in sensor networks. It detects the existence of all the motes in a wireless network, displays mote identification number and the number of messages sent from each mote [11]. Figure 7.4 is a screen shot of the sensor networks of 14 node.

7.1 Procedure for Running the Sensor Program

Surge_Reliable and all the XSensor-Series applications are not included in the main tiny0s-1.x distribution. These applications are on the TinyOS Support Tools CDROM under Crossbow [11] Software/xbow.zip. Unzip this file in the *opt/tinyos-1.x/contrib* directory. Drivers are located under *contrib/xbow/tos/sensorboards/*.

- Upload some number of mote with given mote ID, any value except 0.
- Upload one mote as base station, with the given mote ID 0.
- Put the base station on the programming board and run the SerialForwarder [56]

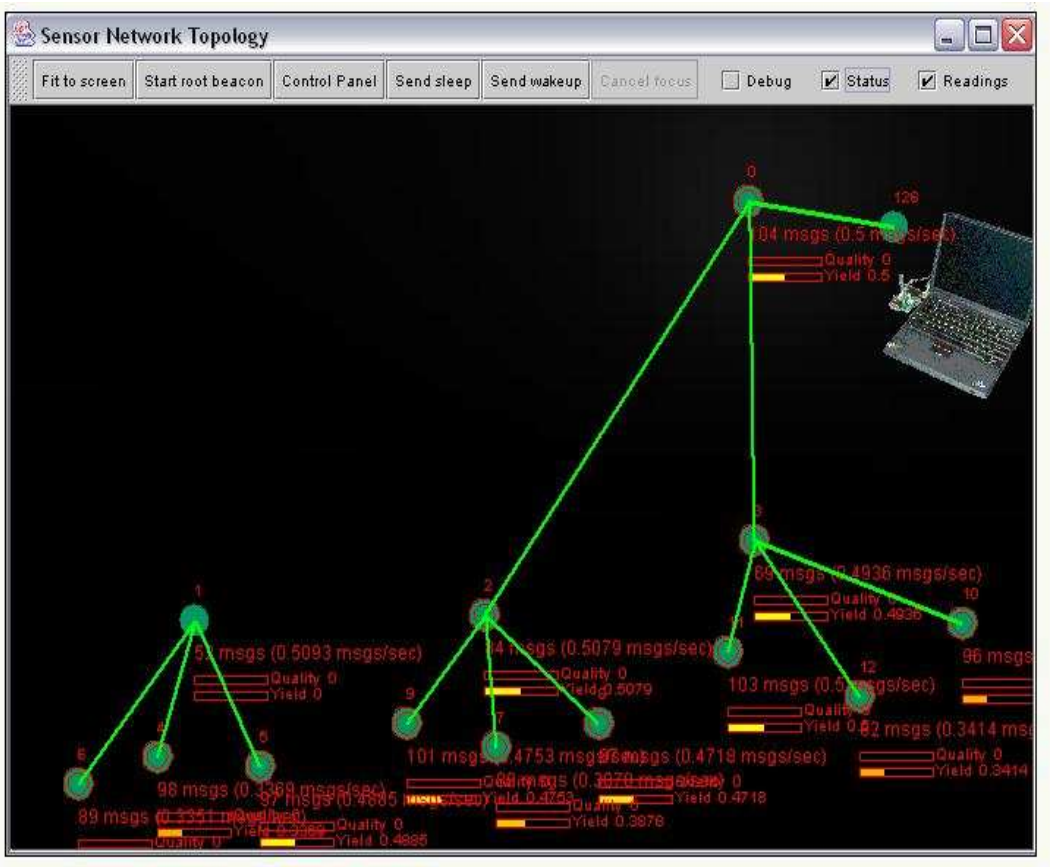

Figure 7.4. An Example screen shot of the network topology.

- Compile Surge, using make
- Run Surge: java net.tinyos.surge MainClass

7.2 Implementation

We have implemented all three proposed approaches, on Mica2 sensor nodes. Figure 7.5 is a snapshot of the dynamic topology of the sensor network. In this chapter we would like to compare their performances for different scenarios. Our goal is to find the most appropriate intrusion detection approach based on the characteristic of the system.

