A BAYESIAN GAME APPROACH TO COEXISTENCE WITH MALICIOUS AND SELFISH NODES IN WIRELESS AD-HOC NETWORKS

By

J. ANDREW ROLES

A SENIOR RESEARCH PAPER PRESENTED TO THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE OF STETSON UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

STETSON UNIVERSITY
2014

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

ABSTRACT

A BAYESIAN GAME APPROACH TO COEXISTENCE WITH MALICIOUS AND SELFISH NODES IN WIRELESS AD-HOC NETWORKS

By

J. Andrew Roles

Apr 2014

Advisor: Dr. Hala ElAarag and Dr. Erich Friedman
Department: Mathematics and Computer Science

Mobile Ad-hoc networks are self-organized systems of nodes or installations, all cooperating to provide network functions such as routing and forwarding. Utilized in open environments, mobile ad-hoc networks are vulnerable to attack by malicious nodes, causing harm or disorder. These nodes do not reveal their identities while disrupting service. Thus, early detection is important. The network may also contain selfish nodes, installations that choose to conserve power resources rather than provide network function. Identification of selfish nodes, too, is necessary so that functional nodes do not waste resources attempting to communicate with them. Malicious node detection has previously been modeled as a Bayesian game with imperfect information. In this attacker/defender game the defender is unsure of the type of its opponent and must select strategies based on this incomplete information. Malicious nodes attempt to avoid detection by masquerading as regular nodes, providing useful network function at interval. This small contribution to the network may, however, be entirely necessary in a mobile ad-hoc environment with extremely limited resources and selfish nodes. Thus, exploiting the malicious node may be a viable option. In this paper we demonstrate that selfish and malicious nodes can be successfully identified through our proposed attacker/defender game. In addition we show that once identified, a malicious node may be exploited if the benefit it provides to the network is greater than the damage accrued. In this paper we propose a more robust model that is capable of identifying any type of installation found in a mobile ad-hoc environment. Our technique is more

advantageous because it conserve power resources and improves network performance compared to previous works.

I. INTRODUCTION

Mobile ad-hoc networks (MANETs) are self-organized systems in which network operation is provided through collaboration of nodes within a neighborhood. Each node agrees to perform network functions such as routing and forwarding with other nodes in the network, without prior trust. Typically, all nodes are programmed to maximize a particular utility function by choosing strategies to maximize a payoff. Usually, a node's utility function is the benefit that node derives from other nodes in the network. That is, providing useful and efficient communication between nodes is any individual node's utmost priority. There may, however, be nodes in the network with other utility functions. A selfish node may choose not to forward packets as a means to conserving power. This node's utility function would then be to minimize its power usage. This would degrade network throughput, make routing almost impossible, or in the worst case immobilize the network as a whole. It is important to note that selfish nodes may only choose to communicate with a certain probability, based on the node's current power constraints. They do not necessarily always decline communication. It is possible that there exist some nodes in the network whose utility function is to maximize harm and disorder to the network. These nodes, called malicious nodes, can mount attacks that compromise individual nodes or degrade the overall network performance. All the while, they avoid being caught and isolated. Malicious nodes do not reveal their identities, but rather try to conceal them, masquerading as traditional nodes.

To counter malicious nodes and promote proper network activity, there must be nodes that monitor and evaluate their neighbors. Nevertheless, detection and differentiation between types of nodes has challenges. First, monitoring neighboring nodes is expensive. Monitoring entails a node listen to the channel and process the information sent by nodes under suspicion. This consumes both processing and reserve power, thus continuous monitoring is impractical. To improve monitoring efficiency, a game-theory based approach is suggested. Given the current

state of the channel of communication, games would determine whether a node should monitor or not monitor the channel. Second, malicious nodes can disguise themselves. To reduce the likelihood of being detected, a malicious node can behave like a regular node and attack the network only when it believes it is not being monitored. Moreover, malicious nodes have the additional strategy of fleeing to avoid isolation. That way the node can return to its malicious behavior with a new location and clean history in the network. This strategy is referred to as "Hit and Run" and comes with its own associated cost [9]. That is, the energy spent to change locations. Third, the randomness and unreliability of the wireless channel will inflate the uncertainty of the monitoring and detection process. In some cases it is nearly impossible to determine whether a packet was dropped as a result of wireless error or malicious intent.

In any case, detection is not our ultimate goal. We must next determine how to respond to malicious and selfish nodes. In most cases, a malicious node is isolated upon detection. However, there may be situations in which the malicious nodes can be kept in the network and made use of, coexistence. We rely on the fact that a malicious node does not know if it has been identified or not. Hence, the node will continue to provide useful network function under the assumption that it is avoiding detection. Thus we can exploit the node to improve network throughput as long as the benefits to the network outweigh the damage. In a network with limited resources, the benefit malicious nodes involuntary supply can be substantial and necessary. These networks have limited numbers of nodes and possibly many selfish nodes, unwilling to cooperate. However, this gives monitoring nodes the additional task to determining when to terminate its coexistence and isolate the opposing node. A malicious node is isolated and banned from the network when the damage it inflicts on the network outweigh the involuntary benefit it provides. A selfish node will be banned when the likelihood that it will participate in network activity drops below a particular threshold.

We model the interactions between nodes as an attacker/defender game of incomplete information. The defender, or monitoring node, attempts to glean the type of the attacker: selfish or malicious based only on its actions. Game theory approaches are taken to select strategies for the defending node based on its current belief of the type of its opponent. Unlike previous work, we consider the attacking player to be of either malicious or selfish type. Others, such as Liu et al [10] and Wang et al [11] differentiate between nodes of malicious or regular type. They define regular nodes as nodes that always choose to participate. We believe that our model is more robust as our definition of selfish nodes can include nodes that always participate as well as nodes that participate only a portion of the time. Others researchers, consider only opponents which are either selfish or not [2-4, 6, 8]. We show that monitoring nodes are able to accurately distinguish between selfish and malicious types of nodes and respond accordingly. In addition, nodes will be evaluated to decide whether they should be included in the network or banned from all activity.

The rest of this paper is organized as follows. In Section II, we discuss recent research in this area of work. Section III introduces our proposed Bayesian game of detection. In Section IV we derive a Bayesian Nash Equilibrium from our detection game. Section V presents our Dynamic Bayesian game. In Section VI we present our simulation and results. Section VI discusses a fault tolerant improvement to the model. Finally, Section V concludes the paper.

