To the Graduate Council:

I am submitting herewith a thesis written by Alma Cemerlic entitled "Fine-Grained Reputation-Based Routing in Wireless Ad Hoc Networks." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

_____

Li Yang, Ph.D., Major Professor

We have read this thesis and
recommend its acceptance:

_____

Joseph M Kizza, Ph.D.

_____

Mina Sartipi, Ph.D.

Acceptance for the Council:

_____

Stephanie L. Bellar, Ph.D.

Interim Dean of the Graduate School

FINE-GRAINED REPUTATION-BASED ROUTING IN WIRELESS AD HOC NETWORKS

A thesis presented for
The Master of Science Degree

The University of Tennessee at Chattanooga
2008

Alma Cemerlic

DEDICATION

This thesis is dedicated to my best friends, Chibi and Galen.

ABSTRACT

Ad hoc networks are very helpful in situations when no fixed network infrastructure is available. They are especially important in emergency situations such as natural disasters and military conflicts. Most developed wireless ad hoc routing protocols are designed to discover and maintain an active path from source to destination with an assumption that every node is friendly and cooperative. However, it is possible that the participating nodes are selfish or malicious. A mechanism to evaluate reputation for each node is essential for the reliability and security of routing protocol in ad hoc networks.

We propose a fine-grained reputation system for wireless ad hoc routing protocols based on constantly monitored and updated first and second -hand reputation information. The nodes in the network monitor their neighbors and obtain first-hand information based on the perceived behavior. Second-hand information is obtained by sharing first-hand information with nodes' neighbors. Our system is able to distinguish between selfish and malicious nodes and take appropriate actions in either case. We employ the moving-window mechanism which enables us to assign more weight to more recent observations and adjust responsiveness of our reputation system to changes in nodes' behavior.

We show that our fine-grained reputation system is able to improve both reliability and security of an ad hoc network when compared to a reputation system that does not distinguish between selfish and malicious nodes.

TABLE OF CONTENTS

Chapter I: Introduction 1
1.1 Objective of Study 4
1.2 Definitions of Terms 5
1.3 Characteristics of MAC Layer 6
1.4 Node Behavior and Classification 7
1.5 Reputation 8
1.6 Trust 10
1.7 Reputation Evaluation 11
1.8 Node Misbehavior Models 12
1.9 Performance Metrics 14
Chapter II: Literature Review 17
Chapter III: Methodology 28
3.1 Fine-Grained Reputation System 28
3.1.1 First-Hand Reputation 29
3.1.2 Second-Hand Reputation 33
3.1.3 Trust 35
3.1.4 Total Reputation Based on First and Second-Hand Reputation 37
3.1.5 Moving Window Mechanism 39
3.2 Integrating Reputation into Wireless Ad Hoc Routing 41
3.2.1 Wireless Ad Hoc Routing Without Reputation 41
3.2.2 Integrating Reputation into Wireless Ad Hoc Routing 43
Chapter IV: Analysis of Results 49
4.1 Introduction 49
4.2 Simulation Results 52
4.2.1 Effective Throughput 53
4.2.2 Percentage of Injected Packets 54
4.2.3 Effective Throughput and Percentage of Injected Packets with Moving Window Mechanism 56
4.2.4 Connectivity 59
4.2.5 Discussion of Results 62
Chapter V: Conclusion 65
References 67

LIST OF FIGURES

Figure *Page*

1.1 Wireless network in infrastructure mode vs. wireless network in ad hoc mode 2
1.2 Neighbors and neighborhood 6
1.3 Classification of nodes based on their behavior 7
1.4 A simple observation window model 9
1.5 Reputation evaluation 12
3.1 Node A is not able to can not overhear node B forwarding P1 to D because P1 collides with P2 sent by node C (Laniepce 2006) 34
3.2 Second-hand reputation mechanism allows node A to find out about node C's behavior, even though C is not A's neighbor 44
3.3 Redemption of a malicious node reflected through the change in the friendly reputation component 48
4.1 Effective throughput - comparison for the fine-grained <f,s,m> and beta <g,b> reputation systems 53
4.2 Percentage of injected packets - comparison for the fine-grained <f,s,m> and beta <g,b> reputation systems 55
4.3 Effective throughput - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 20% node misbehavior. 57
4.4 Percentage of injected packets - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 20% node misbehavior 57

4.5 Effective throughput - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 80% node misbehavior 58
4.6 Percentage of injected packets - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 80% node misbehavior 59
4.7 Total connectivity of the network - comparison for the fine-grained <f,s,m> and beta <g,b> reputation systems over the cases in the 40% simulation set 60

LIST OF TABLES

Table *Page*

3.1 Reputation calculation based on the observations node A makes about its neighbor, node B..... 33
3.2 Trust calculation for node B based on the deviation tests on node A as it receives second-hand reputation information from node B..... 37
3.3 Illustration of early detection by moving window mechanism. 40

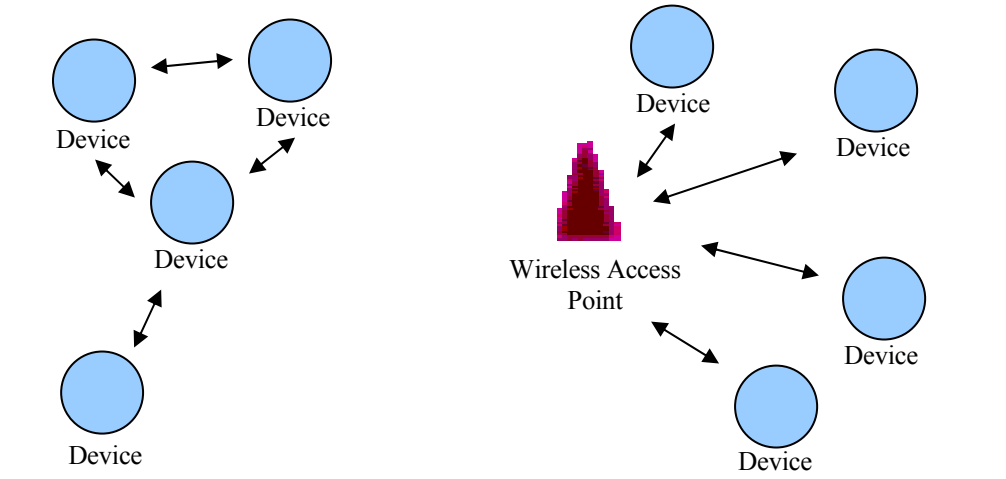
Chapter I

INTRODUCTION

A wireless ad hoc network is a decentralized wireless local area computer network that allows wireless devices to directly communicate with each other. Ad hoc devices are part of a network for the duration of a communication session, or as in the case of mobile devices, for as long as it is in the range of a network. Ad hoc is a Latin phrase meaning "for this purpose" (Traupman 2007). As such, ad hoc computer networks can be described as temporary ones. These networks are found to be suitable for emergency situations such as natural disasters or military conflicts because of their decentralized nature, minimal configuration, and ability to be quickly deployed.

Ad hoc networks are sometimes referred to as a "self-organized communication system" (Buchegger 2008). They are characterized by the absence of infrastructure and organized according to the peer-to-peer (P2P) principle (Buchegger 2008): all the nodes participating in a network have equivalent responsibilities (they are all peers). Another type of network, an infrastructure mode wireless network, relies on the power of access points to cover a wide communication area and relay data between the devices in the network. Conversely, ad hoc networks have no designated routers or access points, routing decisions are made depending on current connection availability, and all nodes in a network have equal responsibility to forward packets for all the other networked nodes (Figure 1.1). The communication range of an ad hoc

network depends on the power available in the participating devices. If a device tries to send data to another device not within its communication range, the data must be forwarded through multiple nodes in order to reach its destination (Rackley 2007).



(a) Wireless Network in Ad Hoc Mode

(b) Wireless Network in Infrastructure Mode

Figure 1.1: Wireless network in infrastructure mode vs. wireless network in ad hoc mode.

Most wireless ad hoc routing protocols are concerned only with maintaining the connectivity among the nodes and assume that participating nodes will cooperate and perform their routing duties for other nodes. The sender of the data has no influence over what path the data takes on the way to the destination. In the presence of non-cooperative nodes, the performance of such an ad hoc network will degrade, until the network is ultimately rendered useless. Because of the characteristics presented above, self-organized communication systems, including ad-hoc networks,

suffer from several issues: lack of cooperation, malicious attacks, and random failures of networked nodes (Buchegger 2008).

The experiment results presented in (Buchegger 2002a) suggest that a 50-node network using the defenseless Dynamic Source Routing (DSR) ad hoc routing protocol will lose about 70% of packets when one third of the nodes are misbehaving.

It is reasonable to expect users of a self-organized communication system to be concerned primarily about their own benefit, thus cooperation and fairness cannot be guaranteed. Since ad hoc networks are completely dependent on willingness of their participants to forward packets for each other, nodes that exhibit selfish behavior and do not forward packets for their peers can lead to diminishing quality of service or a complete collapse of network connectivity. Selfish behavior can be a consequence of a node's physical properties (loss of battery power), a purposeful attempt to save its own resources, or a random failure.

Wireless networks are inherently less secure than the wired networks because the signal (the exchanged packets) is broadcasted in the air allowing anyone to access it. Wireless ad hoc networks are especially vulnerable to malicious behavior in which nodes inject or misroute packets they are supposed to forward for other nodes.

Various types of reputation systems have emerged in recent years as an attempt to address these problems. The need for the reputation systems is obvious if we consider the established security models. In the case of standard wireless networks, it

is possible to talk about traditional information security, commonly defined as "the preservation of confidentiality, integrity and availability of information (CIA model)"(Jøsang 2007) . But the CIA model is not adequate for protection against deceitful service providers (Jøsang 2007). Rasmussen & Jansson (Jøsang 2007) first described "soft security mechanisms". The purpose of these mechanisms is to stimulate ethical behavior and integrity of members in collaborative environments such as ad hoc networks, where the ethical norms are not fixed but are rather dynamically defined by its participants. Soft security mechanisms are able to recognize and sanction intolerable behavior and reward members who follow the norms.

Reputation systems are classified as a type of soft security mechanisms (Jøsang 2007). When applied to ad hoc networks, reputation systems require every node to keep track of their peers' behavior. This information is then used to determine which peers should be avoided and which can be cooperated with (Buchegger 2008).

1.1 Objective of Study

In this project, we incorporate a fine-grained reputation system into a wireless ad hoc network with the aim to protect the network from misbehaving (selfish and malicious) nodes.

The main objectives of this system are to improve connectivity and cooperation among the nodes in the network, and to minimize interaction with malicious nodes which present a security threat for the entire network. Our system is

based on the Dirichlet distribution - a multinomial probability distribution. We show the benefits of the granular approach as opposed to the reputation systems based on the Binomial distribution.

1.2 Definitions of Terms

In a wireless ad hoc network all the participating nodes have equal responsibility to forward packets for other nodes in the network, thus each node acts as a router for other nodes. All nodes that are one hop away from a selected node A are said to be A's *neighbors* and compose A's *neighborhood* (Figure 1.2). In a network with infrastructure, a *hop* is defined to be "the trip a data packet takes form one router to another in the network" (SearchCIO-Midmarket.com 2008). Since in an ad hoc network every node is a router, a hop is any intermediate node on the path that a data packet takes from its source to its destination. Each node in the fine-grained reputation system is responsible for *observing* how its neighbors are handling the packets they need to forward for other nodes. That is, whether they correctly forward, drop, or maliciously modify each packet they receive. *Reputation* information indicates how well a node behaves as participant in the ad hoc network, i.e. whether it is correctly forwarding packets for other nodes. *Trust* indicates how well a node behaves as a participant in the reputation system, i.e. whether a node accurately reports observations it made about its neighbors.

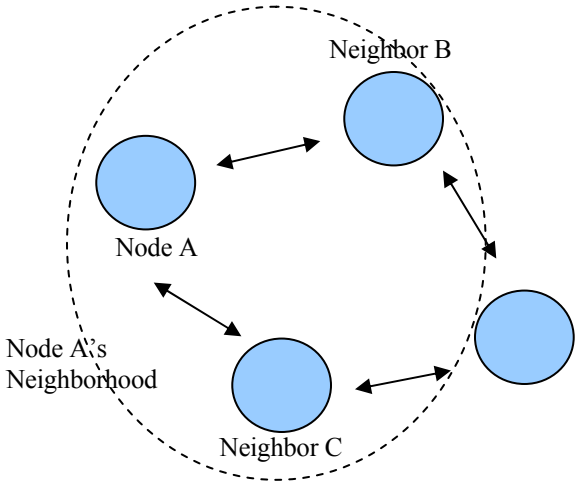


Figure 1.2: Neighbors and neighborhood.

1.3 Characteristics of MAC Layer

We assume that nodes are capable of bidirectional communication on every link. This assumption means that it is possible to send data from node A to node B or from node B to node A at any point in time. Many wireless Medium Access Control (MAC) layer protocols, including IEEE 802.11, require bidirectional communication for reliable transmission. We also assume that network interfaces on the nodes support promiscuous mode operation. Promiscuous mode "means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A" (Marti 2000).

1.4 Node Behavior and Classification

Based on observations about their forwarding behavior, we classify wireless nodes into three categories: friendly, selfish, and malicious (Figure 1.3). Different from classifying nodes into good and bad categories, as done in reputation systems in CONFIDANT (Buchegger 2002a), SORI (Wang 2004b), CORE (Michiardi 2002a), and SAFE (Rebahi 2005), our approach allows for more precise categorization and finer granularity.