Figure 7.6 depicts the percentage of correct intrusion detection by all three approaches in a network consisting of 30% malicious nodes. The repeated game approach

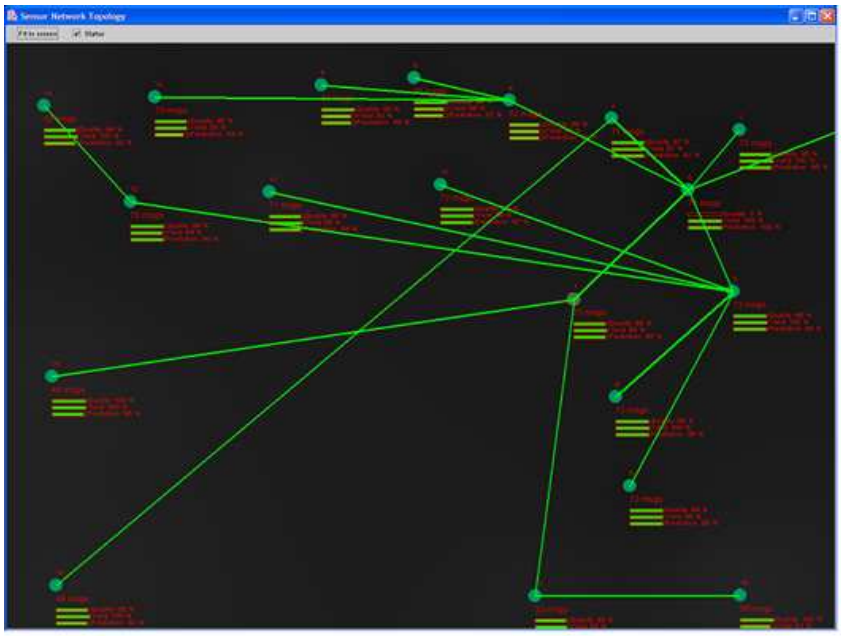

Figure 7.5. Sensor Network Snapshot.

has two times better chance than the UDSR approach, to detect malicious nodes and it performs 65% better than the SAR protocol. The reason for this superiority is that the IDS plays a repeated game and minimizes its loss over time.

Figure 7.7 shows that all three approaches lose more packets as more malicious nodes are introduced in the network. A smaller number of packets are dropped in SAR compared to UDSR. The reason behind this is that in SAR the equilibrium bid is a factor of the total number of nodes in the network. Hence as we increase the number of nodes, it still is able to perform better than UDSR. But the repeated game approach drops fewer packet compared to the other two. This is because in this approach, a rating mechanism is used for computing utility, and cooperation of nodes is not a factor.

Figure 7.8 illustrates the throughput of the three approaches. In this scenario, 60% of senor nodes are malicious. We notice that the repeated game approach has the best