II. RELATED WORKS

Game theory has proven as an excellent tool for both modeling and solving problems in wireless networks. Problems such as agent negotiation [12], routing algorithms [13], risk assessment [14], access admission control [17], congestion control [18], and even wireless jamming [20] have all been modeled or solved using various game theory techniques. Roy et al [16] provides an excellent survey paper of game theory as applied to network security. The paper details the theory behind a number of popular games and reveals examples of how they are used in network security. Uses include detecting cyber intrusion, denial of service attacks, and even intrusion detection in mobile ad-hoc networks. Another interesting survey paper comes from Agrawal and Lingawar [19], who discuss gray hole, wormhole, blackhole attacks, and other attacks in mobile ad-hoc networks. Unlike the previous survey paper, Agrawal does not detail game theory techniques, but reviews how to analyze and detect various attacks using the tool NS2. The most informative survey paper, however, comes from Manshaei et al [21]. They bring substantially more detail to the problems solved by game theoretic techniques and introduce a number of new topics including security at the MAC layer, cryptography, and anonymity. Srivastava et al adds yet another survey of game theory as applied to wireless ad-hoc networks, however, presents little information unattainable from the other survey papers [26].

There has been a great deal of research on understanding the selfish nature of nodes in ad-hoc networks. Recall, a selfish node is one that chooses to decline communication with other nodes as a means of saving power and self preservation. We contest that it is important to identify these nodes since they waste the resources of neighboring nodes attempting to communicate with them. Monitoring nodes must evaluate selfish installations, in addition to malicious ones, and determine whether they are benefitting or impeding the network as a whole. Komali et al [24] discusses the selfish behavior of nodes in ad-hoc and mesh networks and suggests a topology control algorithm that examines energy efficiency in the network and forces cooperation of particular nodes. Urpi [8] too discusses selfishness in ad-hoc networks and also concludes that

forcing communication between nodes is the only effective method of ensuring throughput in the network. Rather than forcing communication for selfish nodes, Marden and Effros suggest a game theoretic approach [25]. Under their model, the input is the number of wireless transmissions required in the problem. The distributed algorithm then determines the best way to satisfy the input while minimizing the cost to individual nodes. However, we are not only interested in selfish behavior, but malicious behavior as well.

Several works have studied the incentives nodes have to cooperate with their neighbors [1-3]. These works, however, model a selfish or malicious node as never cooperative. This model is too simple and inapplicable in real world applications. Many others focus on modeling cooperation and selfishness in a network using game-theoretic approaches [4-8]. In these games, nodes decide whether to forward or not forward a packet based on a cost and benefit model. Their cost being the energy consumption necessary to forward the packet and benefits being improved network throughput and collaboration with neighboring nodes. In each, they show that enforcing cooperation between nodes can improve throughput, but nodes may exhaust their power storage and retreat from the network. These works, however, do not consider the existence of malicious nodes capable of disguising their presence by providing useful network function. It is unrealistic to consider a malicious node as always attacking or a selfish node as always declining. We must consider the multitude of actions these type of nodes may choose to take. Liu and Zang develop a game theoretic model in that is capable of inferring the intent, objectives, and strategies of an attacker in a wireless ad hoc network [22]. They found that the generality of their model allowed for its application in a variety of types of attacks and even tested the system on a denial of service attack. We would like to cultivate a similar level of generality in our model, as the type and severity of a malicious attack could come in a variety of forms.

Li and Wu suggest a dynamic Bayesian game framework to analyze the interactions between regular and malicious nodes in MANETs [9]. In this system regular nodes form beliefs

to evaluate the type of its opponent, refusing communication with identified malicious nodes. The malicious node regularly evaluates the risk of being caught and chooses when to flee and restart at another location. The paper, however, considers only interactions between two nodes. Their game is not applicable in a multi-user, real-world environment. Liu et al also suggest a Bayesian game [10]. In their model the attacker seeks to inflict the most damage without being detected and the defender tries to monitor and isolate nodes while conserving energy expenditure. They demonstrated that detection is feasible in both static and dynamic Bayesian models, concluding that a hybrid strategy is most effective. Theodorakopoulos and Baras present a similar work. Their model, however, appears over simplified as it only considers two factors in the decision making process: benefit to the network and energy expenditure [23]. Wang et al demonstrates a more robust model, though considers only the case of two nodes as do [9] and [10]. Wang develops a malicious node detection system played by a regular node and malicious node [11]. Both players' strategies are based on Bayesian games of imperfect information in addition to a post-game played by the regular node upon detection to allow for coexistence with malicious nodes. Simulation results show that the perfect Nash Equilibrium achieved in the post-game helps to extend the length of games and improve the throughput of the network.

Unlike previous work, we study the interactions malicious and selfish types of nodes, together, in an ad-hoc network. This generalizes the model as every node in an ad-hoc network is inherently selfish. That is, every node seeks to maximize its particular payoff function. As well, we suggest a method of determining the benefit and damage a particular node inflicts on the network so that beneficial selfish and malicious nodes can be utilized to improve the throughput of the network.

III. GAME MODEL

We consider a MANET which contains a number of nodes connected with each other. Nodes can dynamically leave or join the network during movement. We assume authentication measures are in place, e.g. public-key based authentication. When a node newly joins the network, other nodes authenticate the node and set their beliefs toward the newcomer to an initial value. We distinguish between three types of nodes: benevolent, malicious, and selfish nodes. The actions of each node are rational and are governed by their underlying utility function. A rational action may be to refuse cooperation with other nodes if battery constraints become too limited. Table 1 presents the types of nodes and their associated pure strategies.

Table 1: Pure strategy profiles for each type of node

Type of Node:	***Pure Strategies:***
Benevolent	Monitor, Idle
Malicious	Attack, Disguise
Selfish	Participate, Decline

We define a benevolent node as one which always cooperates in network function and regularly monitors the channel. A benevolent node may choose to ban nodes that have been identified as malicious if the damage they cause to the network is greater than the benefit they provide. Similarly, selfish nodes may be banned from the network for not participating enough. Forcing cooperation between benevolent nodes and others allows the nodes more opportunity to update their beliefs about opponents. These nodes, therefore, require either extended battery life or a

permanent power source. A benevolent node has two pure strategies: Monitor or Idle. Each has an associated cost to the node. Monitoring, for instance is a costly measure (e.g., consumes the node's power) and nodes cannot afford to monitor all of the time. We must therefore develop a method in which benevolent nodes monitor only part of the time. A node who has evaluated the safety of the channel may choose to remain idle and conserve power for a short period. However, careful monitoring may reveal a trusted node to in fact be malicious.