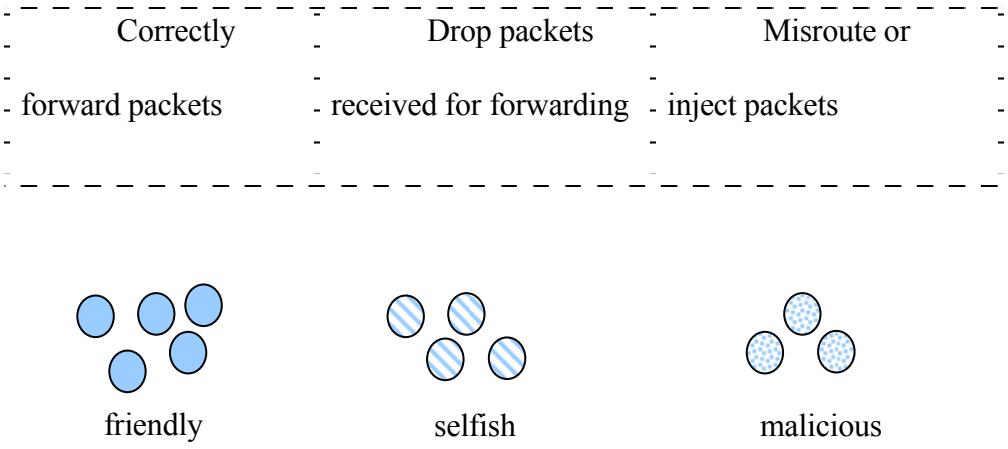


Figure 1.3: Classification of nodes based on their behavior.

Friendly nodes correctly forward packets they receive from other nodes. These packets are delivered to the destination with preserved data integrity. Friendly behavior is the expected behavior in the ad hoc networks. Selfish nodes drop packets they receive for forwarding from other nodes, but they expect the other nodes to forward packets that the selfish nodes send. Selfish behavior may be a result of a node's physical properties (loss of battery power, overload with forwarding requests),

purposeful attempt to save its own resources (battery and computing resources), or a random failure. This class of misbehaving nodes lowers reliability of the network. Malicious nodes misroute, modify, or inject packets (making them a part of a different data transfer). These nodes are primarily interested in attacking and damaging the network. Malicious nodes lower security and integrity of the network traffic.

For various reasons, a node's behavior can change. For example, if a node starts losing its battery power, it may begin to behave selfishly and drop packets. Even though this node was previously categorized as friendly, our reputation system will react to its recent selfish behavior and reclassify it as selfish.

1.5 Reputation

Each node assigns a total reputation value to other nodes in the network. This value describes the node's belief about a neighbor's behavior, i.e. whether the node will behave friendly, selfishly, or maliciously. The total reputation value is evaluated from both first-hand and second-hand reputation.

The first-hand reputation value is based on direct observations in promiscuous mode that node A makes about the behavior of node B. We denote it as $F_{A,B}$. Node A observes node B's behavior over some predefined time interval before it calculates $F_{A,B}$. We call this interval a *window*. The windows allow us to limit the amount of historic information we use when calculating the current first-hand reputation value (Figure 1.4). We will later discuss the concept of windows in more detail.

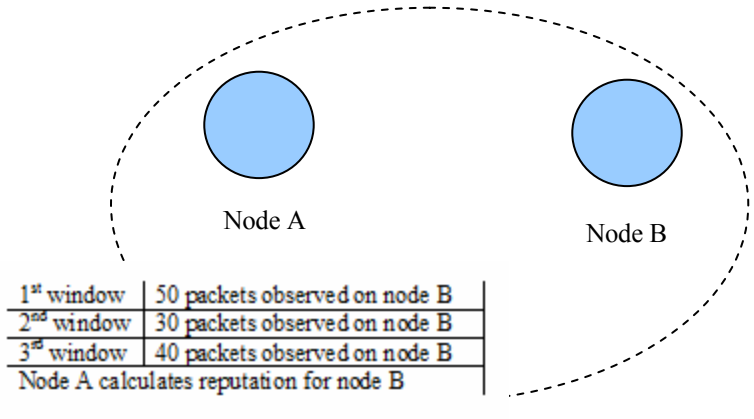


Figure 1.4: A simple observation window model.

The second-hand reputation value is obtained by sharing the first-hand reputation value with neighbors. All nodes are required to periodically broadcast their first-hand reputation values to their neighborhood. When node A receives a report form its neighbor C about node B's behavior (denoted as $F_{C,B}$), it will merge the two values, the first-hand reputation $F_{A,B}$ and the second-hand reputation $F_{C,B}$, in order to calculate the total reputation value for node B. The total reputation value depends on several factors as we describe later. Note that we only allow propagation of first-hand reputation. Buchegger et al. explain that "passing on information received from others, as opposed to direct observation turns out to not only offer no gain in reputation accuracy or speed, but also to introduce vulnerabilities by creating a spiral of self-reinforcing information" (Buchegger 2008).

1.6 Trust

The purpose of the trust mechanism is to protect the reputation system from incorrect second-hand reputation reports. The trust value indicates how accurately a node reports reputation information to other nodes. Each node is required to calculate and record trust values for its neighbors when it receives second-hand reputation values from them. If node C is reporting to node A about node B, trust reflects node A's opinion about trustworthiness of the report considering that it came from node C. Every time a node receives second-hand reputation from another node, it needs to decide how to consider this information. Since the node regards its knowledge to be the most accurate, it will compare the received information with its own. The new information will only be accepted if it fits within a specified acceptable deviation. Using such test to evaluate each piece of information is "a more fine-grained and adaptive approach than only considering the rater reputation of the node providing the reputation information" (Buchegger 2008). Following the example of interaction between nodes A and C, every time C's second-hand reputation passes the deviation test on node A, the trust that A has for C's reports will increase. On the other hand, if the second-hand reputation does not pass the test, the trust for C on node A will decrease. Finally, when node A merges $F_{A,B}$ and $F_{C,B}$ as previously discussed, node A discounts $F_{C,B}$ by a certain amount depending on its distrust in the C's reports. We will later discuss the trust mechanism and related calculations in more detail.

1.7 Reputation Evaluation

As shown in the Figure 1.5, reputation evaluation in our fine-grained reputation system includes six steps: (1) Node A calculates first-hand reputation information for node B ($F_{A,B}$) based on its observations of node B's behavior over some number of time intervals (windows); (2) Node A receives second-hand information about node B's behavior from nodes C and D ($F_{C,B}$ and $F_{D,B}$); (3) Node A performs a deviation test for the reports $F_{C,B}$ and $F_{D,B}$ and examines the current trust values for nodes C and D (ω_{AC} and ω_{AD}); (4) If the current trust values ω_{AC} and ω_{AD} indicate the nodes are trustworthy (the trust threshold test) and if the reports pass the deviation test, the reports are accepted. If either of the condition fails, they are discarded; (5) To calculate the total reputation for node B, node A merges the first-hand reputation information with the accepted second-hand reputation values and incorporates the result with the previous total reputation; (6) Based on the outcome of the deviation test, node A updates trust values for nodes C and D (ω_{AC} and ω_{AD}). If a report passes the deviation test, the trust value is increased. If it fails the test, the trust value is decreased. Later, we will describe each of these steps in grater detail.

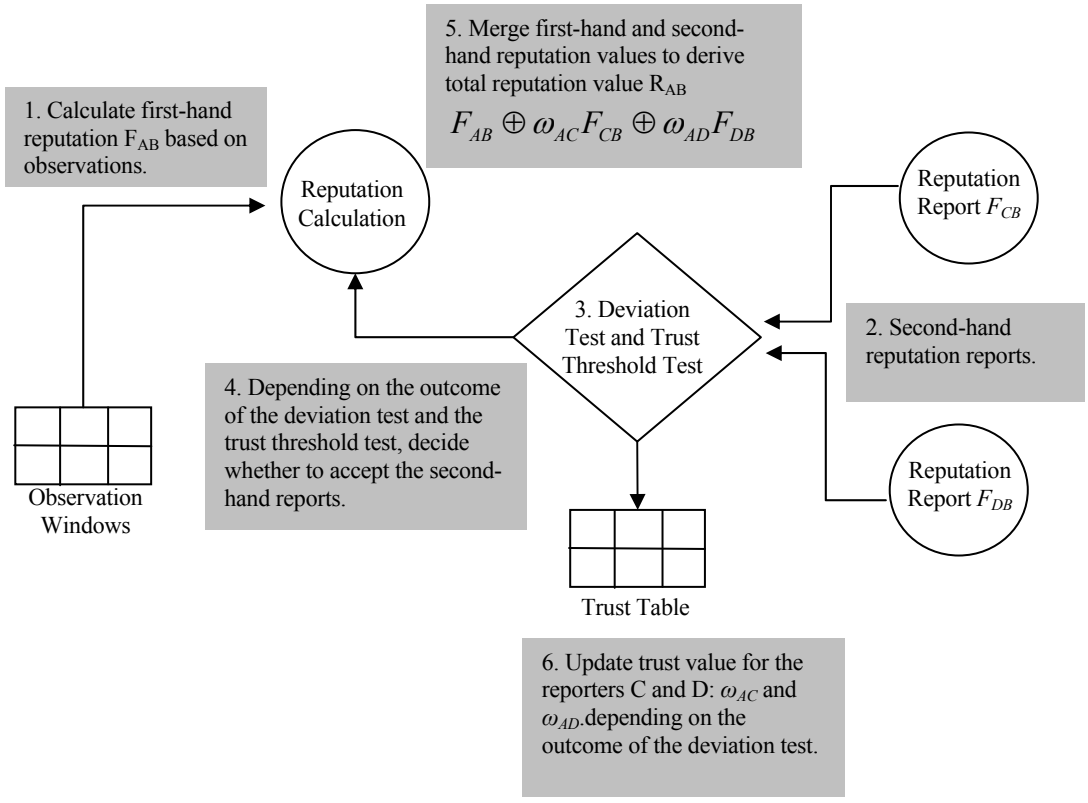


Figure 1.5: Reputation evaluation.

1.8 Node Misbehavior Models

We now precisely define the adversaries' abilities considered by our system. We define adversaries as nodes which behave in such way that they are degrading security, integrity, or reliability of the ad hoc network. The adversaries are grouped according to consistency of misbehavior and whether they are selfish or malicious. We assume that malicious nodes inject packets when they are supposed to forward packets for other nodes and thus lower security and integrity of the network. Malicious behavior is an intentional attempt to hurt the network. We assume that selfish nodes

only drop packets they are supposed to forward for other nodes. This class of adversaries lowers reliability of the network. We attribute selfish behavior to several factors: loss of battery power or overload by forwarding requests, selfish behavior in order to save resources, or a random failure of software or hardware on the node. Our fine-grained reputation system is able to detect and defend against the following four attack models:

Model 1 - consistent individual malicious nodes:

A malicious node *always injects* packets when it is in a selected routing path. Our reputation system is able to detect and avoid such nodes.

Model 2 – occasional individual malicious nodes:

A malicious node *occasionally injects* packets when it is in a selected routing path. The moving window mechanism allows us to adjust the responsiveness of the fine-grained reputation system to changes in nodes' behavior. If the window size is small, the system will quickly recognize any occasional changes in behavior.

Model 3 – consistent individual selfish nodes:

A selfish node *always drops* packets when it is in a selected routing path. Our system is able to recognize and avoid such nodes.

Model 4 – occasional individual selfish nodes:

A selfish node *occasionally drops* packets when it is in a selected routing path. This model is handled in the same manner as Model 2.

While this work does not focus on the function and effectiveness of the trust system, it is important to mention that the trust system enables the fine-grained

reputation system to also recognize and avoid nodes that falsely report reputation values of other nodes (lying nodes). Propagation of false low reputation values will result in low reputation values for good nodes. This may lead to a denial of service attack, where the falsely accused nodes are punished and denied network services, while at the same time the entire network suffers from decreased connectivity since the falsely accused nodes will be avoided in routing paths. If false high reputation is propagated, the network may utilize malicious and selfish nodes and both security and reliability are going to decrease. Using the trust deviation test, the nodes are able to calculate the difference *d* between the reported and the current reputation value and decide how to consider the reported value. Mundinger et al. show that for a small *d*, the system will be more robust against lying nodes. However, "smaller *d* means less use of second-hand information"(Mundinger 2005), and the reputation system may take longer to converge to the true reputation value. Thus the trust system requires consideration of a tradeoff between speed and accuracy of the reputation system.

1.9 Performance Metrics

In order to evaluate the performance of our model, we consider the following metrics: effective throughput, connectivity, and the percentage of injected or modified packets introduced in the system by malicious nodes.

Effective throughput is a measurement of the effective aggregate bandwidth of our system. It is the ratio of the bandwidth occupied by the legitimate (not modified or

injected) packets that reached their destination and the bandwidth carrying all the packets that reached their destination. We express this mathematically as:

$$ET_N = \frac{\left(\frac{P_R - P_I}{P_S}\right)}{\left(\frac{P_R - P_I}{P_S} + \frac{P_I}{P_S}\right)},$$

where ET_N is the effective throughput for a network during some defined time interval, and P_R, P_I and P_S are, respectively, the total number of received, injected, and sent packets during this time interval.

We define connectivity as a function of the number and a cost of possible paths in the network connecting all available nodes capable of forwarding packets. We calculate connectivity according to the following formula:

$$C_N = \sum_{A=0}^{n} \sum_{B=0}^{n} \frac{1}{S_{AB}},$$

where C_N is the total connectivity of a network at some point in time, n is the number of nodes in the network, and S_{AB} is the reputation cost of reaching node B from node A as observed on node A and given that A and B are neighbors. If node A and B are not neighbors, then the connectivity between them is zero. Decreased connectivity among nodes may lead to partitioning of the network, a scenario in which two nodes engaged in a communication session suddenly become physically unreachable from each other because no nodes are capable of forwarding packets between them. Our goal is to

obtain higher connectivity than the reputation systems that classify nodes only as good and malicious and react to the misbehavior by disconnecting the misbehaving nodes from the network.

Finally, the percentage of injected or modified packets indicates how often malicious nodes receive an opportunity to forward data packets for other nodes. The fine-grained reputation system distinguishes malicious from selfish behavior and avoids malicious nodes when making routing decisions. Our goal is to decrease the number of injected or modified packets present in the system when compared to the reputation system that classifies nodes as good and bad.