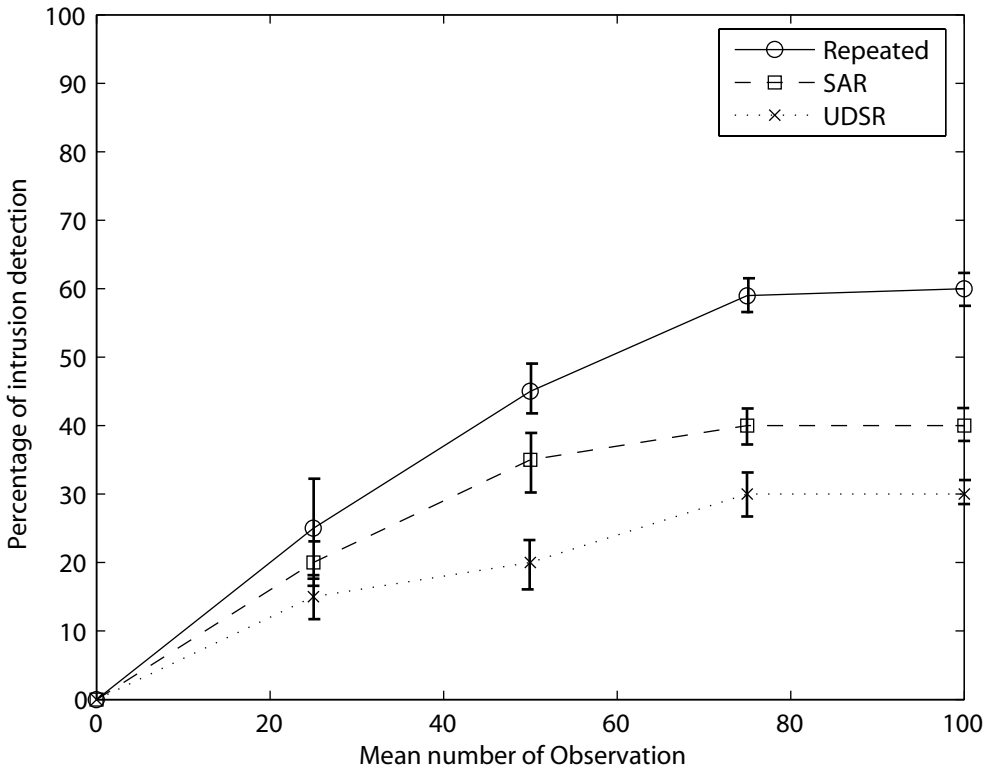

Figure 7.6. Success rate of Intrusion detection.

throughput but still if no malicious node is present in the network, it can lose packets 40% of the time. Improving this throughput is part of our future work.

Figure 7.9 shows the average number of packets dropped in the proposed approaches versus the total number of nodes present in the network. The reason that SAR and UDSR are losing more packets is that, when a packet on a path does not get to the destination, all nodes on that path get negative reputation regardless of being malicious or normal.

Figure 7.10 depicts a comparison between the detection rate of the three proposed approaches. Here 60% of nodes are acting maliciously. UDSR has a steady rate and this is due to the fact that the size of the network is not a major factor in computing utility in

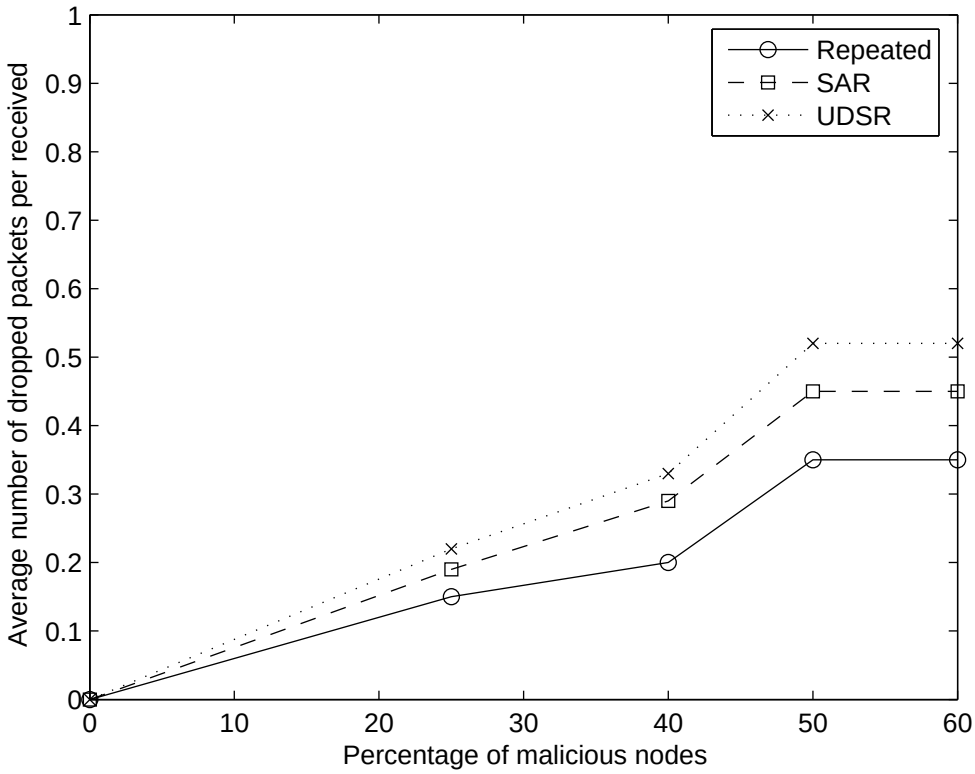

Figure 7.7. Dropped packets vs. malicious nodes.

this approach. We notice that this detection degrades when more than half of all nodes in the network are malicious.

In order to compare the performance of all three proposed approaches, regardless of the value of their weight parameters in utility functions, Figure 7.11 depicts the performance of the average detection rate of the three proposed approaches for different values of $\alpha = 0.2, 0.4, 0.6, 0.8$, where *alpha* is the reputation coefficient. This figure shows that the repeated game approach outperforms UDSR and SAR.