Malicious nodes seek to cause damage and disorder to the network; however they can avoid detection by disguising themselves as regular nodes. Thus, its pure strategies are Attack and Disguise. A malicious node attacks to waste resources and disrupt network operation. Attacks can come in a variety of forms including denial-of-service (DoS) at different network layers, packet dropping, and routing disruption at the network layer. We must therefore develop a method of measure of how much damage is inflicted by various attacks. Like benevolent nodes, malicious nodes form beliefs about other nodes in the network. The malicious node tracks the benevolent node's trust opinion, evaluating the risk of being caught. A malicious node seeks to disguise itself if it is monitored by a benevolent node and attack the network if it is not monitored.

Finally, we are left with selfish nodes. These nodes never intentionally cause harm to the network, but may reduce the effectiveness of the network by choosing to not participate with other nodes. Thus, the pure strategies for selfish nodes are Participate or Decline. Some nodes may choose to never participate in network functions such as forwarding packets and routing. These nodes only slow down the network and waste the resources of other nodes attempting to communicate with it. Thus, they should be kicked from the network to prevent further waste. On the other hand, there may be selfish nodes that always participate in the network. Monitoring these nodes would be a waste of resources, so we do so rarely. However, most selfish nodes will strike a balance, participating in the network part of the time and retreating within itself the rest of

the time. Like the two previous types of nodes, selfish nodes have profit and costs associated to their actions.

To simplify the interactions among the nodes, we consider a two-player game played by nodes, i and j. The types of these nodes are private information, not available to the opponent. Since the type of each player is hidden, and observation of the opponent is not accurate, it is a Bayesian game with imperfect information. Bayesian games are a combination of game theory and probability theory that allow incomplete information to be taken into account and influence future decisions [9]. In these games, players are allowed to have some private information (e.g., the type of the node in question) that affects the progress of the game. Players form beliefs about the private information of their opponent and respond accordingly. Their beliefs are represented as probability distributions and are updated using Bayes' rule as new information is learned. Figure 1 illustrates the extensive form of the static Bayesian detection game.

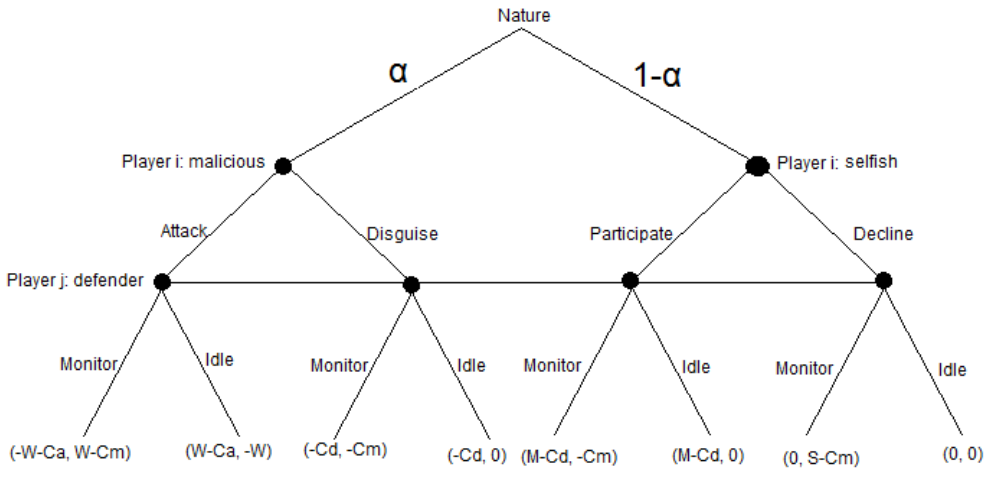

Figure 1: Extensive form of static Bayesian Detection Game

As our detection game will always be played between two nodes with incomplete information, we look to a special category of Bayesian game called a signaling game. A signaling game is played between a sender and receiver. The sender has a certain type and knows the type of the receiver. Based on this information the receiver sends a message. However, the receiver does not know the type of the sender and can only observe the messages sent to it. The receiver must observe the messages sent to it and determine what actions to take in response. In our detection game, the sender, node i, can be malicious with probability α, or selfish with probability $(1- \alpha)$. In our detection game the sender knows that the receiving node is benevolent. It also knows that the receiving node believes the sender is malicious with probability α. The receiver/benevolent node knows only its current belief about the type of the sender and must choose a response based on this incomplete information. Once the benevolent node is certain of the type of its opponent, it will evaluate the benefit and damage that node causes. If the damage to the network is greater than the benefit, the node will be banned. Otherwise, the identified node will remain in the network and continue to be evaluated.

Table 2: Summary of notation used in model

Symbolic Notation:	***Definition:***
Cm	Cost associated with monitoring and analyzing the channel
Ca	Cost of mounting an attack against the network
Cd	Cost of providing effective useful network operation
W	Measure of the degree of damage inflicted to the network by malicious nodes
M	Measure of the benefit to the network provided by regular nodes participating
S	Measure of the benefit of detecting a selfish node

To further construct the game we define the following values. Let W refer to the payoff of a malicious node if it successfully attack. The cost of mounting such an attack is Ca. In the case that a malicious node chooses to disguise itself in the network it will incur a cost of Cd. Likewise, regular nodes incur the same cost for providing network function. Let M refer to the payoff of a regular node if it chooses to participate in the network at the time it is monitored. For the defending node, j, the cost of monitoring the channel is given by Cm. S is the benefit a defending node receives from successfully detecting a regular node being selfish. Table 2 illustrates the notation used to represent the profits and costs of nodal interactions.

The actions chosen by nodes are given by their action profiles a_i and a_j for nodes i and j, respectively. Hence, for the action profile (a_i, a_j) = (Attack, Idle), the utility for a successful attack by node i is W-Ca. The loss for node j is –W since j is idle and cannot detect the attack. Similarly, if the action profile is (a_i, a_j) = (Attack, Monitor), the attacking node i loses W+Ca. The node suffers a loss because it chooses to attack at the same time it is being monitored. The defending node j will gain W-Cm since it is monitoring at the time of the attack and thus, detected the attack. This can be seen in Table 2, which illustrates the costs and profits associated with the interactions between benevolent and malicious nodes. Notice that the payoffs in Table 3 indicate that node j benefits the most from playing Monitor if i plays Attack and from choosing Idle if j chose Disguise. Similarly, node i benefits the most by playing Attack when j plays Idle and by choosing Disguise when j plays Monitor.