Chapter II

LITERATURE REVIEW

Securing ad hoc routing protocols against selfish and malicious nodes has been the subject of intense research efforts over the past few years. Here, we will present several solutions for encouraging node cooperation. The general idea behind these solutions is that all nodes keep track of behavior of their peers and use these observations to calculate a reputation value for each individual peer. This reputation value acts as an estimate of the quality with which each peer performs its networking functions. When making routing decisions, nodes will avoid peers with low reputation values. The goal of the reputation system is to ensure quality of network services by either encouraging or forcing all nodes to maintain high reputation values. Wang et al. (Wang 2004b) classify reputation systems that use incentives to encourage cooperation are classified as motivation based. On the contrary, detection based reputation systems exclude misbehaving nodes from the network soon as their reputation falls below some threshold. Regardless of the strategy taken against misbehaving nodes, all reputation systems should be fully decentralized and scalable in order to fit the ad hoc architecture.

Mahmoud et al. in (Mahmoud 2005) propose a solution, Reputed-ARAN protocol, in which each node promiscuously monitors its first neighbors and makes routing decisions based solely on its experiences with them. As pointed out by

Laniepce et al. in (Laniepce 2006), there are a number of situations in which promiscuous mode monitoring can fail. For instance, node A may not be able to correctly observe node B forwarding a packet if A receives a packet form another node outside of B's receiving zone. This scenario results in a packet collision (Laniepce 2006). Additionally, nodes using Reputed-ARAN described in (Mahmoud 2005) have a very limited localized view of the network and are not be able to make informed routing decisions. More advanced reputation systems propose gathering as much information as possible before evaluating behavior of a node. Nodes in such systems periodically exchange information they collect about their neighbors with other nodes in the network. Such systems are described in (Buchegger 2002a), (Wang 2004b), (Haghpanah 2007), (Michiardi 2002a), and (Rebahi 2005).

Reputation systems that include exchange of collected reputation information among the nodes are referred to as *collaborative*. These systems provide a better view of nodes' networking function and enable participants to make better routing decisions. However they all suffer from a common pitfall – the reputation information becomes incorrect in presence of lying nodes. The lying nodes will maliciously spread false information, reporting that a good node misbehaves or that a malicious node behaves correctly. Several approaches have been taken to counter the lying nodes and their collusions.

The mechanism that extends the reputation system and tracks nodes' behavior as participants in the reputation system is referred to as a *trust mechanism* (Jøsang

2007). For instance, in (Buchegger 2002a), which describes the mechanics of the CONFIDANT ad hoc routing protocol, Buchegger et al. discuss a trust system that is similar to Pretty Good Privacy (PGP) trust, which allows for several trust levels. PGP uses public key cryptography in which each user has a pair of cryptographic keys: a public and a private key. The public key may be widely distributed, while the private key is kept secret. Messages transmitted between a sender and a recipient are encrypted with the recipient's public key and can only be decrypted with his corresponding private key. Thus it is important that the public key which the sender uses to send a message actually belongs to the intended recipient. A public key may be digitally signed by a third party to attest to the association between the user and the key. In order to calculate the validity of a public key, PGP examines the trust levels of attached certifying signatures. Then it proceeds to calculate a weighted validity score. This means that two or more signatures with lower trust level may be considered as good as one with a high trust level. In the ad hoc network described in (Buchegger 2002a), the nodes are required to establish and maintain a list of friends to which they send reports, but the mechanics of this list are not discussed in detail. CONFIDANT only allows spread of negative experience in order to prevent nodes form falsely reporting malicious behavior as good. Since propagation of positive experience is not allowed, the redemption process in case a node improves its behavior is slow.

In contrast, ad hoc routing protocol CORE (Michiardi 2002a) only allows dissemination of positive experience in order to prevent lying nodes to distribute false accusations. This system is thus slow to react to misbehaving nodes. Despite this

limitation, (Michiardi 2002b) shows that, under the right circumstances, CORE is still able to ensure cooperation of at least half of the nodes in the network

A secure ad hoc routing protocol SORI presented in (Wang 2004b) does not include any special safeguards for chronically lying nodes. Haghpanah et al. in (Haghpanah 2007) recognize this problem and introduce a trust mechanism in order to make SORI immune to lying nodes. In addition to keeping track of the reputation values, nodes also keep track of trust values for its neighbors. Whenever a node receives a reputation report from another node, it compares the new report to its own record of observations. If the reported behavior is different from the observed behavior, the trust for the reporting node (that issued the report) will decrease. The trust value of a reporting node directly influences the effect that the reported information will have on the current reputation values on the node which received the report. Results presented in (Haghpanah 2007) show that this mechanism improves performance of SORI in presence of lying nodes. However, because trust values also need to be tracked, this expansion of the ad hoc routing protocol SORI doubles storage requirements for each reputation record or each node.

The ad hoc routing protocol solution SAFE discussed in (Rebahi 2005) does not employ a separate trust system, but rather uses reputation values as an indication of nodes' honesty. This is not an adequate solution, however, since reputation values do not indicate how well the node is performing as a participant in the reputation system.

Motivation based reputation systems offer incentives to nodes in order to stimulate their cooperation. This method does not involve active monitoring of the network, but only works to encourage packet forwarding. A motivation based solution presented in (Buttyan 2001) involves introduction of a virtual currency called nuglets. Nuglets are used in every transaction. There are two different models that involve the use of nuglets: the Packet Purse Model and the Packet Trade Model. In Packet Purse model the node is responsible for including a certain amount of nuglets in a packet it wants to send. Each forwarding node will then take out nuglets as a compensation for its forwarding service. This model discourages users from flooding the network with their packets, but the source node needs to know exactly how many nuglets it needs to include in the packet. It needs to be noted that there is no mechanism to prevent forwarding nodes from taking out more nuglets than they should (Michiardi 2002a). In the Packet Trade Model, each intermediate node pays with nuglets for the packet from a previous node in the path, thus ultimately making the destination paying for the packet delivery. This system does not prevent malicious flooding of the network. Additionally, the intermediate nodes could potentially take the payment and drop the packet (Michiardi 2002a).

Wang et al. in (Wang 2004a) propose a scheme where all nodes on a path receive the maximum possible amount for their forwarding service so they do not benefit from cheating. However, the paper does not discuss how the payments are made. Wang et al. assume that all nodes will truthfully reveal their cost, but they note

that this system may fail in case where a group of nodes attempts to maximize their total payments by circular routing.

Buttyan et al. in (Buttyan 2002) suggest implementation of nuglet counters which are increased for forwarding services and decreased for sending. The nodes are only able to send packets while their nuglet counters are positive. As Buttyan et al. discuss, this scheme requires a special tamper proof hardware to disallow a node to increase its nuglet counters illegitimately. Jakobsson et al. in (Jakobsson 2003) note that a node's ability to accumulate nuglets in this scheme will depend on its position in the network; the nodes positioned in the center will have more chances to perform routing and earn nuglets that the nodes near the edges. Additionally, Jakobsson et al. (Jakobsson 2003) require the presence of base stations and an accounting center in order to insure the correct operation of the incentive system. Base stations and an accounting center are not available in ad hoc networks.

In general, all payment methods require that the payer, in addition to cost, is able to recognize the payee using some form of identity which is not easily established in ad hoc networks. If a method for tracking the identity of payees does not exist, the payer can attach some amount of currency which is to be taken by, to him, unknown payees. In this case, there needs to be an audit system that will track the payments to prevent potential abuse (Jakobsson 2003). Thusly, an incentive method alone without a monitoring component is not sufficient to ensure cooperation in ad hoc networks.

Detection based reputation systems such as those implemented in (Buchegger 2002a), (Wang 2004b), (Rebahi 2005), and (Michiardi 2002a) do not offer incentives but threaten to exclude misbehaving nodes from the network when their reputation falls below some threshold. Excluded nodes are not allowed to either send or forward packets. CONFIDANT protocol allows nodes to rejoin the network after a timeout. This feature may result in a potential vulnerability, as malicious nodes can repeat their malicious behavior after they rejoin. In order to counter this vulnerability, SAFE assigns a critical reputation value to readmitted nodes. In case they continue to behave maliciously, they are excluded again after a shorter time period. However, this mechanism requires keeping track of nodes identities indefinitely after they leave the network. In addition, the solution proposing exclusion of *all* misbehaving nodes leads to a lower global connectivity of the network and reduces its functionality.

It is to be expected that the behavior of a node will change over time, either for better or for worse. For example, as a node is using up its battery power, it is possible that it will start behaving selfishly and drop packets it is supposed to forward for its peers. In order to account for these changes, reputation systems need to track history of nodes' behavior. This historic behavior is, in some fashion, included in the calculation of node's total reputation score. In ad hoc routing protocol SAFE (Rebahi 2005), more weight is assigned to the more recent observations. In CORE (Michiardi 2002a), more weight is given to previous observations than to more recent ones in order to avoid decreasing a node's reputation score in case of sporadic, unintentional selfish behavior. However, if previous observations are given more weight, it is possible for

misbehaving nodes to build up good reputation and "become bottlenecks with impunity" (Laniepce 2006).

Finally, we will discuss several different solutions employed for calculating reputation values upon which nodes in an ad hoc network make routing decisions.

As discussed previously, the reputation system Reputed-ARAN described in (Mahmoud 2005) is not a collaborative reputation system, thus the reputation information is not shared among the nodes. Each node will only track reputation values for its neighbors. The initial reputation value is always 0. For each observed correctly-forwarded-packet, a node is awarded +1 reputation point, while for each dropped packet the node is punished by -2 reputation points. Once the reputation counter for a particular node reaches -40, the node will be excluded from the network. It is allowed to rejoin after 5 minutes and its reputation is reset to 0. This very simple approach may potentially allow nodes to accumulate reputation points and misbehave sporadically, keeping their reputation value above -40.

SORI (Wang 2004b) uses a ratio of correctly forwarded to correctly received (to be forwarded) packets. In addition to reputation values, SORI tracks a value describing the confidence that the current reputation value is true. The confidence value is essentially a measure of the frequency with which a node attempted to route a packet through a particular peer with the same outcome. This solution obviously requires a more complex implementation and more data storage for reputation-related values.

In CONFINDANT (Buchegger 2002a), the reputation system only reacts to negative experiences. In presence of sufficient evidence of malicious behavior, the node's reputation rating is changed "according to a rate function that assigns different weights to the type of behavior detection, namely the greatest weight for own experience, a smaller weight for observations in the neighborhood, and an even smaller weight for reported experience" (Buchegger 2002a). Since the negative reputation values never decrease, the only way for a node to regain its reputation is by either waiting for a time out, when the black list of misbehaving nodes cleared, or by being removed from the black list if it exhibits the correct behavior for a certain amount of time. However, Buchegger et al. do not specify what the time interval is.

In contrast to CONFIDANT, CORE (Michiardi 2002a) reputation system reacts to both positive and negative experiences. However, only positive experiences are exchanged among the nodes. The reputation value nodes use to make routing decision is a combination of the locally observed behavior and the indirect reputation (i.e. observations neighbor nodes made and reported). The local reputation value is a value between -1 and 1, where 0 represents neutral reputation. It is calculated as a weighted mean of the observation's rating factors, giving more weight to the past experiences. Nodes are rated with -1 for a negative impression, and +1 for a positive impression. When local and indirect reputations are combined, the weight of indirect reputation depends on the trustworthiness of the reporting node. Nodes with a negative reputation value are excluded from the network.

Similarly to SORI, SAFE (Rebahi 2005) calculates reputation as a ratio of dropped to forwarded packets. More recent observations are given greater weight. Reputation values vary from 0 to 1. As mentioned previously, SAFE does not employ a separate trust system, but uses reputation values as trust values. Thus reports from nodes that have better reputation will have greater weight than the reports form the nodes with lesser reputation. Disconnected misbehaving nodes are allowed to rejoin the network after a certain time interval, but they are assigned a critical reputation value, lower than the neutral reputation. In case these nodes do not improve their behavior, they are excluded again after a shorter time interval.

Marti et al. in (Marti 2000), propose a mechanism where reputation value is periodically increased for all nodes actively used in routing paths. If a link breaks, and a node does not perform the routing function, the reputation rating is decreased by 0.05. In (Marti 2000) exchange of both positive and negative reputation ratings is allowed, but the exact mechanism of merging local and reported reputation is not discussed.

The reputation systems that use incentives to encourage cooperation among nodes, such as (Buttyan 2001), (Wang 2004a), (Jakobsson 2003), and (Buttyan 2002) rely on currency balance to reflect a current reputation value of a node.

Buchegger et al. state that, “the ratio of good to bad behavior reflects the willingness to cooperate only in relation to a specific extent of demand or opportunity for cooperation. This extent is lost in the ratio and therefore unknown” (Buchegger

2008). This limits the explanatory power of the cooperation ratio. Nodes that are positioned in such a way where they receive greater number of requests for forwarding will have a better opportunity to establish good reputation even if they do not correctly process all of the requests. Furthermore, Buchegger et al. note that "if the absolute number of cooperation instances is taken as the basis for reputation calculation, it is not known what the number of opportunities was out of which the behavior was good or bad" (Buchegger 2008). We believe that our proposed system, as we describe it in the next chapter of this project, contains a solid statistical basis to capture "willingness to cooperate in relation to opportunity"(Buchegger 2008).

In conclusion, we observe that all the discussed schemes group nodes in two categories – good and misbehaving, where misbehaving nodes include both malicious and selfish nodes. We consider malicious nodes more costly in network operations, and thus agree that they should be excluded from the networking function (i.e. not allowed to send packets or trusted to route packets for other nodes). However, selfish nodes should not be excluded from the network, but given an incentive to improve their cooperation. This calls for an ability to distinguish between selfish and malicious nodes. Our system is able to make this distinction.

Chapter III

METHODOLOGY

3.1 Fine-Grained Reputation System

Our fine-grained reputation system is designed to improve reliability and security in wireless ad hoc networks. The main objectives of our system are to improve connectivity and cooperation among wireless ad hoc nodes while reducing interaction with malicious nodes.

In a fine-grained reputation system, each node stores reputation information for other nodes in the network. This reputation information indicates how well the node behaves as a participant in the wireless ad hoc network. Based on the reputation of the intermediate nodes, nodes in an ad hoc network are able to select the most reliable and secure path from the data source to the destination. Reputation information indicating that a particular node behaves inadequately will trigger a response from the rest of the network resulting in disadvantages or a punishment for the misbehaving node. The reputation system allows for a different response depending on the severity of the node's bad behavior. This feature is an advantage in comparison to the reputation systems which classify nodes in only two categories - good and bad.