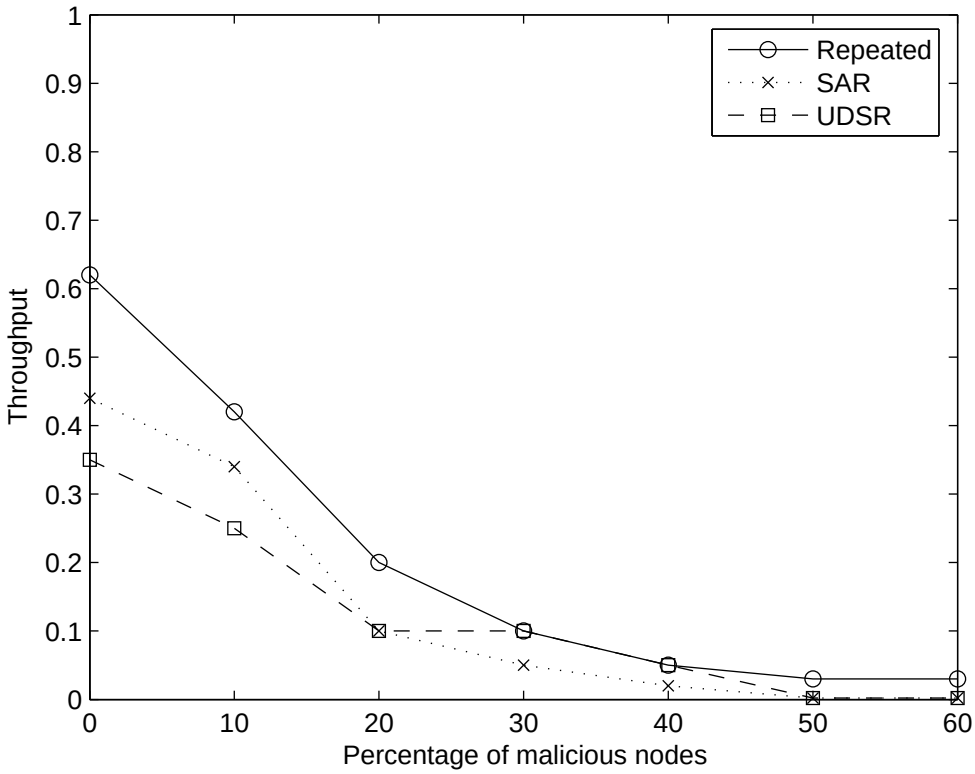

Figure 7.8. Throughput.

7.3 Summary

In the proposed SAR and UDSR, we assumed that every node of the network has some data traffic to be sent through some source route that is the result of the execution of some routing protocol (as an example the DSR protocol). By acting normally a node will forward one or more data packets for the requesting node, whereas by acting maliciously a node will not relay data packets on behalf of the requesting node. Instead of including an accurate description of energetic cost, topology information, we base our model on some basic economic modeling. Results show that incorporating a framework based on repeated game theory will have a higher success rate in detecting intrusions in wireless