Table 4 indicates the strategic form of our static Bayesian detection game where node i is selfish and node j is benevolent. The payoffs indicate that Participate is i's best response to j playing Monitor and that Decline is i's best response to j playing Idle. Likewise, Monitor is j's best response to i playing Decline and Idle is j's best response to i playing Participate. We can see that if the action profile is (Participate, Monitor) then node j receives a payoff of M-Cd. The node

benefits M because it will gain better standing with j for participating in network function. It loses Cd for providing that function. The payoff of sender i is M-Cm since it too benefits from useful network function and loses power by monitoring. We assume this will always be a positive number. That is, M > Cm. This assumption simplifies our later pure strategy Nash Equilibrium. Notice that the action profile (Decline, Idle) results in a zero payoff for both nodes. Neither can be penalized for simply doing nothing.

Table 3: Strategic form of detection game where i=malicious and j=benevolent

	Monitor	***Idle***
Attack	-W-Ca, W-Cm	W-Ca, -W
Disguise	-Cd, -Cm	-Cd, 0

Table 4: Strategic form of detection game where i=selfish and j=benevolent

	Monitor	***Idle***
Participate	M-Cd, -Cm	M-Cd, 0
Decline	0, S-Cm	0, 0

IV. BAYESIAN NASH EQUILIBRIUM (BNE) ANALYSIS

We begin our analysis on the detection game from the extensive form of the static Bayesian game as illustrated in Figure 1. In our attacker/defender game, the attacker would like to play a Bayesian strategy to minimize his chances of being detected and the defender would like to play a Bayesian strategy in order to maximize his chance of detecting attacks. Since we are playing a signaling game, the attacking player, i, knows that the type of its opponent is benevolent. Node j, however believes that node i is regular with probability (1-α) and malicious with probability α. To solve this game, we are interested in finding the possible Bayesian Nash Equilibrium (BNE). We define a Bayesian Nash Equilibrium as a strategy profile in which each player's prescribed strategy is a best response to the strategies of the other players in the Bayesian game [27]. In other words, neither player can change strategies and improve their payoff. Nash Equilibrium can be reached in either a static context in which players always choose one particular strategy or in a mixed context, where players choose strategies with certain probability. In a static game, the BNE is the Nash Equilibrium given the beliefs of both nodes.

First, we will consider only the pure strategies. If player i plays his pure strategy pair (Attack if malicious, Participate if selfish), then the expected payoff of defender j playing its pure strategy Monitor is

$$E_j(Monitor) = \alpha(\mathrm{W} - \mathrm{Cm}) + (1-\alpha)(-\mathrm{Cm}). \quad (1)$$

Likewise, j's expected payoff of playing its pure strategy Idle is

$$E_j(Idle) = \alpha(-W) + (1-\alpha)(0). \quad (2)$$

Setting these two equations equal to each other we get

$$\alpha = \frac{Cm}{2W}. \quad (3)$$

Therefore, if, $E_j(Monitor) > E_j(Idle)$, or if $\alpha > \frac{Cm}{2W}$, then the best response of player j is to play Monitor. However, if defender j plays Monitor, Attack will no longer be the best response

for the malicious type of player i, and he will move to play Disguise instead. Therefore, {(Attack if malicious, Participate if selfish), Monitor} is not a BNE. But, if $\alpha < \frac{Cm}{2W}$, then the best response for defender j is to play Idle. If this were the case, then Participate would no longer be the best strategy for the selfish type of player and he would move to play Decline. Recall, M > Cm. Otherwise, We would have a Bayesian Nash Equilibrium. Thus, {(Attack if malicious, Participate if selfish), Idle} is a Bayesian Nash Equilibrium.

Now we consider the pure strategy pair (Attack if malicious, Decline if selfish) played by player i. The expected payoff of defender j playing its pure strategy Monitor is

$$E_j(Monitor) = \alpha(\mathrm{W} - \mathrm{Cm}) + (1 - \alpha)(\mathrm{S} - \mathrm{Cm}) \quad (4)$$

and the expected payoff of playing its pure strategy Idle is

$$E_j(Idle) = \alpha(-W) + (1 - \alpha)(0). \quad (5)$$

Again, setting these equal to each other gives

$$\alpha = \frac{Cm-S}{2W-S}. \quad (6)$$

Thus, if $E_j(Monitor) > E_j(Idle)$, or if $\alpha > \frac{Cm-S}{2W-S}$, then the best response for player j is to play Monitor. However, like the previous case, the defender will move to play Disguise and so {(Attack if malicious, Decline if selfish), Monitor} is not a BNE. If we have $\alpha < \frac{Cm-S}{2W-S}$, then the best response for player j is to play Idle. This is a Bayesian Nash Equilibrium since the best response of player i is to either continue playing Attack, or to continue to play Decline. So {(Attack if malicious, Decline if selfish), Idle} is a BNE.

We now know that if the malicious player i always plays Attack, then the only Bayesian Nash Equilibrium exists where the regular type plays Decline and the best response by defender j is to play Idle. Now we must consider those situations in which the malicious type of player always plays Disguise. First we consider the pure strategy pair (Disguise if malicious, Decline if selfish) for player i. The expected payoff for j playing the pure strategy Monitor would then be

$$E_j(Monitor) = \alpha(-\mathrm{Cm}) + (1-\alpha)(\mathrm{S}-\mathrm{Cm}) \quad (7)$$

and the expected payoff of playing its pure strategy Idle is

$$E_j(Idle) = \alpha(0) + (1-\alpha)(0). \quad (8)$$

Setting these two equations equal to each other we get

$$\alpha = \frac{Cm-S}{S}. \quad (9)$$

Therefore, if $E_j(Monitor) > E_j(Idle)$, or if $\alpha > \frac{Cm-S}{S}$, then the best response of player j is Monitor. This is a Bayesian Nash Equilibrium because neither the malicious or selfish type of player can do better by changing strategy. However, if we have that $\alpha \leq \frac{Cm-S}{S}$., then the best response for player j is to play Idle. If player j chooses to play Idle, then the best response of the malicious type of player is to switch to Attack. Hence, {(Disguise if malicious, Decline if selfish), Idle} is not a BNE and {(Disguise if malicious, Decline if selfish), Monitor} is a BNE.

Finally, the last pure strategy combination we must consider for player i is (Disguise if malicious, Participate if selfish). If the defending node determines the best response to player i's strategy is to play Idle, then the malicious type of node would move to play Attack instead. If the defending node deems its best response to play Idle, then we have again a BNE.