3.1.1 First-hand reputation

Each node in the fine-grained reputation system is responsible for observing the forwarding behavior of its neighbors using the promiscuous mode on its network interface.

For example, node A observes the number of correctly forwarded, dropped, or maliciously modified packets from its neighbor, node B. Based on outcomes drawn independently from these observations, node A assumes that the behavior of node B follows a probability of $F_{A,B}$. This probability varies for each packet A observes on node B. Since the parameters $F_{A,B}$ are unknown, we model this uncertainty by assuming that $F_{A,B}$ is drawn from a distribution (the prior) and updated when new observations are available. If the likelihood of a node's behavior is binomial, i.e. the behavior can be good or bad and it occurs independently, a good prior distribution is the Beta distribution (Buchegger 2003). Since we want to account for more than just two independent variables (types of behavior), we use the multivariate generalization of the Beta distribution – the *Dirichlet distribution*.

Bayes' theorem allows us to compute the posterior probability when new observations and prior knowledge are available. Expressed in mathematical terms, Bayes' theorem has the form:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}.$$

P(A|B) is the conditional probability of event A being observed given the probability

of event B. This factor is also called posterior probability because it is derived from the specific value of B. *P(B|A)* is a conditional probability of event B given A. *P(A)* is the prior probability of event A, in the sense that it does not depend on the information about B. *P(B)* is the prior probability of B, and it acts as a normalizing constant (Bolstad 2007). In probability theory, a normalizing constant is used to multiply a nonnegative function in order to make it a probability density function (in order that the area under its graph is 1) (Feller 1968). The probabilities of random events are assigned according to the frequencies of their occurrence in random sampling.

We assume that $F_{A,B}$ follows the Dirichlet distribution. The Dirichlet distribution is denoted as Dir($\boldsymbol{\alpha}$), where $\boldsymbol{\alpha}$ is a vector of positive real numbers. $\alpha_1,\ldots,\alpha_n$ are the shape parameters for the probability density function (*pdf*) of the Dirichlet distribution. Informally, a pdf is a smoothed out version of histogram: "if one empirically samples enough values of a continuous random variable, producing a histogram depicting relative frequencies of output ranges, then this histogram will resemble the random variable's probability density, assuming that the output ranges are sufficiently narrow" (Mendenhall 2008).

We use the Dirichlet distribution because it is a conjugate prior to the multinomial distribution. A multinomial probability distribution describes the probability that each independent trial result is exactly one of some fixed finite number *n* of possible outcomes with probabilities $p_1,\ldots,p_n$ and as such is a natural choice in

our solution (Fink 1997). The set of parameters for the *i*th attribute model of the Dirichlet distribution $\alpha_i = \{\alpha_i\}_{i=1}^{r}$ corresponds to a multinomial distribution.

In Bayesian probability, a class of prior probability distributions *P(θ)* is said to be conjugate to a class of probability functions *P(x|θ)* if the resulting posterior distributions *P(θ|x)* are in the same family as *P(θ)* (Raiffa 1961). When a conjugate prior is multiplied with the likelihood function, it results in a posterior probability that has the same functional form as the prior, allowing the posterior to be used as a prior in further computations.

Thus, under the assumption that the prior pdf $f_{k-1}(r)$ follows a Dirichlet distribution, the posterior pdf $f_k(r)$ also follows a Dirichlet distribution. Given the Dirichlet prior $P(X_i) = Dir(X_i \mid \alpha_{i1},...,\alpha_{ir})$, where α_{iz} are positive constants, the posterior distribution of θ_i can be computed using the Bayes' theorem as

$$P(\alpha_i \mid D) = \frac{P(D \mid \alpha_i)P(\alpha_i)}{p(D)} = Dir(\alpha_i \mid \alpha_{i1} + N_{i1},..., \alpha_{ir_i} + N_{ir_i})$$

Starting with the initial state as the prior distribution, the parameters can be updated when new data D is available. Variable N describes instances of new data. We define the reputation value based on the first-hand observations assigned to a node at a time *t* to be equal to the expectation value of the Dir($\boldsymbol{\alpha}$).

For clarity, we now show through an example how this approach works. In Table 3.1 we illustrate how node A updates the reputation value of its neighbor, node

B. Observations about packets forwarded by node B are made in equal time intervals or windows. We will discuss the concept of a window in more detail later in this chapter. In this example, each window contains a constant number of 50 packets. This is equivalent to an assumption that node B processes (forwards) 50 packets per time interval. Given that we are observing three possible events: friendly, selfish, or malicious behavior, α_1, α_2, α_3 are the shape parameters for the Dirichlet probability density function and they indicate whether the node B is friendly, selfish, or malicious. Let observations of correctly forwarded, dropped, or maliciously modified packets be $X = (X_1, \ldots, X_k) \sim \text{Dir}(\boldsymbol{\alpha})$ and $\alpha_0 = \sum_{i=1}^{K} \alpha_i$, where K = 3, then the expectation value of the distribution is the vector $(x_1,\ldots,x_k)$ where $x_i = \frac{\alpha_i}{\alpha_0}$. Parameter K is equal to the number of shape parameters in $\text{Dir}(\boldsymbol{\alpha})$. Since we are observing three independent events, K = 3 in our case.

Initially, A has no knowledge about B's behavior. We choose an optimistic approach and classify B as friendly. The shape parameters α_1, α_2 and α_3 are thus <1,0,0>. Regardless of the initial shape parameters, as new information is collected, α_1, α_2 and α_3 will converge to approximate the node's true behavior. Note that the sum of the expectation values $x_1+x_2+x_3$ has to be equal to one. Each time node A collects new data, the parameters α_1, α_2 and α_3 are updated. In the first window, 50 packets are observed. Node B correctly forwards 40 (N_1) packets, drops 10 (N_2), and does not inject any packets (N_3). The parameters are thus updated as $\alpha_1=\alpha_1+N_1$, $\alpha_2=\alpha_2+N_2$ and

$\alpha_3=\alpha_3+N_3$. The node A calculates the reputation value for its neighbor node B after it observed 5 windows as the expectation value of Dir($\boldsymbol{\alpha}$).

Table 3.1: Reputation calculation based on the observations node A makes about its neighbor, node B.

Window	**Number of observed packets**	**α_1**	**α_2**	**α_3**
1	50	40	10	0
2	50	30	15	5
3	50	40	5	5
4	50	35	10	5
5	50	40	10	0
Total number of observed packets in windows 0-5	250	185	50	15
Expectation value of X_i	---	(185)/ (250) = 0.74	(50)/(250) = 0.2	(15)/(250) = 0.06

3.1.2 Second-Hand Reputation

As Laniepce et al. point out, the promiscuous mode used by nodes to collect information about their neighbors' behavior may fail in certain cases (Laniepce 2006). More specifically, node A will not be able to correctly observe node B forwarding a packet if A receives a packet form another node outside of the B's receiving zone. As described in (Laniepce 2006), this scenario results in collision (Figure 3.1). Generally, a packet collision occurs when two or more devices in a network attempt to send packets over the network at the same time.

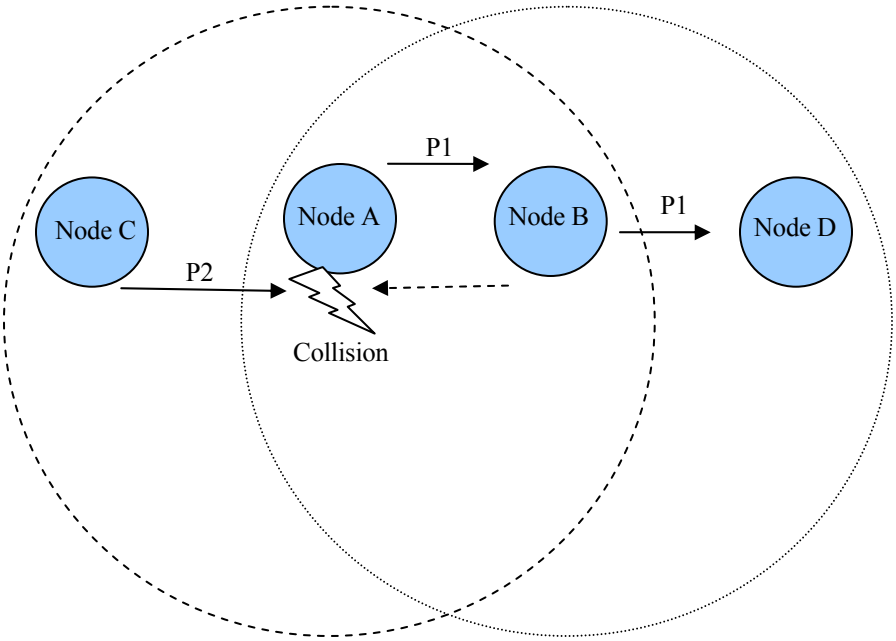


Figure 3.1: Node A is not able to can not overhear node B forwarding P1 to D because P1 collides with P2 sent by node C (Laniepce 2006).

In addition, if the nodes only relied on their own observations, they would have a localized view of the network and would not be able to make informed routing decisions. Because of these limitations, nodes in the fine-grained reputation system gather as much information as they can before making a decision about their neighbor's behavior. We employ collaborative monitoring to allow nodes to exchange observations with each other. However, we only allow nodes to propagate their first-hand observations and not the information they received from other nodes. Buchegger et al. state that passing information other than first-hand observations does not offer any gain in reputation accuracy or speed, but in fact introduces vulnerabilities by creating a "spiral of self-reinforcing information" (Buchegger 2008). They suggest a

theoretical analysis in which they find that spreading only the first-hand reputation is more robust against nodes who report inaccurate reputation information.

In our fine-grained reputation system, nodes are required to periodically broadcast first-hand reputation values they collected into their neighborhood, so that other nodes can incorporate this knowledge into their total reputation values. We refer to these reports as second-hand reputation.

In order to detect and avoid false reports, nodes use a deviation test on all reports they receive. Based on the test, nodes decide whether to increase or decrease the trust in the reporter's accuracy, and ultimately whether or not to incorporate the report into their total reputation value. The formula for the deviation test we use is:

$$| E_{AB}(Dir(\beta_1, \beta_2, \beta_3)) - E_{CB}(Dir(\alpha_1, \alpha_2, \alpha_3)) | \leq d ,$$

where the first term is the second-hand reputation, the second term is the current reputation value the node has, and d is a positive constant as the threshold. We allow 30% deviation, however we do not try to optimize this threshold. A very detailed analysis of the deviation test is available in (Mundinger 2005).

3.1.3 Trust

Each node in the fine-grained reputation system, in addition to reputation information, stores trust information for its neighbors. Trust indicates how trustworthy the neighbor's reports are. We can also say that trust value indicates a node's behavior as an actor in the reputation system. Our trust system uses a similar Bayesian approach

used in the fine-grained reputation system. The difference is that trust is expressed through only two possible instances of behavior: trustworthy and not trustworthy (as opposed to the reputation, where a node can behave friendly, selfishly, or maliciously). Because of this, we choose to use the Beta distribution as a prior. The Beta distribution is in fact a case of the Dirichlet distribution with only two probability density function shape parameters. The Beta distribution is conjugate, which means that a posterior probability will have the same functional form as the prior. Therefore, after updating, the trust value still follows the Beta distribution.

We denote the trust which node A has for node B as $T_{AB} \sim Beta(\gamma, \delta)$, where γ stands for trustworthy reports and δ for not trustworthy. Initially, $\gamma = \delta = 1$, which corresponds to uniform distribution and indicates absence of knowledge. Trust is updated when results of the deviation test are available. If the deviation test holds, the trust for the reporting node is increased by increasing $\gamma = \gamma + 1$. If the deviation test does not hold, the trust is decreased by increasing $\delta = \delta + 1$. Each time new deviation test results are available, trust parameters are updated.

Trust value for node B on node A is calculated as the expectation value of Beta(γ, δ) as $\omega_{AB} = E(Beta(\gamma, \delta)) = \frac{\gamma}{\gamma + \delta}$.

Table 3.2 illustrates changes in T_{AB} as node A receives second-hand reputation reports from node B. We can see how the ω_{AB} decreases as the number of instances of reports that do not pass the deviation test (δ) increases.

Table 3.2: Trust calculation for node B based on the deviation tests on node A as it receives second-hand reputation information from node B.

Time	**Deviation test passed?**	**Trustworthy** γ	**Not trustworthy** Δ	$\omega_{AB} = E(Beta(\gamma, \delta)) = \frac{\gamma}{\gamma + \delta}$
0	--	1	1	0.5
1st	Yes	2	1	0.667
2nd	Yes	3	1	0.75
3rd	No	3	2	0.6
4th	No	3	3	0.5
5th	No	3	4	0.429

A node's decision to accept or deny a second-hand report is based on the trust value threshold. Without attempting to optimize this threshold, we decided to consider all nodes that have $\omega \geq 0.5$ sufficiently trustworthy. Second-hand reputation from these nodes is thusly incorporated into total reputation values when it passes the deviation test. Since ω_{AB} describes the amount of trust the node has in the truthfulness of the report, we use value ω_{AB} as a discounting factor during the reputation merging process. We discuss this process next.

3.1.4 Total Reputation Based on First and Second-Hand Reputation

In order to establish the total reputation value for neighbor C, node A merges its own observations (first-hand reputation) with observations reported by other nodes (for example, node B). We refer to this value as total reputation value, and denote it as R_{AC}. Every node believes that its prior knowledge about the behavior of its neighbor, node B, is the most accurate. Thus, first-hand reputation information is merged as-is, while second-hand information is discounted by the factor ω_{AC} expressing the disbelief

in the accuracy of the report. The maximum value for trust is 1, corresponding to an undeniable fact.

The total reputation value node A calculates for node B, given second-hand reputation reported by node C is calculated as:

$$R_{AB} = F_{AB} + \omega_{AC} F_{CB},$$

where F_{AB} is the first-hand reputation, F_{CB} is the second hand reputation, and ω_{AC} is the trust node A has for node C's reports.