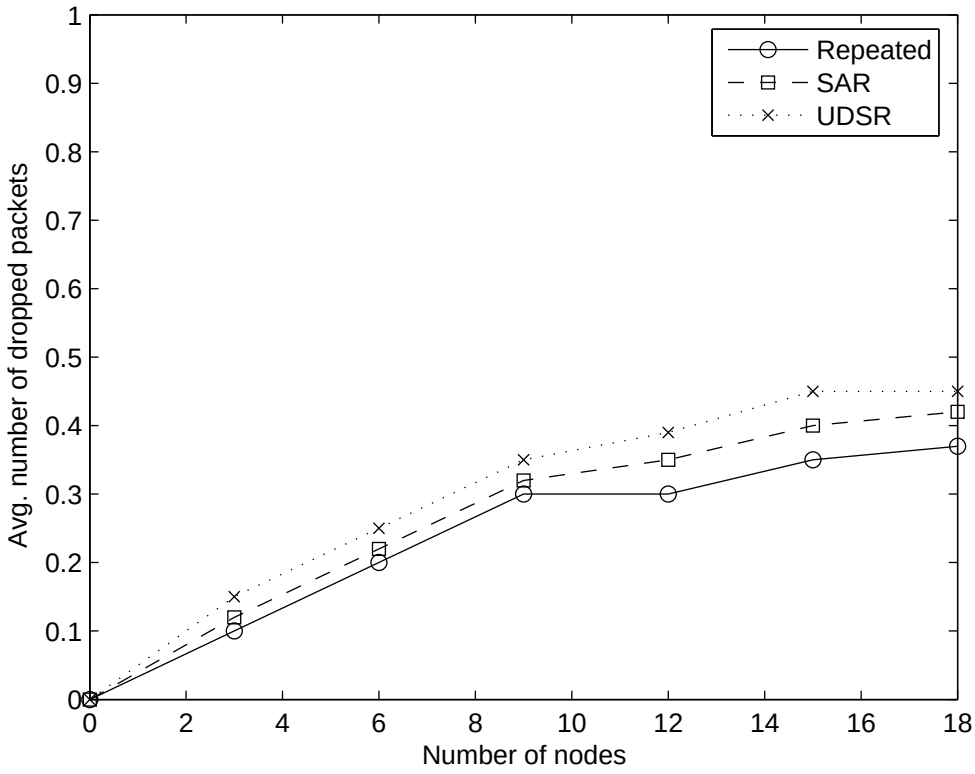

Figure 7.9. Dropped packet vs. total number of nodes.

sensor networks. This is due to the fact that the game strategies are conditioned on past histories.

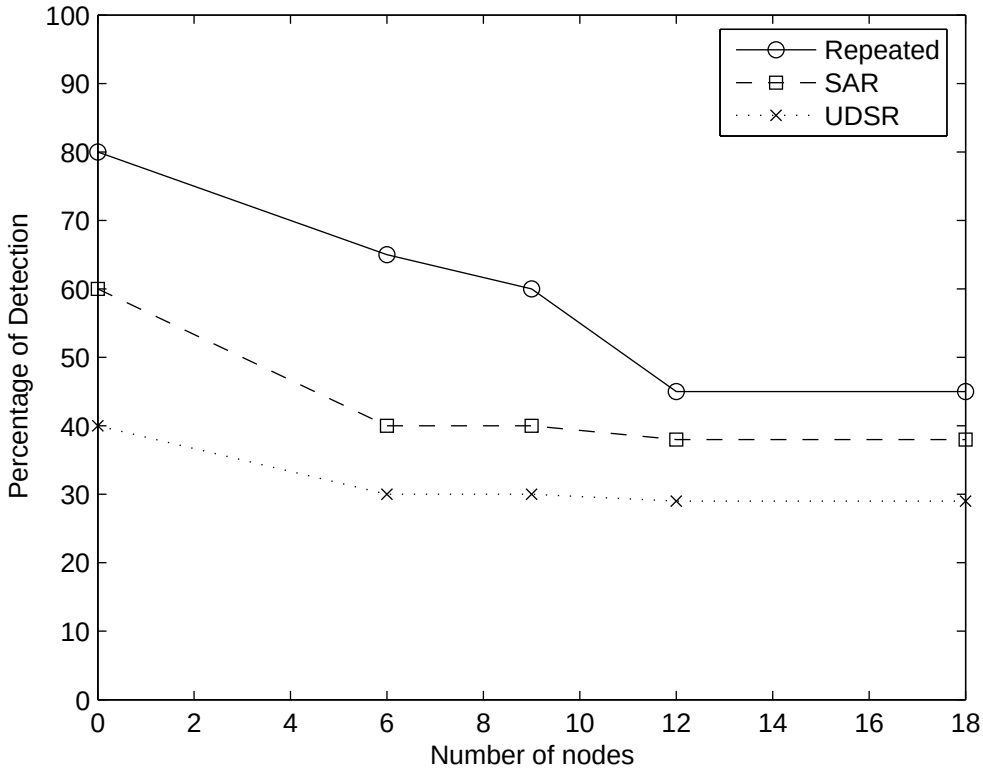

Figure 7.10. Detection rate.