We have shown that there exists a number of Bayesian Nash Equilibrium for particular pure strategy profiles meeting certain criteria. Now we seek to find a mixed-strategy BNE for the cases that did not result in a pure strategy BNE. For this we must introduce two new belief probabilities to our defender and one to our attacker. Let p denote the belief of defender j about probability with which player i will play Attack and q be its belief of the probability with which player i will play Participate. Let r represent the attacking player's belief about the likelihood that the defender will play Monitor. Hence, the expected payoff for player j playing Monitor is

$$E_j(Monitor) = \alpha p(W-Cm) + \alpha(1-p)(-Cm) + (1-\alpha)q(-Cm) + (1-\alpha)(1-q)(S-Cm) \quad (10)$$

and the expected payoff of defender j playing Idle is

$$E_j(Idle) = \alpha p(-W) + \alpha(1-p)(0) + (1-\alpha)q(0) + (1-\alpha)(1-q)(0). \quad (11)$$

By imposing $E_j(Monitor) = E_j(Idle)$, we get the malicious player should play Attack with probability p and that the regular type of player should play Participate with probability q so that

$$2\alpha Wp + (\alpha Cm - S)q = \alpha(Cm + S) + Cm - S. \quad (12)$$

We determine the probability that a malicious node will Attack similarly. The expected value of a malicious node playing Attack is

$$E_i(Attack) = r(-W - Ca) + (1-r)(-W), \quad (13)$$

and the expected value of playing Disguise is

$$E_i(Diguise) = r(-Cd) + (1-r)(-Cd). \quad (14)$$

As before, we impose $E_i(Attack) = E_i(Disguise)$, and get that the defending player should play Monitor with probability $r = -\frac{Cd-W}{Ca}$. In the case that $r \leq 0$, then node i should play attack with probability 1. Recall Table 3. Notice that the payoffs for a selfish node are the same whether its opponent plays Monitor or Idle. Thus, the probability that j will play Monitor does not affect nodes of the regular type. Thus, the strategy pair ((p if malicious, q if selfish), r, α) is a mixed-strategy BNE if we have that

$$2\alpha Wp + (\alpha Cm - S)q = \alpha(Cm + S) + Cm - S. \quad (15)$$

Table 5: List of Pure Strategy Nash Equilibrium Found

Pure Strategy Nash Equilibrium
{(Attack if malicious, Participate if selfish), Idle}
{(Attack if malicious, Decline if selfish), Idle}
{(Disguise if malicious, Participate if selfish), Monitor}
{(Disguise if malicious, Decline if selfish), Monitor}

In summary, our static Bayesian detection game has four pure-strategy Bayesian Nash Equilibrium. Table 5 details the four pure-strategy equilibrium found. Notice these form two situations. One, the malicious type of node plays Attack and the defending node plays Monitor. Two, the malicious type of node plays Disguise and the defending node plays Monitor. These situations result in pure strategy BNEs, regardless of the play made by the selfish type of node. There also exists a mixed strategy Bayesian Nash Equilibrium for which the defender j plays Monitor with a probability derived from its beliefs p, q, and α and the attacker i plays Attack with a probability based on its belief, r, and neither player can improve their payoff by changing strategies. In the next section, we will derive these probability functions.

V. BELIEF UPDATE AND DYNAMMIC BAYESIAN GAMES

The previously described static Bayesian game is a one-stage game in which both attacker and defender attempt to maximize their payoff based on a fixed prior belief set about the type of their opponent. In addition we have illustrated the equilibria associated with these prior belief sets. Due to the difficulty of assigning accurate probabilities for each player's beliefs, we extend the static Bayesian game to a multi-stage dynamic Bayesian game in which each player updates its belief probabilities in response to actions taken by its opponent and its previously held belief.

We assume that the static Bayesian game is repeatedly played at each time slot t_k, where k = 0, 1, 2, … The payoffs of the players in each stage game have no discount. That is to say that the payoffs will remain the same for every stage game. In addition, an arbitrary interval of T seconds may be selected for each stage game and we consider the game to have an infinite horizon as any node will not know when its neighboring nodes will leave the network. Furthermore, we assume that the identities of players do not change as the game progresses. Thus, our model relies on authentication measures to counteract impersonation attacks, spoofing, and Sybil attacks [28]. During each stage, players will interpret all incoming messages from neighboring nodes and update their probability distributions according to Bayes' theorem.

We construct our belief updating rules based on Bayes' theorem. For each node, its belief {benevolent:{α: probability of malicious opponent, (1-α): probability of selfish opponent} and malicious:{r: probability of opponent monitoring, (1-r): probability of opponent not monitoring}} can be determined by its most recently updated belief and the interpretation of actions. We do not consider the case of a selfish node because selfish nodes do not form beliefs about their neighbors. Thus, we write the belief of a benevolent node at the $(t+1)^{th}$ stage as:

$$\alpha_{(t+1)} = \begin{cases} P(malicious \mid attack) = \frac{\alpha_t \, p_t}{\alpha_t \, p_t + (1-\alpha)_t \, (0)} = 1 \\ P(malicious \mid disguise) = \frac{\alpha_t \, (1-p)_t}{\alpha_t \, (1-p)_t + (1-\alpha)_t \, q_t} \end{cases} \text{ or } \quad (16)$$

$$(1-\alpha)_{(t+1)} = \begin{cases} P(selfish \mid participate) = \frac{(1-\alpha)_t\, q_t}{\alpha_t\,(1-p)_t + (1-\alpha)_t\, q_t} \\ P(selfish \mid decline) = \frac{(1-\alpha)_t\,(1-q)_t}{\alpha_t\,(1-p)_t + (1-\alpha)_t\,(1-q)_t} \end{cases} . \quad (17)$$

We can then use these updated probabilities to recalculate the supplementary belief set {p: probability of attack, (1-p): probability of disguise, q: probability of participate, (1-q): probability of decline} as follow:

$$p_{(t+1)} = P(attack \mid \alpha_t) = \frac{p_t\,\alpha_t}{p_t\,\alpha_t + (1-p)_t\,\alpha_t} \text{ and} \quad (18)$$

$$q_{(t+1)} = P(participate \mid (1-\alpha)_t) = \frac{q_t\,(1-\alpha)_t}{q_t\,(1-\alpha)_t + (1-q)_t\,(1-p)_t} . \quad (19)$$