Let us now use an example in order to illustrate total reputation value calculation and trust value update performed by our fine-grained reputation system. Suppose that node A is calculating the total reputation value for node B. Upon receiving a second-hand reputation F_{CB} from node C, node A performs the deviation test: $|F_{CB} - F_{AB}| \leq d$. Next, node A examines the current trust value for node C, ω_{AC}. ω_{AC} is calculated as the expectation value of Beta(γ, δ), where γ is the trustworthy parameter, and δ is the not trustworthy parameter. If the report passes the deviation test and if ω_{AC} value is above the trust threshold, F_{CB} is incorporated into total reputation value. If the report does not pass the deviation test or ω_{AC} is not above the threshold, F_{CB} report is discarded. In either case, if F_{CB} passed the deviation test, the trust parameter γ is increased as $\gamma = \gamma + 1$, and thus the overall trust ω_{AC} is increased. If F_{CB} does not pass the deviation test, the trust parameter δ is increased as $\delta = \delta + 1$. As the consequence, the overall trust ω_{AC} is decreased.

Assuming that F_{CB} passed the deviation test and that ω_{AC} is above the threshold, the total reputation is calculated as $R_{AB} = F_{AB} + \omega_{AC} F_{CB}$, where we discount F_{CB} by the factor ω_{AC}. The term ω_{AC} is the trust value for node C on node A, and we use it here to account for node A's suspicion about C's honesty.

3.1.5 Moving Window Mechanism

As previously noted, nodes in the fine-grained reputation system observe their neighbors over equal time intervals or windows. Let us now discuss the details and the role that the window mechanism has in the reputation system.

In general, there are two alternative ways to update first-hand reputation. One is to update based on all observations, the other is to update based only on the most recent observations. We choose to consider the passing of time in our reputation system and update based on the most recent observation for the following reasons: it reduces computation complexity, it makes possible to early detect changes in subjects behavior, and it provides the possibility of redemption over time for a node that has been repaired. The moving window mechanism allows us to divide historic information into time intervals of equal size and consider only a limited number of these intervals for the calculation of first-hand reputation.

Table 3.3: illustrates a case in which node A is observing node B which was behaving friendly during the first 3 windows, but then became malicious.

Table 3.3: Illustration of early detection by moving window mechanism.

Window	**Number of observed packets**	**α_1**	**α_2**	**α_3**
1	50	50	0	0
2	50	50	0	0
3	50	40	0	10
4	50	40	0	10
5	50	40	0	10
Windows 0-5				
Total number of observed packets in windows 0-5	250	220	0	30
Expectation value of X_i	---	(220)/(250)= 0.88	(0)/(250) = 0	(30)/(250)= 0.12
Windows 3-5				
Total number of observed packets in windows 3-5	150	120	0	30
Expectation value of X_i	---	(120)/(150)= 0.8	(0)/(150)= 0	(30)/(150)= 0.2

From Table 3.3 we see that if the size of our moving window is five, the first-hand reputation is $F1_{AB} = < 0.88, 0, 0.12>$. If the size is three, only the three most recent observations are considered and the first-hand reputation is $F2_{AB} = < 0.8, 0, 0.2>$. We see that F2 better reflects the change of behavior in node B, as the malicious component of F2 is higher than in F1.

However, we must not forget that a wireless network is prone to errors in packet content and occasional packet dropping due to interference or other wireless-medium-related causes. If we use a small moving window, the reputation system will be extremely responsive to even minor changes in nodes behavior. In the case of a wireless network, this may be undesirable. The reputation system could overreact and label a node as misbehaving only because of medium-related issues. If we choose a

larger moving window, reputation will not oscillate due to occasional errors, but the system may not be able to immediately detect changes in node's behavior. We did not attempt to optimize the size of the moving window as it depends on physical properties of the network. We chose the size that seemed to be the most appropriate for the purposes of our analysis.

Next, we discuss the approach we have taken to integrate fine-grained reputation into a wireless ad hoc routing mechanism.

3.2 Integrating Reputation into Wireless Ad Hoc Routing

An ad hoc routing protocol can be defined as "a convention or a standard that control how nodes come to agree which way to route packets between computing devices in a mobile ad hoc network" (Lang 2008). Next, we summarize characteristics of wireless ad hoc routing without extensions offered by a reputation system.

3.2.1 Wireless Ad Hoc Routing Without Reputation

Most ad hoc routing protocols assume that the participating nodes will cooperate and perform their forwarding duties for other nodes. The only task such protocol needs to handle is maintaining connectivity among the nodes. Ad hoc routing protocols classified as reactive or on-demand (NetworkWorkingGroup 2007), (NetworkWorkingGroup 2003) find a route from a source to a destination only when it is needed. Typically a source node floods the network with some type of route

discovery control packets. Route discovery packets propagate through the network and accumulate addresses of each node between the source and destination. Intermediate nodes may take advantage of the accumulated path information and cache it for future use. Once the path is discovered, the source node will receive a reply packet containing the address of each intermediate node the packet needs to traverse (Lang 2008). Because the best path is determined by the routing protocol, the user has no influence over what path his or her data takes on the way to the destination. Routing protocols typically calculate the "goodness" of a path based on a cost of each individual node on that path. The cost of each node is assigned using some cost function. The protocols search for the path that has the minimum total cost, i.e. the least-cost path. The simplest example is the shortest-path approach, where each node has the same cost of one. When the routing protocol searches for a path, if will find the one that contains the smallest number of intermediate nodes, since its cost will be the smallest.

If the ad hoc routing protocol does not have a mechanism to detect presence and activity of misbehaving nodes, the chosen path may contain selfish and/or malicious nodes. As a consequence, in the presence of non-cooperative nodes, the performance of the network will degrade, until ultimately the network is rendered useless. One of many examples is the experiment presented in (Buchegger 2002a), which shows that "the defenseless" DSR (Dynamic Source Routing) protocol loses about 70% of packets when the fraction of misbehaving nodes is one third of the total population of 50 nodes. Integrating a reputation system into a wireless ad hoc protocol

allows the protocol to be aware of the behavior of each node and makes the ad hoc network more robust in the presence of misbehaving nodes.

3.2.2 Integrating Reputation into Wireless Ad Hoc Routing

The advantage that a reputation system offers to a routing protocol is the ability to classify nodes into good and misbehaving and take appropriate actions against the misbehaving nodes in order to protect the security and reliability of the network. Our fine-grained reputation system further extends this feature by allowing the protocol to classify nodes in one of the three categories - friendly, selfish, or malicious, and treat selfish and malicious nodes separately. By definition, malicious nodes compromise the integrity of the data packets, thus we consider them more dangerous than selfish nodes. While both security and reliability of an ad hoc network are important, we think that the damage inflicted by a security breach is more likely to be serious and irreparable than a loss of data packets caused by selfish nodes. Because of this, we suggest that malicious nodes should be temporarily isolated by disallowing them to either send or forward packets. Selfish nodes, however, should not be excluded from the network in the same fashion as malicious, since this approach lowers the overall network connectivity. They should be prevented to benefit from their selfish behavior, and given an opportunity and incentive to improve.

The fine-grained reputation system is a distributed reputation system, meaning that each node stores a reputation table representing its own view of its environment.

In other words, a global view of the reputation for all nodes does not exist. Our design is an appropriate match for the decentralized, infrastructure-less architecture of wireless ad hoc networks. The first and second-hand reputation mechanisms allow every node to obtain reputation values for nodes at most two hops away (Figure 3.2).

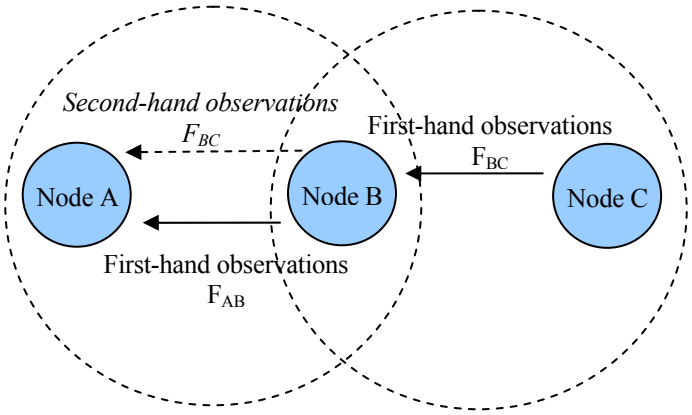


Figure 3.2: Second-hand reputation mechanism allows node A to find out about node C's behavior, even though C is not A's neighbor.

In a wireless routing scheme coupled with the fine-grained routing protocol, each node assigns cost to other nodes depending on the current total reputation values for those nodes. Since it is more practical to describe cost as one numerical value, we convert the reputation 3-tuple *(<f,s,m>)* into a single-valued cost using the following function:

$$C(f_A, s_A, m_A) = af_A + bs_A + cm_A,$$

where *a*, *b*, and *c* are constants such that $1 \le a < b < c$, and $< f_A, s_A, m_A >$ is the total reputation value for node A for which holds that $f_A + s_A + m_A = 1$. The constants *a*, *b*,

and *c* are used to distinguish between the cost of friendly, selfish and malicious nodes. The constants can be adjusted in such way where nodes with a certain high probability of selfish behavior are considered to have the same cost as malicious nodes, or where a certain amount of malicious behavior is tolerable as it can be attributed to transmission errors. Regardless of the approach, the cost boundaries between friendly and selfish, and selfish and malicious nodes have to be clearly defined so that the routing protocol is able to distinguish among the groups. For instance, we consider selfish nodes to always have greater cost than the friendly nodes (regardless of how small the probability of selfish behavior may be), and similarly, the malicious nodes to always have greater cost than the selfish nodes. We express this relationship as: $1 \leq af_A < bs_A < cm_A$. The upper bound for cm_A is equal to the infinite cost indicating a disconnected link. This is consistent with our strategy to temporarily isolate malicious nodes after they reach and pass a certain malicious threshold.

If one path in the network has a slightly better reputation than other paths, it will most likely be selected by multiple sources as the best path. It is possible that nodes on the path become overloaded by forwarding requests. These nodes will not be able to forward all the packets they receive and will start dropping packets, thereby increasing their selfish component. Eventually, the reputation of the nodes on this path will decrease, and a new best path will be selected. The friendly nodes which were not able to process packets because of their physical limitations and unrealistic expectations will be treated as selfish. Since the routing protocol will keep seeking the

least-cost path, this scenario could continue repeating indefinitely. To balance packet load on friendly nodes, we suggest that instead of one best path, the routing protocol selects two or three disjointed "best" paths, if such exist. Disjointed means they do not have any intermediate nodes in common. In the set of selected paths, one path will be the best and the inferior paths may have a greater selfish component. It is likely that one of the selected paths is better than the rest. However, we let the source occasionally, with a predetermined (lower) probability, select one of the inferior paths. By doing this, the selfish nodes on the path are given a chance to improve their behavior, and the otherwise friendly nodes on the best path will have a lower chance of becoming overloaded.

As we discussed previously, the moving window mechanism allows us to adjust responsiveness of the fine-grained reputation system to changes in nodes' behavior. The smaller the size of the moving window, the quicker is the system to detect misbehavior. The moving window allows us to implement a gradual redemption of malicious nodes once they have been isolated. If we were to allow them to quickly recover their reputation, malicious nodes may be tempted to misbehave again. The procedure is the following: once a malicious node reaches the reputation threshold for isolation, we set its reputation cost to infinity and forbid it from forwarding packets by not sending it any data. The reputation threshold is adjustable and depends on the tolerance the network has for malicious behavior. After a timeout ((Buchegger 2002a), (Rebahi 2005)), we reset the node's reputation to just above the threshold value. In case the node repeats its malicious behavior, its reputation will immediately fall below

the threshold, and it will be isolated again. We also increase the moving window size used for monitoring of this node in order to increase the time needed for the node to recover. In Figure 3.3 we illustrate the redemption process for a malicious node that exhibit the same rate of change in behavior as a selfish node. The moving window size is 3, and the number of observed packets in each window is 50. The threshold for malicious behavior is $m=0.7$, where m is the malicious component of the reputation vector $<f,s,m>$. We illustrate the variation in the reputation through changes in the friendly component. Given that $f+s+m=1$, and $s=0$, it follows that $f=0.3$. We triple the size of the moving window for the purpose of the redemption of the malicious node. We see in Figure 3.3 that an increase in friendly behavior in both nodes after the 9^{th} window. Due to the timeout, reputation for the malicious node was reset to the threshold value, $f=0.3$. Reputation for the malicious node was not reset, thus the node started its redemption from a lower f value. We see from the graph that in the 27^{th} window, the selfish node almost redeemed its friendly reputation, $f=1$. In contrast, for the malicious node the friendly reputation was not fully regained even after 37 windows. This is due to the fact that new reputation for the selfish node was calculated over the moving window size of 3, while for the malicious node it was calculated over the triple size of the original moving window.

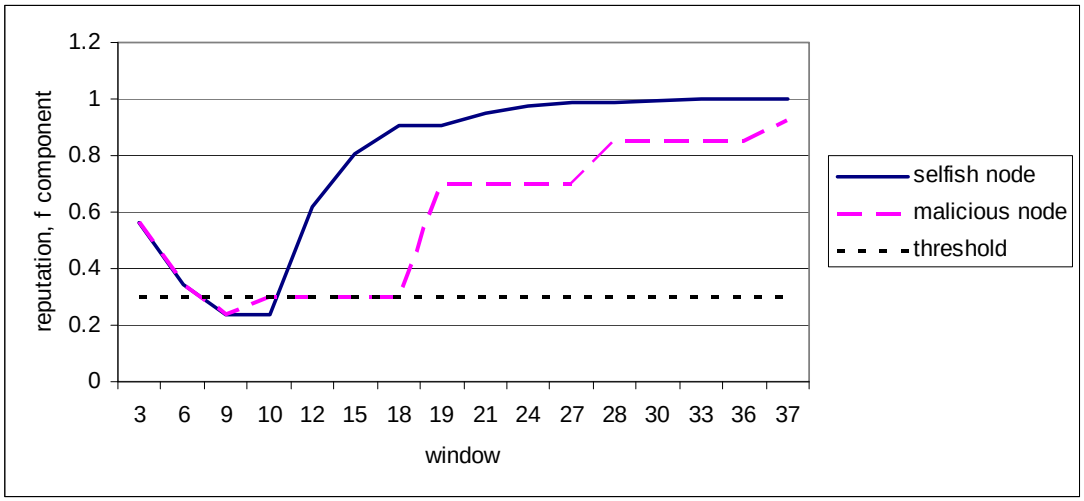


Figure 3.3: Redemption of a malicious node reflected through the change in the friendly reputation component.