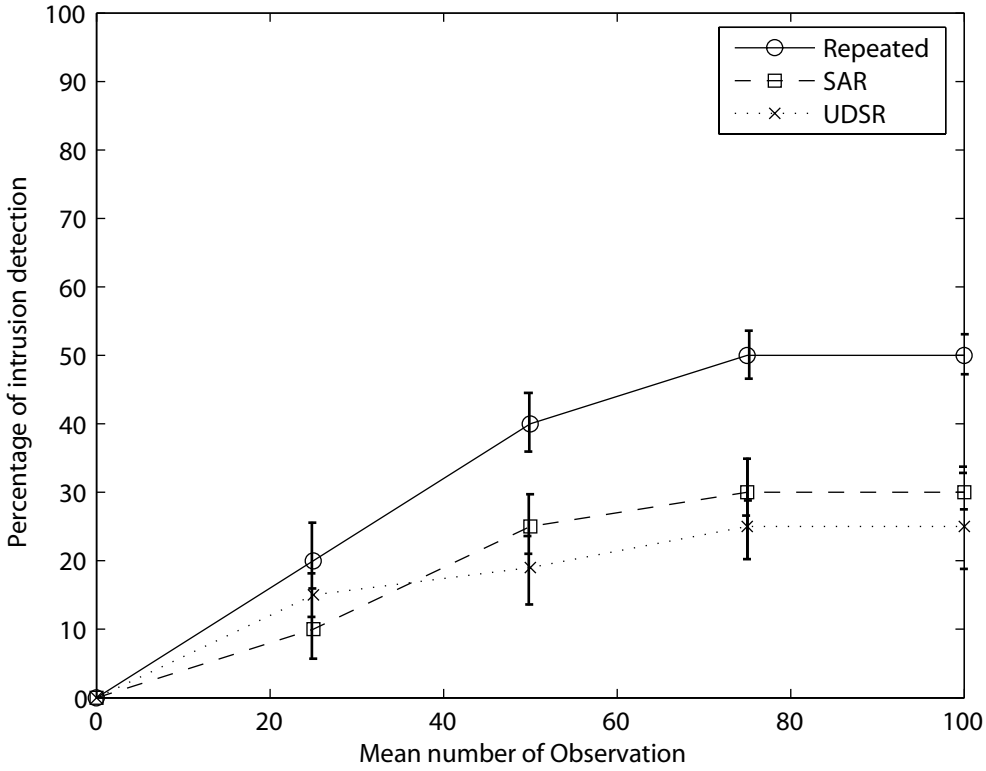

Figure 7.11. Average detection rate for $\alpha = 0.2, 0.4, 0.6, 0.8$ and $\beta = 0.5$.

CHAPTER 8

CONCLUSION AND FUTURE WORK

This dissertation addresses security enforcement, one of the most essential problems in wireless sensor networks. The selfish nature of sensor nodes in such networks shows that in order to maximize their benefits, nodes have no incentive to be honest. Therefore, when designing cooperation strategies, optimality criteria such as the Nash equilibrium must be considered.

After studying a non-cooperative two-player packet forwarding game, we concluded that a necessary condition for a strategy to be optimal from the network's point of view is that the most vulnerable node is the one with the highest amount of utility. After studying a wireless sensor network's routing protocol under a game theoretic framework, we concluded that nodes need to have reputation among other nodes in order to be entitled to use network resources in the future. The key to solve this problem is when nodes of a network use resources, they have to contribute to the network life in order to be entitled to use resources in the future.

Finally observing the behavior of an intrusion detector present in a wireless sensor network, use of a repeated game theory framework showed the need for cooperation enforcement schemes when selfish entities are present in the network. Infinite repetition can be the key for obtaining cooperative behavior in the stage games. IDS rates nodes through a monitoring mechanism, then these observations are collected and processed to evaluate a reputation of each node.

We ensured the finiteness of the repeated game payoffs by introducing *discount* of future payoffs relative to earlier payoffs. The results prove that the proposed framework are able to achieve the goal of detecting malicious nodes.

This game theoretic framework can be furthermore extended in several aspects:

- *Punishment:* A very interesting open questions is: What is an appropriate equilibrium notion for repeated games? That is how to come up with a defensible equilibrium refinement that motivates a reasonable class of strategies, such as punishment strategies for any malicious behavior.
- *Cooperation:* Another interesting direction that needs further research can be offered by the analysis of cooperation strategies and coalition formation algorithms through game theory. It would be worthwhile to propose a game theoretic framework for designing a generic cooperation scheme. All of our proposed protocols were based on the foundation of non-cooperative game; it is interesting to analyze how a subset of nodes/players can form coalition and play against the rest of the network/players as a cooperative game framework. We also need to verify how these coalitions can affect the reputations and how to make these coalitions stable.
- *Game Formulation:* Applying the proposed framework to other layers is also another possibility to extend this work. To achieve this, we first need to generalize the game formulation. Then for different protocols, customize the payoff function, utility function and thresholds for false positive and false negatives, according to the specifications and characteristics of the desired intrusion detection system.
- *Learning:* Adding intelligence in decision making part for each player will form possible extension to this work. In addition to having access to the history of game, each player will use a learning mechanism to predict the behavior of other players. A drawback of this approach is the limited memory of sensors but it is more realistic to happen in real life situations.