Similarly, we write the belief set of a malicious node at the $(t+1)^{th}$ stage as:

$$r_{(t+1)} = \begin{cases} P(monitor \mid participate) = \frac{r_t\, q_t}{r_t\, q_t + (1-r)_t\, q_t} \\ P(monitor \mid decline) = \frac{r_t * 0}{r_t * 0 + (1-r)_t\,(1-q)_t} = 0 \end{cases} \quad (20)$$

$$(1-r)_{(t+1)} = \begin{cases} P(idle \mid participate) = \frac{(1-r)_t\, q_t}{(1-r)_t\, q_t + r_t * 1} \\ P(idle \mid decline) = \frac{(1-r)_t\,(1-p)_t}{(1-r)_t\,(1-p)_t + r_t * 0} = 1 \end{cases} . \quad (21)$$

These equations for building and updating belief probabilities constitute what we refer to as a *belief system*. We define a *belief system* to be a function that assigns each belief probability distribution over the histories in the set. Every node's set of belief distributions are assigned initial values at the start of the game. Players update this set by observing actions in the current stage game and the previous belief it holds. Beliefs are the result of previous situations and can be backtracked to the initial belief and action observed. Therefore, the current belief set and observed action can fully represent the histories in the information sets, and those information sets can be reached with positive probabilities if the strategies are carefully designed.

VI. Simulation and Results

The strategies outlined in the previous sections have been implemented on a custom simulator based on ds, developed by Dr. Li Wu [29]. Wu's simulator contains methods only for generating arbitrary numbers of nodes and placing them within a rectangular area. Thus, we took on the responsibility of implementing communication between neighboring nodes via Directed Sequence Distance Vector Routing in addition to our model specifications. Simulations are conducted in randomly generated MANETs. We assume that any node only has access to the packets directly addressed to it.

A specified number of wireless nodes are randomly placed in a 900*m* X 900*m* region with a transmission range of 300*m*. Once placed, nodes communicate with neighbors to build routing tables by broadcasting network probe packets. Each node stores the minimum number of hops to reach the destination node as well as the next node in the path. If a node receives a packet destined for a neighbor it refers to its routing tables and forwards the packet to the next node in the path to the destination node.

Each simulation is repeated 500 times, and the average is taken and used for results. We set the number of malicious nodes in the network to 20, the number of benevolent nodes is 20, and the remaining 20 nodes are of selfish type. For every benevolent node, its beliefs are initialized to α=0.2, p=0.5, q=0.5. The beliefs of the malicious nodes are initialized to r=0.5, q=0.5. In addition, the constants (which do not change throughout the simulation) are set to Cm=3, Ca=5, Cd=1, W=8, M=2, and S=4. Simulations showed that changing these constants had little effect on the results.

In Figure 2, we illustrate the change in belief of any particular benevolent node, i, interacting with any malicious node, j. The figure illustrates two plots for the belief of node i over time. The first plot details the belief set of a benevolent node updated with our static model, while the second plot describes the same belief updated using our dynamic model.

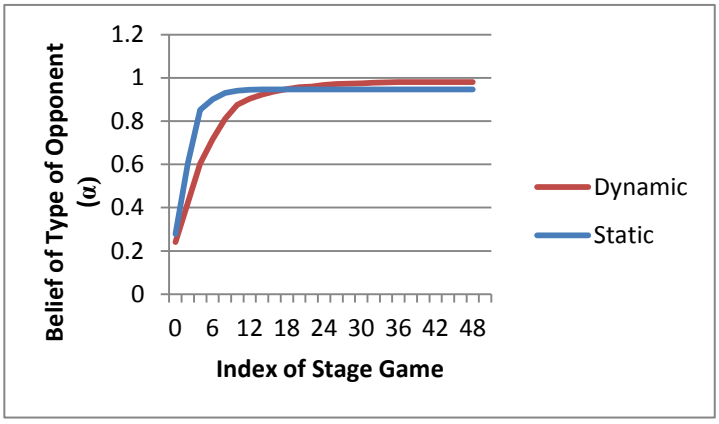

Figure 2: Belief of a benevolent node about the type of its opponent (malicious) as the game progresses

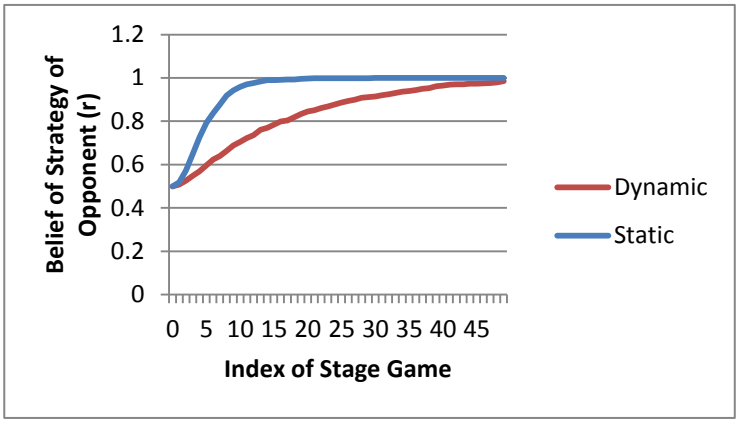

Figure 3: Belief of a malicious node about the strategy of its opponent (benevolent) as the game progresses

Notice, both plots quickly reach a steady state value of about .95, indicating a strong likelihood that the opposing installation is malicious. Our static model plot reaches this steady state earlier as continuous monitoring of opponents offers more chances to detect malicious activity. Nevertheless, our dynamic is able to identify the malicious installation only a few stage games behind its static counterpart. This tells us that our dynamic model is capable of identifying malicious installations within a reasonable amount of time. Figure 3, on the other hand, plots the belief of a particular malicious installation as it interacts with benevolent nodes over time. The graph shows us that malicious nodes are capable of identifying their monitoring counterparts within a timely manner as well. Notice, the malicious node identifies its opponent to be likely Monitoring earlier in the static model than in the dynamic one. This makes sense as the benevolent player monitors constantly in the static model. This gives the malicious player more opportunities to recognize its opponent's strategy.