In the next chapter we will describe the analysis of the performance of the fine-grained reputation system.

Chapter IV

ANALYSIS OF RESULTS

4.1 Introduction

In order to evaluate the benefits of the fine-grained reputation system, we compare it with a reputation system that is able to classify nodes only as friendly (good) and misbehaving (bad). We will refer to this system as to a beta reputation system because it assumes that the nodes' behaviors follow the Beta distribution. We consider two variations of the beta reputation system: a system which classifies all misbehaving nodes as selfish, and a system which classifies all misbehaving nodes as malicious.

We used a custom simulator developed with Adobe Flash CS3 in ActionScript3.0, targeted for Flash player version 9 or greater. (The simulator code is available upon request. Email: alma.cemerlic@gmail.com) We chose Flash because of its excellent graphical and interactive capabilities, support for object-oriented programming provided by ActionScript3.0, and platform independence and portability. The simulations are run on Intel Core2 Duo at 2.0GHz laptop with RAM size of 2 GB. The operating system is Windows Vista Ultimate SP1.

In our simulations, we monitor data packets exchange among nodes, and we are not concerned with any wireless ad hoc protocol control packets used to establish and maintain connectivity among nodes. We are also not concerned with any packets exchanged as a part of the reputation system. Data packets are sent at an average rate of 50 packets per second (Yuen 2008). Injected packets are assigned to a packet flow form a source to a destination when the flow is routed through a malicious node. We assume that the routing paths are provided and maintained by the underlining routing protocol. We also assume that the reputation information is exchanged in a correct manner regardless of a node's reputation classification.

For our simulation cases, we use a constant number of 50 nodes randomly placed on a 800m by 800m stage. The diameter of the maximum communication range for each node is 75 meters. Every node has at least one edge connecting it to the rest of the network. All edges are bidirectional, meaning that if nodes A and B are neighbors, and A is able to receive packets from node B, node A could instead send packets to node B. We assume that the mobility rate in the ad hoc network is low, and considering the duration of our simulation cases (500 seconds), we assume the nodes are static.

The pattern of packet exchange among the nodes is random for each simulation case. The same cases are run on the fine-grained and the beta reputation systems. During the simulation, packets are sent from a source to a destination every second.

All misbehaving nodes belong to one of the two classes – they are either selfish or malicious. For the purpose of the simulation cases, we define selfish nodes as the ones that consistently drop approximately 20% of the packets they are supposed to forward for other nodes. We define malicious nodes as the ones that consistently inject 10 packets per 50 forwarded packets. The total percent of misbehaving nodes varies from 4% to 40%. We follow the example set in (Michiardi 2002a) and limit the population of misbehaving nodes to 40% of the total node population. Michiardi et al. claim that this is in most cases unrealistically high ratio of misbehaving nodes, and thus we take it to be sufficient to test the performance of the reputation systems. In each simulation case, the ratio of selfish and malicious nodes is different, but their total number never exceeds 40% of the total size of the node population. We group simulation cases in four main sets according to the total number of misbehaving nodes: 40%, 30%, 20% and up to 10%. We then vary the ratio of selfish and malicious nodes within each set as follows:

Set 1 (40%): case (1) 4% selfish, 36% malicious, case (2) 10% selfish, 30% malicious, case (3) 20% selfish, 20% malicious, case (4) 30% selfish, 10% malicious, case (5) 36% selfish, 4% malicious.

Set 2 (30%): case (1) 4% selfish,26% malicious, case (2) 10% selfish, 20% malicious, case (3) 20% selfish, 10% malicious, case (4) 26% selfish, 4% malicious

Set 3 (20%): case (1) 4% selfish, 16% malicious, case (2) 10% selfish, 10% malicious, case (3) 16% selfish, 4% malicious

Set 4 (up to 10%): case (1) 4% selfish, 6% malicious, case (2) 4% selfish, 4% malicious, case (3) 6% selfish, 4% malicious.

The performance metrics used to compare the fine-grained and the beta reputation systems are effective throughput, connectivity, and the percentage of injected packets introduced in the system by malicious nodes. Effective throughput is used to express the fraction of total occupied bandwidth used by legitimate traffic, where legitimate means that the traffic was not produced as a consequence of the malicious activity. Connectivity is a function of the number and a cost of possible paths in the network connecting all available nodes capable of forwarding packets. We use this metric to distinguish the fine-grained reputation system from the beta system which treats all misbehaving nodes as malicious and reacts to misbehavior by disconnecting the nodes from the network. The percentage of injected packets to total number of packets instantiated in the network is used to evaluate the opportunity given to the malicious nodes to affect the network traffic. This metric is used to compare the fine-grained reputation system with the beta system which treats all misbehaving nodes as selfish and assigns their cost with respect to the probability of misbehavior.

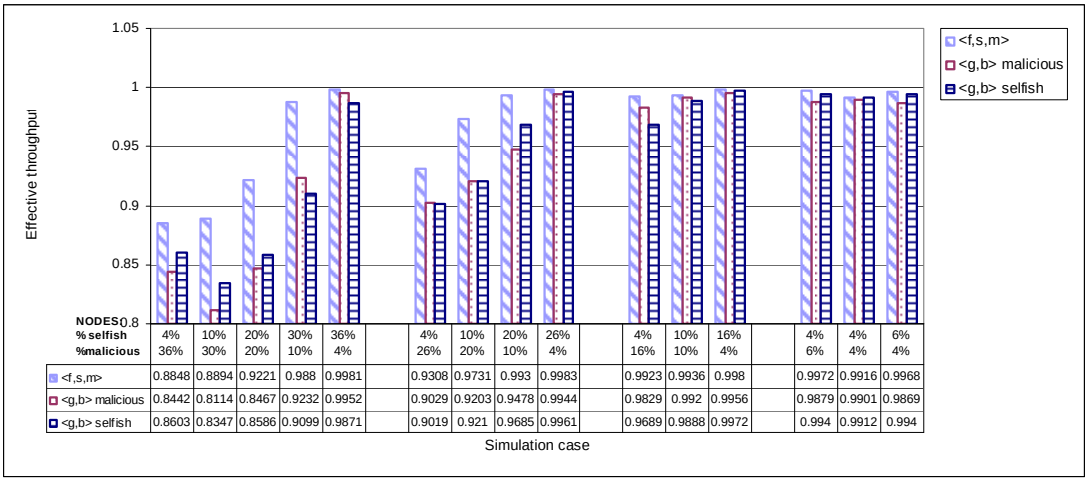
4.2 Simulation Results

The following is a comparison of the fine-grained reputation system and a reputation system able to classify behavior of ad hoc nodes only as friendly (good) or misbehaving (bad). We consider two variations of the beta system – the one that treats

all nodes as malicious and the other that treats all nodes as selfish. The reputation systems are compared using the following metrics: effective throughput, connectivity, and the percentage of injected packets introduced in the network by malicious nodes.

4.2.1 Effective Throughput

Figure 4.1 compares the effective throughput for the fine-grained and the beta reputation systems across the simulation cases. From the figure, it is possible to see that the fine-grained reputation system has overall higher effective throughput than either of the beta reputation system variations.



NODES	4%	10%	20%	30%	36%		4%	10%	20%	26%		4%	10%	16%		4%	4%	6%
% selfish	36%	30%	20%	10%	4%		26%	20%	10%	4%		16%	10%	4%		6%	4%	4%
%malicious																		
<f,s,m>	0.8848	0.8894	0.9221	0.988	0.9981		0.9308	0.9731	0.993	0.9983		0.9923	0.9936	0.998		0.9972	0.9916	0.9968
<g,b> malicious	0.8442	0.8114	0.8467	0.9232	0.9952		0.9029	0.9203	0.9478	0.9944		0.9829	0.992	0.9956		0.9879	0.9901	0.9869
<g,b> selfish	0.8603	0.8347	0.8586	0.9099	0.9871		0.9019	0.921	0.9685	0.9961		0.9689	0.9888	0.9972		0.994	0.9912	0.994

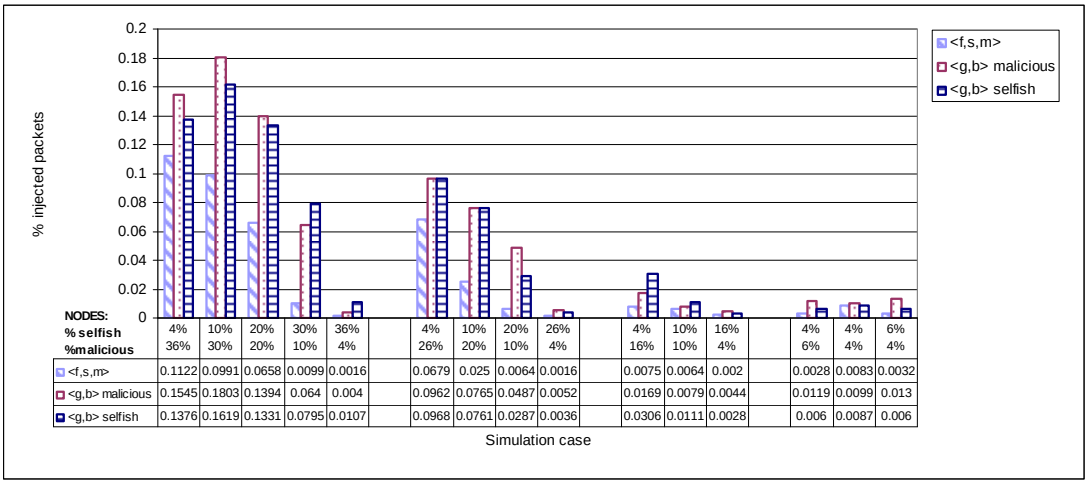
Figure 4.1: Effective throughput - comparison for the fine-grained <f,s,m> and beta <g,b> reputation systems.

Since in the beta reputation systems both selfish and malicious behavior are considered to have equal cost, the beta reputation systems are likely to make a mistake and choose malicious over selfish nodes when estimating a path cost. Such false decisions will compromise the security of the network, which we always consider to

be more costly than a possibility of packet loss. In general, we can say that we expect a better effective throughput from the fine-grained reputation system than the beta reputation systems that treat all misbehaving nodes as either selfish or malicious. This is especially true in the presence of a larger number of malicious nodes.

4.2.2 Percentage of Injected Packets

Next, we examine the percent of injected packets that traversed the network during the simulation. Injected packets are introduced in the network as a result of malicious activity. By our definition, when a legitimate traffic is routed through a malicious node, the node will inject 10 packets per every 50 packets it forwards. These packets are assigned to the legitimate packet stream and travel to the destination. Figure 4.2 shows a graphical comparison of the fine-grained reputation system and the two beta systems. The numbers are consistent with the previous comparison of the effective throughput. We observe that the percent of injected packets is lower for the fine-grained reputation system than for either of the beta reputation systems.



NODES: % selfish %malicious	4% 36%	10% 30%	20% 20%	30% 10%	36% 4%		4% 26%	10% 20%	20% 10%	26% 4%		4% 16%	10% 10%	16% 4%		4% 6%	4% 4%	6% 4%
<f,s,m>	0.1122	0.0991	0.0658	0.0099	0.0016		0.0679	0.025	0.0064	0.0016		0.0075	0.0064	0.002		0.0028	0.0083	0.0032
<g,b> malicious	0.1545	0.1803	0.1394	0.064	0.004		0.0962	0.0765	0.0487	0.0052		0.0169	0.0079	0.0044		0.0119	0.0099	0.013
<g,b> selfish	0.1376	0.1619	0.1331	0.0795	0.0107		0.0968	0.0761	0.0287	0.0036		0.0306	0.0111	0.0028		0.006	0.0087	0.006

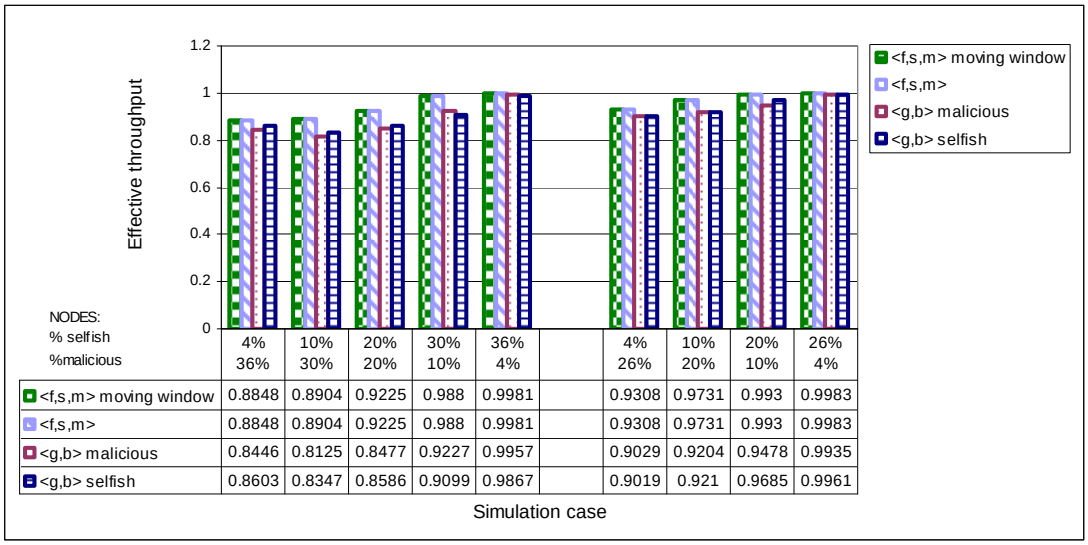
Figure 4.2: Percentage of injected packets - comparison for the fine-grained <f,s,m> and beta <g,b> reputation systems.