REFERENCES

[1] A. Agah, K. Basu, and S. K. Das. A game theory based approach for security in sensor networks. *International Performance Computing and Communications Conference (IPCCC)*, pages 259–263, 2004.

[2] A. Agah, K. Basu, and S. K. Das. Enforcing security for prevention of dos attack in wireless sensor networks using economical modeling. *Proceedings of 2nd IEEE International conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, 2005.

[3] A. Agah, K. Basu, and S. K. Das. Preventing dos attack in sensor and actor networks: A game theoretic approach. *IEEE International Conference on Communications (ICC)*, pages 3218–3222, 2005.

[4] A. Agah, K. Basu, and S. K. Das. Security enforcement in wireless sensor networks using non-cooperative game theory framework. *Pervasive and Mobile Computing Journal (PMC), Elsevier Publisher*, 2005.

[5] I. F. Akyldiz, , and I. H. Kasimogl. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2004.

[6] I. F. Akyldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 3(8):393–422, 2002.

[7] T. Başar and G. T. Olsder. *Dynamic Non cooperative Game Theory*. Society of Industrial and Applied Mathematic, 1999.

[8] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. Secure pebblenet. *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 156–163, 2001.

[9] L. Blazevic, L. Buttyaan, S. Capkun, S. Giordano, J. P. Hubaux, and J. LeBoudec. Self-organization in mobile ad hoc networks: the approach of terminodes. *IEEE Communication Magazine*, 39(6):161–174, 2001.

[10] G. E. Bolton and A. Ockenfels. Erc a theory of equity, reciprocity, and competition. *The American Economic Review*, 90, 2000.

[11] Cross Bow. *http://www.xbow.com.*

[12] S. Buchegger and J. L. Boudec. Nodes bearing grudges: toward routing security, fairness and robustness in mobile ad hoc networks. In *The 10th Euronicro Workshop on parallel Distributed and Network based Proceedings*, pages 403–410, Canary Islands, Spain, January 2002.

[13] S. Buchegger and J. L. Boudec. Performance analysis of the confidant protocol cooperation of nodes-fairness in dynamic ad-hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 226–236, 2002.

[14] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann. Scalable coordination for wireless sensor networks: self-configuring localization systems. In *International Symposium on Communication Theory and Applications (ISCTA)*, Ambleside, UK, July 2001.

[15] L. Buttyaan and J. P. Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. In *Technical Report DSC/2001/001, Department of Communication Systems*, Swiss Federal Institute of Technology, 2001.

[16] L. Buttyaan and J. P. Hubaux. Report on a working session on security in wireless ad hoc network. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 6(3):7–9, 2002.

[17] H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer*, 36(10):103–105, 2003.

[18] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy(SP'03)*, 2003.

[19] K. Chen and K. Nahrstedt. ipass: an incentive compatible auction scheme to enable packet forwarding service in manet. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, Tokyo,Japan, March 2004.

[20] C. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenge. *IEEE special issue on sensor networks and application*, 91(8):1247–1256, 2003.

[21] J. Deng, R. Han, and S. Mishra. Insens:intrusion-tolerant routing in wireless sensor networks. In *Technical Report TR CU-CS-939-02, Dept. of Computer Science, University of Colorado*, Colorado, 2002.

[22] S. Doshi, S. Bhandare, and T. Brown. An on-demand minimum energy routing protocol for a wireless ad hoc networks,. *ACM Mobile Computing and Communications Review*, 6(3), July 2002.

[23] M. B. Dydensborg. Connection oriented sensor networks. *Ph.D. Dissertation, University of Copenhagen*, 2004.

[24] K. Fall and K. Varadha. ns notes and documentation. *The VINT project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Feb 2000, http://www.isi.edu/nsnam/ns.*

[25] L. Feeney. A taxonomy for routing protocols in mobile ad hoc networks. In *Technical Report T99/07*, Swedish Institute of Computer Science, October 1999.