Now that we are certain that our model allows for identification of malicious and benevolent entities, we will compare it to other relevant works. First, we must define some terminology. We define goodput (G) to be the quotient of the number of legitimate packets and the total number of packets to travel through the network. throughput (T) is defined as the total number of legitimate packets divided by time. A higher value for throughput indicates better

performance in the network and higher values in goodput point to less waste of network resources. In Figures 4 and 5 we analyze the goodput and throughput

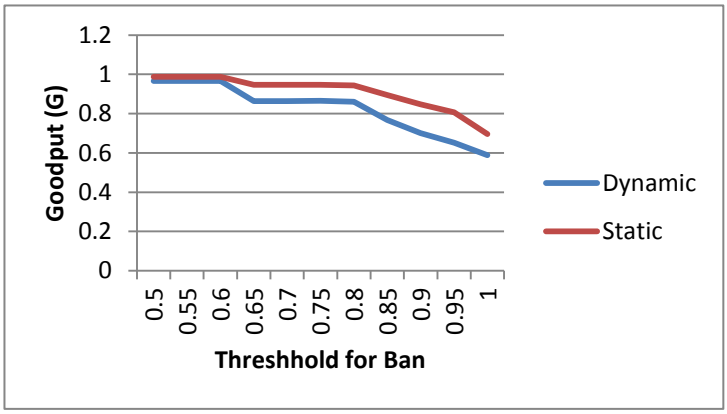

Figure 4: Goodput as the Threshhold for banning possibly malicious installations increases

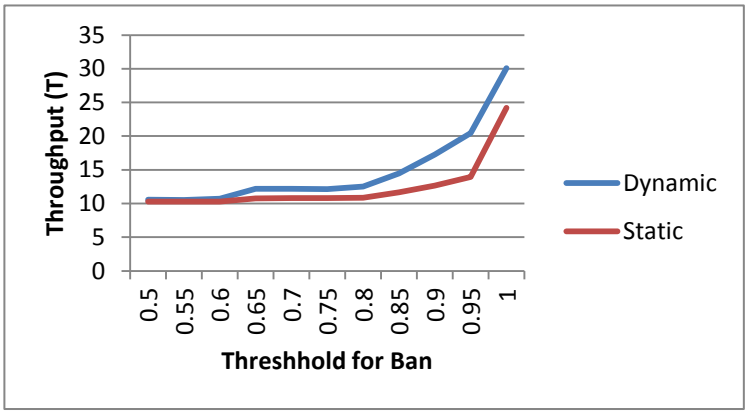

Figure 5: Change in Throughput as the Threshhold for banning possibly malicious installations increases

of our network as the threshold for banning malicious installations varies. The threshold for banning is a maximum tolerance level allowed for a benevolent node's belief p, the likelihood that the opposing player will play attack. If p reaches a certain threshold, we ban the installation. Figure 4 plots the goodput in relation to the ban threshold. Notice that goodput falls as the threshold increases. If we have a low value for the threshold, then nodes are banned as soon as they take any malicious actions, thus goodput is high since malicious installations are removed from the network quickly. However, if the threshold is high, then the malicious nodes are left in the network longer and goodput degrades as there are more attack packets in circulation. Our proposed dynamic model is only slightly worse than our static model as non-continuous monitoring allows malicious nodes to stay in the network slightly longer.

Figure 5 indicates the changes in throughput as the threshold for banning varies. However, we see the opposite effect as the previous. That is, throughput increases as the threshold rises. Consider the case in which the threshold is 0.5. The Throughput is low because malicious nodes will be excluded from the network early on in the game. Thus, we are losing all of the legitimate packets generated from malicious nodes attempting to hide within the network.

Conversely, if the threshold is set to 1.0, then malicious nodes will stay in the network much longer before being removed. Hence, the Disguise strategy of these installations will increase the number of legitimate packets and therefore, improve throughput. This is our motivation behind our goal of coexistence. If we only exclude those installations that attack the network regularly we can significantly improve throughput in comparison to models that instantly ban nodes that have been identified as malicious. For this reason, we fix the threshold in future results at 0.75 so that we balance the gain in throughput with the loss in goodput.

In Figures 6 and 7 we show the effect of varying the number of benevolent nodes in the network on the goodput and throughput. The number of devices of malicious or selfish type is equally divided between 40 remaining installations. Figure 6 details goodput. Notice that

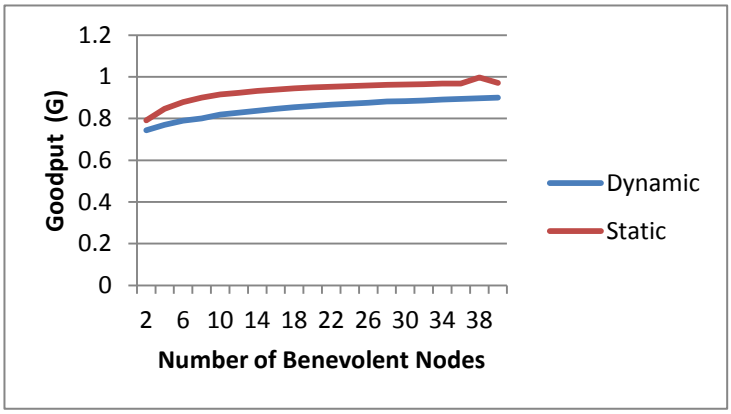

Figure 6: Goodput as the number of benevolent nodes varies

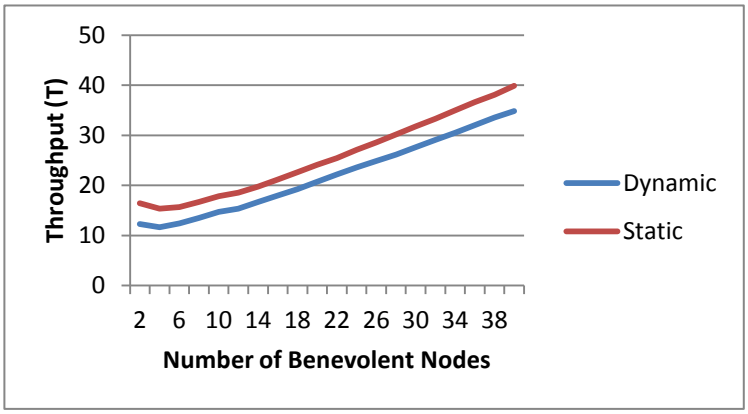

Figure 7: Throughput as the number of benevolent ndoes varies

goodput increases as the number of benevolent nodes increases. This makes sense as an increase in the population of monitoring installations will reduce the time that malicious nodes are left in the network, thus reducing the number of non-legitimate packets. Similarly, throughput increases as the number of benevolent nodes rises. This increase in the number of legitimate packets per unit of time is the result of two factors. First, benevolent nodes generate or forward only legitimate messages. Second, increasing the number of benevolent nodes reduces the time that malicious installations exist in the network. Thus, reducing the number of legitimate packets lost to malicious attacks. Therefore, we can conclude that increasing the number of monitoring nodes improves the network as a whole.