We attribute the results of the comparison for both, the effective throughput and the percentage of injected packets, to the fact that the beta reputation systems are not able to make distinction between the two classes of misbehaving nodes – selfish and malicious. In the fine-grained reputation system, malicious nodes are always assigned higher cost than selfish nodes. This means that a very low probability of malicious behavior is always more costly than a very high probability of selfish behavior. In the beta reputation system, the cost of nodes depends on their relative probability of misbehaving, regardless of whether the nature of misbehavior is selfish or malicious. In case that a malicious node has a lower probability of misbehavior than a selfish node, the beta reputation system will assign a lower cost to the path containing that malicious node than to the one containing the selfish node. In the fine-grained reputation system this could not happen, as there is a clear distinction between selfish and malicious behavior.

4.2.3 Effective Throughput and Percentage of Injected Packets with Moving Window Mechanism

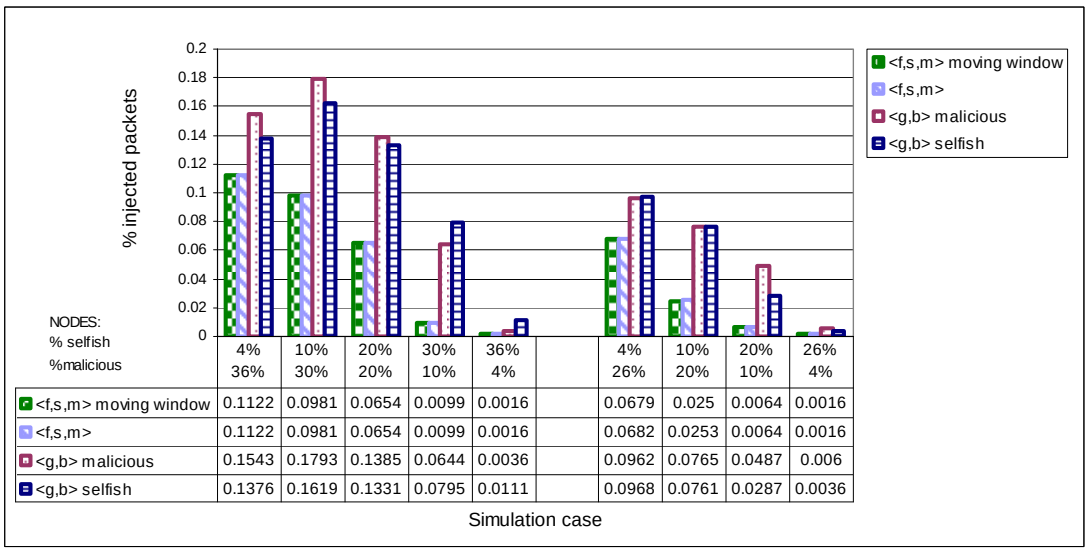
We now show how the moving window mechanism influences the performance of the fine-grained reputation system. The moving window mechanism reacts to a node's misbehavior by tripling the size of the observation window once the node's malicious behavior reached a threshold. We did not attempt to optimize this threshold. The size of the observation window is increased only in the positive direction – if the node's behavior improves, it will take a greater number of observations to reflect this improvement. However, if the node continues to behave maliciously, the observation window will still be equivalent to the original window size, and the misbehavior will be detected faster.

Figures 4.3 and 4.4 present a comparison of the effective throughput and the percent of the injected packets for the fine-grained reputation system with and without the moving window, and the two beta reputation systems. Nodes are 20% misbehaving, meaning that they will drop or maliciously manipulate 20% of the packets they handle. We see no difference between the performances of the fine-grained reputation system with and without the moving window mechanism.



NODES: % selfish %malicious	4% 36%	10% 30%	20% 20%	30% 10%	36% 4%		4% 26%	10% 20%	20% 10%	26% 4%
<f,s,m> moving window	0.8848	0.8904	0.9225	0.988	0.9981		0.9308	0.9731	0.993	0.9983
<f,s,m>	0.8848	0.8904	0.9225	0.988	0.9981		0.9308	0.9731	0.993	0.9983
<g,b> malicious	0.8446	0.8125	0.8477	0.9227	0.9957		0.9029	0.9204	0.9478	0.9935
<g,b> selfish	0.8603	0.8347	0.8586	0.9099	0.9867		0.9019	0.921	0.9685	0.9961

Figure 4.3: Effective throughput - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 20% node misbehavior.

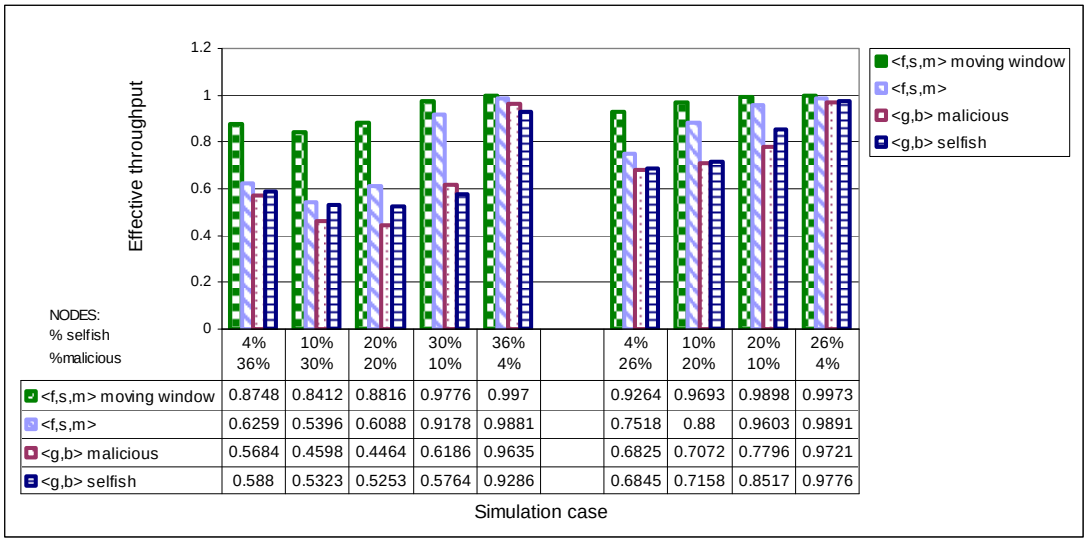


NODES: % selfish %malicious	4% 36%	10% 30%	20% 20%	30% 10%	36% 4%		4% 26%	10% 20%	20% 10%	26% 4%
<f,s,m> moving window	0.1122	0.0981	0.0654	0.0099	0.0016		0.0679	0.025	0.0064	0.0016
<f,s,m>	0.1122	0.0981	0.0654	0.0099	0.0016		0.0682	0.0253	0.0064	0.0016
<g,b> malicious	0.1543	0.1793	0.1385	0.0644	0.0036		0.0962	0.0765	0.0487	0.006
<g,b> selfish	0.1376	0.1619	0.1331	0.0795	0.0111		0.0968	0.0761	0.0287	0.0036

Figure 4.4: Percentage of injected packets - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 20% node misbehavior.

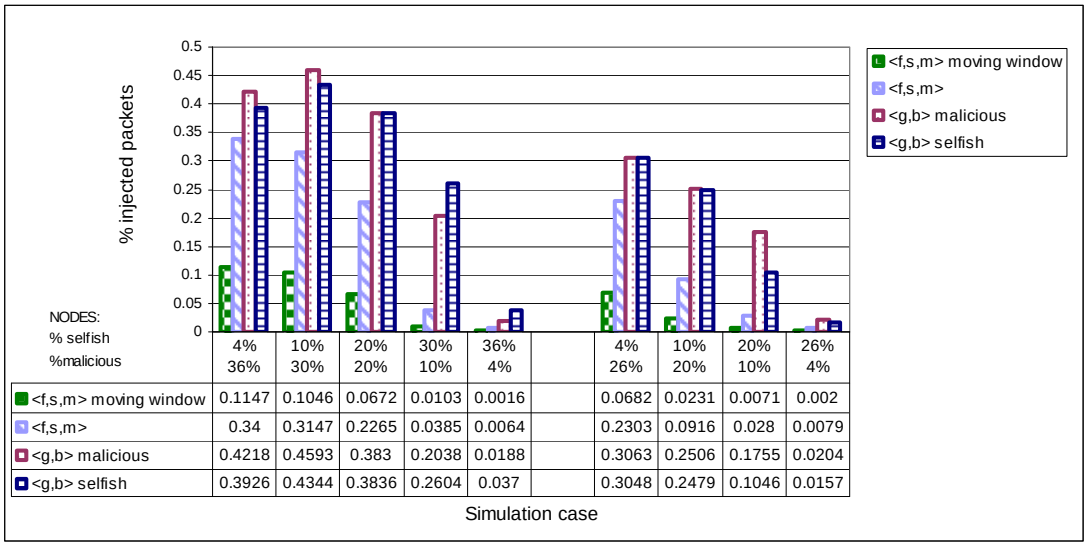
In figures 4.5 and 4.6, we increased the nodes' misbehavior to 80%. We noted a much bigger improvement in performance of the fine-grained reputation system with the moving window mechanism when compared to the fine-grainer reputation system

without it. We conclude that this difference is due to the misbehavior threshold used to engage the moving window mechanism. Thus, the moving window mechanism contributes to the effectiveness of the fine-grained system, since the threshold can be fine-tuned to be more responsive to changes in nodes' behavior.



NODES: % selfish %malicious	4% 36%	10% 30%	20% 20%	30% 10%	36% 4%		4% 26%	10% 20%	20% 10%	26% 4%
<f,s,m> moving window	0.8748	0.8412	0.8816	0.9776	0.997		0.9264	0.9693	0.9898	0.9973
<f,s,m>	0.6259	0.5396	0.6088	0.9178	0.9881		0.7518	0.88	0.9603	0.9891
<g,b> malicious	0.5684	0.4598	0.4464	0.6186	0.9635		0.6825	0.7072	0.7796	0.9721
<g,b> selfish	0.588	0.5323	0.5253	0.5764	0.9286		0.6845	0.7158	0.8517	0.9776

Figure 4.5: Effective throughput - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 80% node misbehavior.



NODES: % selfish %malicious	4% 36%	10% 30%	20% 20%	30% 10%	36% 4%		4% 26%	10% 20%	20% 10%	26% 4%
<f,s,m> moving window	0.1147	0.1046	0.0672	0.0103	0.0016		0.0682	0.0231	0.0071	0.002
<f,s,m>	0.34	0.3147	0.2265	0.0385	0.0064		0.2303	0.0916	0.028	0.0079
<g,b> malicious	0.4218	0.4593	0.383	0.2038	0.0188		0.3063	0.2506	0.1755	0.0204
<g,b> selfish	0.3926	0.4344	0.3836	0.2604	0.037		0.3048	0.2479	0.1046	0.0157

Simulation case

Figure 4.6: Percentage of injected packets - comparison for the fine-grained <f,s,m> reputation system with and without the moving window mechanism and beta <g,b> reputation systems with 80% node misbehavior.

4.2.4 Connectivity

To protect the network from the malicious nodes, a common approach reputation systems take is to temporarily isolate or disconnect these nodes. The beta reputation system, which classifies all the misbehaving nodes in one category and treats them as malicious, will lower the overall connectivity of the network if it chooses to isolate every misbehaving node for which the probability of misbehavior exceeds a certain threshold. The beta reputation system which classifies all the misbehaving nodes as selfish will show a higher overall connectivity, but also a higher percent of injected packets.

In order to compare the fine-grained and the beta reputation systems, we calculate total connectivity for each system. The calculation is performed after the

same time interval for all simulation cases. Total connectivity is calculated according to the formula:

$$C_N = \sum_{A=0}^{n} \sum_{B=0}^{n} \frac{1}{S_{AB}},$$

where C_N is the total connectivity of a network at some point in time, n is the number of nodes in the network, and S_{AB} is the reputation cost of reaching node B from node A as observed on node A and given that A and B are neighbors. If node A and B are not neighbors, then the cost is infinite and the connectivity between them is zero. A higher total connectivity value indicates that the overall reputation cost is lower. Figure 4.7 illustrates the difference in connectivity values among the cases in the 40% simulation set.

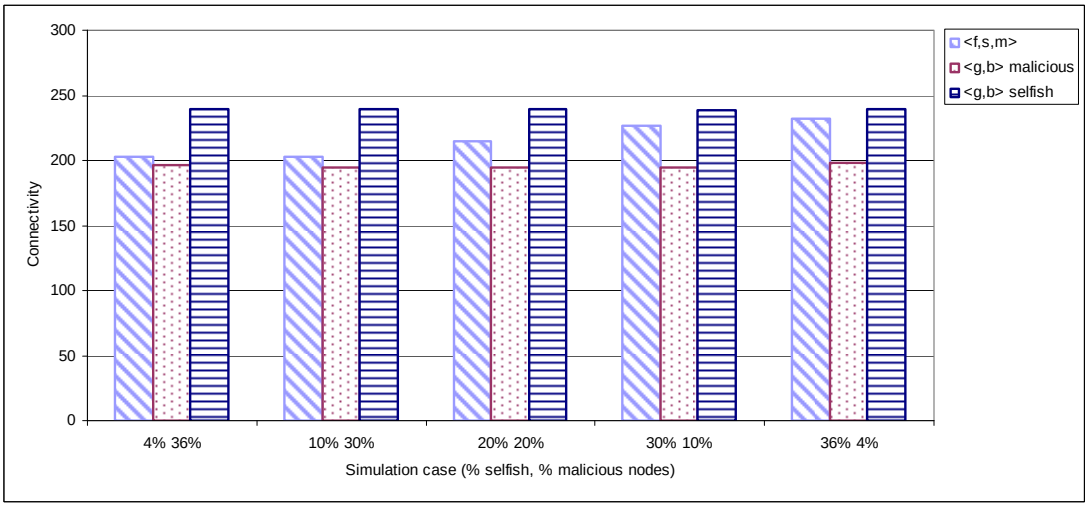


Figure 4.7: Total connectivity of the network - comparison for the fine-grained <f,s,m> and beta <g,b> reputation systems over the cases in the 40% simulation set.

As defined earlier, the reputation value in the fine-grained reputation system is calculated as a vector <*f, s, m*> where *f* corresponds to the friendly component, *s* to the

selfish, and m to the malicious and $f+s+m = 1$. The beta reputation system only has two components, $<g, b>$, where $b = s+m$, and $g+b = 1$.