[26] M. Felegyhazi, L. Buttyan, and J. P. Hubaux. Equilibrium analysis of packet forwarding strategies in wireless ad hoc networks the static case. In *Personal Wireless Communications (PWC '03)*, Venice, Italy, September 2003.

[27] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless micro sensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002.

[28] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. In *Proceedings of the Hawaii International Conference System Sciences*, Hawaii, January 2002.

[29] F. Hu and N. K. sharma. Security considerations in ad hoc sensor networks. *Ad Hoc Networks (Elsevier)*, 3(1):12–23, August 2005.

[30] Y. C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Technical Report TR01-383*, Department of Computer Science, Rice University, 2001.

[31] Y. C. Hu, A. Perrig, and D. B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Technical Report TR01-384*, Department of Computer Science, Rice University, June 2002.

[32] D. B. Johnson and D. A. Malt. The dynamic source routing protocol for mobile ad hoc networks. In *Mobile Ad Hoc Network (MANET) Working Group (IETF)*, October 1999.

[33] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications (SPNA)*, 2003.

[34] P. Klemperer. *Auctions: Theory and Practice*. Princeton University Press, 2004.

[35] M. Kodialam and T. V. Lakshman. Detecting network intrusions via sampling: A game theoretic approach. In *IEEE Conference on Computer Communication (INFOCOM)*, 2003.

[36] V. Krishna. *Auction Theory*. Academic Press, 2002.

[37] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, August 2000.

[38] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Communications and Multimedia Security Conference*, 2002.

[39] P. Michiardi and R. Molva. Game theoretic analysis of security in mobile ad hoc networks. In *Institute Eurecom*, France, 2002.

[40] P. Michiardi and R. Molva. Prevention of denial of service attack and selfishness in mobile ad hoc networks. In *Research Report RR-02-063, Institute Eurécom*, France, 2002.

[41] P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *European Wireless 2002: Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Florance, Italy, February 2002.

[42] J. F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, September 1951.

[43] ns2. http://www.ns2.com.

[44] G. Owen. *Game Theory.* Academic Press, New York, NY, 2001.

[45] S. Patil. Performance measurement of ad-hoc sensor networks under threats. In *Wireless Communications and Networking Conference (WCNC)*, 2004.

[46] A. Perrig, R. Canetti, J. Tygar, and D. Song. Efficient authentication and signing for multicast. In *Network and Distributed System Security Symposium (NDSS)*, pages 35–46, February 2001.

[47] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 189–199, Italy, July 2001.

[48] R. Pietro, L. V. Mancini, and S. Jajodia. Secure selective exclusion in ad-hoc wireless networks. In *Security in Information Society: Visions and Perspectives*, pages 423–434, Boston, 2002. Kluwer.

[49] T. S. Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall, 2002.

[50] J. Ratliff. *http://www.virtualperfection.com/gametheory*.

[51] S. M. Ross. *Introduction to Probability Models*. Academic Press, 1997.

[52] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A secure routing protocol for ad hoc networks. In *International Conference on Network Protocol (ICNP)*, pages 78–87, Paris, France, November 2002.

[53] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *MACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 272–286, Italy, July 2001.

[54] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *IEEE Conference on Computer Communication (INFOCOM)*, 2003.

[55] L. Subramanian and R. H. Katz. An architecture for building self configurable systems. In *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing*, MA, August 2000.

[56] TinyOS. *http://www.tinyos.net*.

[57] J. Undercoffer, S. Avancha, A. Joshi, and J. Pinkston. Security for sensor networks. *IEEE Computer Networks: The International Journal of Computer and Telecommunications Networking, Special issue: Wireless sensor networks*, 43(4):421–435, 2003.

[58] E. Wolfstetter. Auctions: An introduction. In *Technical Report*, Humboldt University, 1994.

[59] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. In *IEEE COmputer*, pages 54–62, 2002.

[60] F. Ye, A. Chen, S. Lu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. *Proceedings of the 10th International Conference on Computer Communications and Networks (ICCCN)*, pages 304–309, 2001.

BIOGRAPHICAL STATEMENT

The author received her doctorate in Computer Science and Engineering from The University of Texas at Arlington in December 2005. Prior to that she received her Master degrees from Azad University, Tehran, Iran and Kansas State University in Manhattan, Kansas. Her research interests include security in wireless sensor networks and mobile ad hoc networks. She is a member of IEEE and society of Woman Engineers (SWE).