In the fine-grained reputation system, the cost of reaching a selfish node is $1 < c \leq 2$, and the cost of reaching a malicious node is $101 < c \leq 102$. The cost c is calculated as $1+s$ for the selfish behavior, and $101+m$ for the malicious. In the beta reputation system which considers all misbehaving nodes as malicious, the cost to all misbehaving nodes is $101 < c \leq 102$, and c is calculated as $101+b$. Conversely, in the beta reputation system which considers all misbehaving nodes as selfish, the cost to all misbehaving nodes is $1 < c \leq 2$, and c is calculated as $1+b$

Suppose that the entire misbehaving population of the network consists of selfish nodes. Suppose also that the misbehaving threshold for isolating a node from the network is set to $m=0.2$, or equivalently in the beta system that treats all misbehaving nodes as malicious to $b=0.2$. This corresponds to cost $c=101.2$ in both systems. Given that there exists a selfish node for which $s \geq 0.2$, this node will not be isolated in the fine-grained reputation system as its cost is determined to be $1.2 \leq c \leq 2$. In the beta reputation system the cost for the same node will be $101.2 \leq c \leq 102$, and therefore the node will be punished and isolated. From this example if follows that we can generally expect the beta reputation system that treats all misbehaving nodes as malicious to have lower network connectivity than the fine-grained reputation system given that selfish nodes are present in the network, and that their cost exceeds the isolation threshold

4.2.5 Discussion of Results

Overall, the simulation results support our claim that the fine-grained reputation system is more effective than either of the two variations of the beta reputation system. When compared to the beta system which treats all misbehaving nodes as selfish, the fine-grained reputation system lowers the number of injected packets in the system, or more generally, lowers the activity of malicious nodes. When compared to the beta system which treats all misbehaving nodes as malicious, the fine-grained system results in higher network connectivity because it does not use the same strategy for both selfish and malicious nodes.

In the presented simulation cases, selfish and malicious nodes correspondingly drop or inject approximately 20% packets for every 50 packets they forward. The nodes only misbehave when they are a part of a packet route in use. The data exchange pattern among nodes is random, thus it is not guaranteed that all sender-receiver combinations are going to be repeated or even used.

Under these conditions, we observed the following improvements: over all simulation cases the average percent increase in the effective throughput for the fine-grained reputation system was 2.84% when compared to the beta reputation system that treats all nodes as malicious, and 2.5% when compared to the beta reputation system that treats all nodes as selfish; the average percent decrease in the number of injected packets was 2.76% when compared to the beta system that treats all nodes as malicious and 2.49% when compared to the beta system that treats all nodes as selfish.

Over the simulation set that contained 40% of misbehaving nodes, the fine-grained reputation system showed an average percent increase in connectivity of 9.55% when compared the beta reputation system that treats all nodes as malicious. When we compared the fine-grained system with the beta system that treats all nodes as selfish, we observed an increase in connectivity of 10.56% for the beta system. However, the tradeoff was an average of 4.69% more injected packets and 4.64% decrease in the effective throughput of the beta system.

Without a physical implementation or a simulation that accounts for physical aspects of the environment and each ad hoc node, it is not possible to evaluate overhead introduced by the addition of the fine-grained reputation system on an existing ad hoc routing protocol. We define the reputation system overhead as the following: control packets required by the reputation system, memory for storage of the reputation data, CPU time required for reputation related tasks (calculations and monitoring), and battery power required for promiscuous mode operation and dissemination of the reputation information among neighbors. The overhead also depends highly on the underlying routing protocol.

Although we currently do not have an adequate mechanism to evaluate the fine-grained reputation system overhead, the existing beta-reputation systems' implementations and performance analysis can be used to derive an estimated overhead. When compared to the beta systems, the fine-grained reputation system introduces one additional variable, namely the third reputation vector component.

Thusly we would expect a minimal increase in computational and memory requirements. Pirzada et al. offer a detailed performance analysis of a beta reputation system implemented on three wireless ad hoc routing protocols, TORA, AODV, and DSR. A detailed discussion of the results presented in (Pirzada 2006) is beyond the scope of this work. Pirzada et al. support our statement that the performance of the fine-grained reputation system will very much depend on the underlying routing protocol – greater routing overhead introduced by the protocol itself causes greater overall overhead. This inevitably leads to packet collisions, packet loss, and decrease in the quality of service. Intuitively from (Pirzada 2006), it follows that greater overall mobility of the network will require more power for route discovery. Additionally, a network with higher mobility will require more frequent distribution of reputation information - in order for nodes to adapt they need to receive reputation information from newly discovered neighbors. Since the overhead metrics depend on the implementation of the fine-grained reputation system itself, and the properties of any given ad hoc network, we suggest a detailed overhead analysis of the fine-grained reputation system as future work.

Chapter V

CONCLUSION

Most wireless ad hoc routing protocols assume cooperation among the ad hoc nodes. However, in reality, there is no guarantee that ad hoc nodes will behave in non-malicious and cooperative way. Various types of reputation systems have emerged in recent years as an attempt to address security and reliability of wireless ad hoc networks. We propose a fine-grained reputation system to address the problem of node misbehavior. Our solution makes distinction between selfish and malicious misbehavior, as opposed to reputation systems that are only able to classify nodes as good and misbehaving (bad). Our reputation system integrates the Dirichlet distribution with Bayes' theorem, which allows for finer granularity of node classification and updating of the reputation based on new knowledge. Our system is distributed and collaborative in a sense that nodes acquire new knowledge through first-hand observations of their neighbors' behavior, and through the reports they receive from other nodes containing their first-hand observations. The fine-grained reputation system includes a trust component based on the Beta distribution in order to be able to prevent nodes from spreading false reputation information.

Whereas systems that classify nodes in good and bad treat all misbehaving nodes in the same way, we propose different strategies for different classes of misbehavior: we do not forward data packets for selfish nodes until they improve their

behavior, and we temporarily isolate malicious nodes and make their reputation redemption slower than in the case of selfish nodes. We show that our system is able to improve network connectivity and decrease the number of maliciously modified or injected packets in the network when compared to the beta reputation systems.

For future work, one of the biggest issues in reputation systems is node identity. An identity mechanism for ad hoc networks has to be resistant to identity spoofing attacks and needs to support identity persistence over some time (Buchegger 2008). We think that investigating a real incentive scheme for selfish nodes in the fine-grained reputation system could contribute to even better performance of the system. Incorporating the fine-grained reputation system with a routing ad hoc protocol such as DSR in an effective and efficient way would allow us to test the system in reality and to be able to tell what kind of performance tradeoff our system requires in exchange for providing security and reliability to the ad hoc network.

REFERENCES

Bechler, M. H., H.J. Kraft, D. Pahlke, F. Wolf, L. (2004). A cluster-based security architecture for ad hoc networks. INFOCOM 2004, IEEE.

Bolstad, W. M. (2007). Introduction to Bayesian statistics, Wiley-Interscience.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y., Jetcheva, J. (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. International Conference on Mobile Computing and Networking, Dallas, TX, ACM.

Buchegger, S., Le Boudec, J. (2002a). Performance analysis of the CONFIDANT protocol: Cooperation Of Nodes – Fairness In Dynamic Ad-hoc NeTworks. IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, Switzerland.

Buchegger, S., Le Boudec, J. (2003). The effect of rumor spreading in reputation systems for mobile ad-hoc networks. WiOpt '03, Sophia-Antipolis, France.

Buchegger, S., Mundinger, J., Le Boudec, J. (2008). Reputation systems for self-organized networks: lessons learned. IEEE Technology and Society Magazine, Special Issue on Limits of Cooperation in Wireless Communications.

Buchegger S., L. B. J. (2004). A robust reputation system for peer-to-peer and mobile ad hoc networks. P2PEcon, Harvard University, Cambridge, MA.

Buttyan, L., Hubaux, J. (2001). Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks. Lausanne, Switzerland, Institute for Computer Communications and Applications, Swiss Federal Institute of Technology.

Buttyan, L., Hubaux, J. (2002). "Stimulating cooperation in self-organizing mobile ad hoc networks." Mobile networks and applications **8**(5).

Cahill, V. G., E. Seigneur, J.-M. Jensen, C.D. Yong Chen Shand, B. Dimmock, N. Twigg, A. Bacon, J. English, C. Wagealla, W. Terzis, S. Nixon, P. Di Marzo Serugendo, G. Bryce, C. Carbone, M. Krukow, K. Nielson, M. (2003). "Using trust for secure collaboration inm uncertain environments." Pervasive Computing, IEEE **2**(3).

Cemerlic, A., Yang, L. (2006). Trust-based routing in wireless ad hoc networks. ACM Middle-Southeast Conference, Gatlinburg, TN.

Cormen, T. H., Leiserson, C.E., Rivest, R.L., Stein, C. (2001). Introduction to algorithms, The MIT Press.

Dimmock, N., Belokosztolszki, A., Eyers, D., Bacon, J., Moody, K. (2004). Using trust and risk in role-based access control policies. Ninth ACM symposium on Access control models and technologies, Yorktown Heights, NY, ACM.

Feller, W. (1968). An introduction to probability theory and its applications, Wiley.

Fink, D. (1997). A compendium of conjugate priors. Bozeman, MT, Montana State University.

Haghpanah, N., Akhoondi, M., Kargar, M., Movaghar, A. (2007). Trusted secure routing for ad hoc networks. MobiWac '07, Chania, Crete Island, Greece, ACM.

Hu, T. C., Shing, M.T. (2002). Combinational algorithms, enlarged second edition, Dover Publications Inc. .

Jakobsson, M., Hubaux, J., Buttyan, L. (2003). A micro-payment scheme encouraging collaboration in multi-hop cellular networks. Proceedings of Financial Crypto, La Guadeloupe.

Jaramillo, J. J., Srikant, R. (2007). DARWIN: distributed and adaptive reputation mechanism for wireless ad-hoc networks. MobiCom '07, ACM.

Jøsang, A., Ed. (2007). Trust and reputation systems. Foundations of security analysis and design IV. Bertinoro, Italy, Springer.

Josang, A., Haller, J. (2007). Dirichlet reputation systems. ARES '07. Vienna, Austria.

Lang, D. (2008). Routing protocols for mobile ad hoc networks - classification, evaluation and challenges, VMD Verlag Dr. Mueller e.K.

Laniepce, S., Demerjian, J., Mokhtari, A. (2006). Cooperation monitoring issues in ad hoc networks. IWCM, Vancouver, Canada.

Mahmoud, A., Sameh, A., El-Kassas, S. (2005). Reputed Authenticated Routing for Ad hoc Networks protocol (Reputed-ARAN). PE-WASUN '05, Montreal, Canada, ACM.

Marti, S., Garcia-Molina, H. (2006). "Taxonomy of trust: categorizing P2P reputation systems." Computer Networks **50**(4).

Marti, S., Giuli, T.J., Lai, K., Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. MobiCom '00, Boston, MA, ACM.

Mendenhall, W., Beaver, R.J., Beaver, B.M. (2008). Introduction to probability and statistics, Duxbury Press.

Michiardi, P., Molva, R. (2002a). CORE: A COllaborative Reputation mEchanism to enforce node cooperation in mobile ad hoc networks. The IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security.

Michiardi, P., Molva, R. (2002b). Game theoretic analysis of security in mobile ad hoc networks. Sophia-Antipolis, France, Institut Eurecom.

Mundinger, J., Le Boudec, J. (2005). Analysis of a reputation system for mobile ad-hoc networks with liars. WiOpt '05, Trentino, Italy.

Neapolitan, R. E. (2003). Learning Bayesian networks, Prentice Hall.

NetworkWorkingGroup (2003). RFC 3561: Ad hoc On-Demand Distance Vector (AODV) routing. **RFC 3561**.

NetworkWorkingGroup (2007). RFC 4728: The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for IPv4. **RFC 4728**.

Ngai, E. C. H., Lyu, M.R., Chin, R.T. (2004). An authentication service against dishonest users in mobile ad hoc networks. Aerospace Conference, IEEE.

Pirzada, A. A. M., C. Datta, A. (2006). "Performance comparison of trust-based reactive routing protocols." IEEE Transactions on Mobile Computing **6**(5).

Rackley, S. (2007). Wireless networking technology: from principles to successful implementation, Newnes.

Raiffa, H., Schlaifer, R. (1961). Applied statistical decision theory, Harvard University Press.

Rebahi, Y., Mujica, V. E., Simons, C., Sisalem, D. (2005). SAFE: Securing pAcket Forwarding in ad hoc nEtworks. 5th Workshop on Applications and Services in Wireless Networks. Paris, France.

SearchCIO-Midmarket.com. (2008). "Definition: hop." from http://searchcio-midmarket.techtarget.com/sDefinition/0,,sid183_gci212253,00.html#.

Sterbenz, J. P. G., Krishnan, R., Hain, R.R., Jackson, A.W., Levin, D., Ramanathan, R., Zao, J. (2002). Survivable mobile wireless networks: issues,

challenges, and research directions. The 1st ACM workshop on Wireless security, Atlanta, GA, ACM.

Traupman, J. (2007). Ad hoc. The Bantam new college Latin & English dictionary, revised edition, Bantam.

Wang, W., Li, X., Wang, Y. (2004a). Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. MobiCom '04, San Diego (CA), ACM.

Wang, Y., Giruka, V. C., Singhal, M. (2004b). A fair distributed solution for selfish nodes problem in wireless ad hoc networks. Ad-Hoc, Mobile, and Wireless Networks, Springer Berlin / Heidelberg. **3158:** 211-224.

Yang, L., Kizza, J.M., Cemerlic, A., Liu, F. (2007). Fine-grained reputation-based routing in wireles ad hoc networks. IEEE International Conference on Intelligence and Security Informatics, New Brunswick, NY, IEEE.

Yang, L., Novobilski, A, Ege, R. (2008). "Trust-based usage control in collaborative environment." The International Journal of Information Security and Privacy **2**(2).

Yuen, W. H., Lee, H., Andersen, H. (2008). "A simple and effective cross layer networking system for mobile ad hoc networks." from www.cacs.louisiana.edu/~wu/619/presentations/xuyang.ppt.