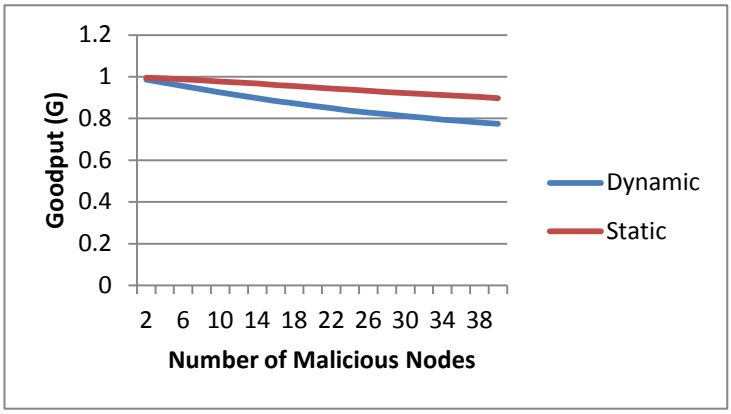

Figure 8: Goodput as the number of malicious nodes varies

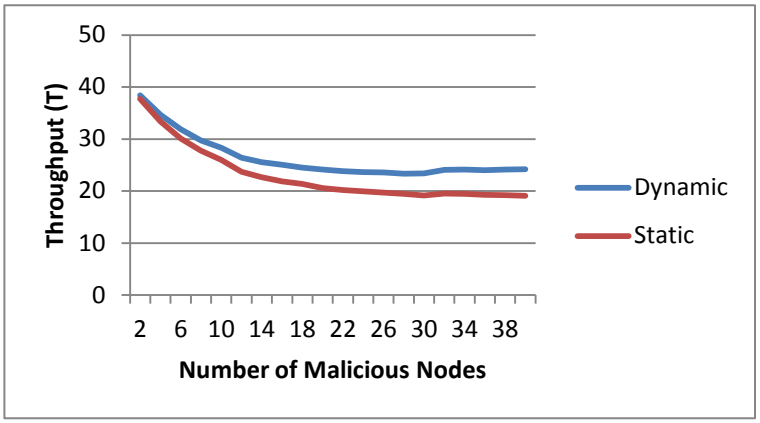

Figure 9: Throughput as the number of malicious nodes varies

We show the effect of changing the number of malicious nodes in the network on throughput and goodput in Figures 8 and 9. In both cases, the number of benevolent nodes is held constant at 20, along with 20 selfish installations. Goodput decreases as the number of malicious nodes increases. This is the result of an increase in the total number of non-legitimate packets in the network originating from the rising number of malicious nodes. Likewise, throughput decreases for the same reason.

Notice in Figures 2-9, our dynamic model performed slightly worse than our static model. We consider these acceptable losses as the goal of our dynamic model is to conserve power by monitoring only a portion of the time. Thus, it is necessary to show that our model

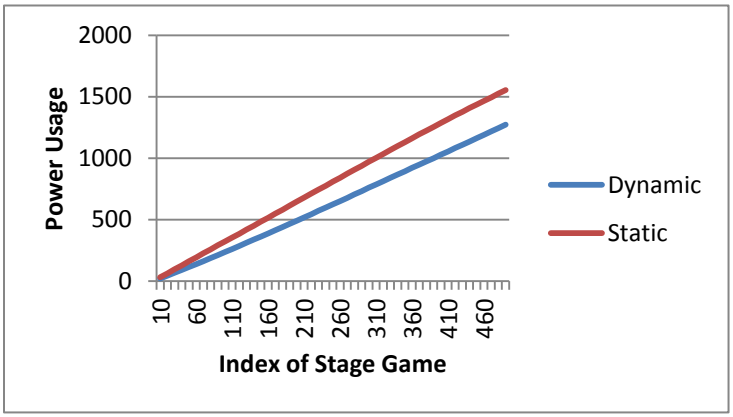

Figure 10: Total power usage of a benevolent node as the game progresses

does, in fact conserve power. Figure 10 shows the total power usage of a particular benevolent node over time. Recall, the data is the result of an average over 500 iterations. It is clear that at any point in time, the total power usage generated by our dynamic model is lower than that of the static. The difference between these two rises as the length of the game increases. Therefore, we have shown that our model achieves adequate values for goodput and throughput in addition to conserving power for benevolent installations.

VII. Fault Tolerant Dynamic Bayesian Games

At this point we would like to introduce a few other constants to improve the accuracy of the model. Let β denote the probability of successfully identifying an attack packet and Ψ denote the probability of successfully identifying a decline message. Including these new parameters means our Bayesian Nash Equilibria will change. For instance, consider the pure strategy pair (Attack if malicious, Participate if selfish), then the expected payoff of the defending (monitoring) player is

$$E_j(Monitor) = \alpha\beta(\mathrm{W} - \mathrm{Cm}) + \alpha(1-\beta)(-CM) + (1-\alpha)(-\mathrm{Cm}) \quad (22)$$

Likewise, the player's expected payoff of playing its pure strategy Idle is

$$E_j(Idle) = \alpha(-W) + (1-\alpha)(0). \quad (23)$$

Setting these two equations equal to each other we get

$$\alpha = \frac{Cm}{W(1+\beta)}. \quad (24)$$

Therefore, if, $E_j(Monitor) > E_j(Idle)$, or if $\alpha > \frac{Cm}{W(1+\beta)}$, then the best response of the defending player is to play Monitor. The other pure and mixed strategies can be calculated similarly.

We will also have to include these new parameters in the belief updating system we described in Section V. Thus, we recalculate the belief of a benevolent node at the (t+1)th stage as:

$$\alpha_{(t+1)} = \begin{cases} P(\alpha_{(t+1)} = malicious \mid attack) = \frac{\alpha_t\, p_t\, \beta_t}{\alpha_t\, p_t\, \beta_t + \alpha_t\, p_t\, (1-\beta_t) + \alpha_t\, (1-p_t)(1-\beta_t)} \\ P(\alpha_{(t+1)} = malicious \mid disguise) = \frac{\alpha_t\, (1-p)_t\, \beta_t}{\alpha_t\, (1-p)_t\, \beta_t + \alpha_t\, (1-p)_t\, (1-\beta_t) + (1-\alpha)_t\, q_t} \end{cases} \text{ or } \quad (25)$$

The other probabilities can be updated in the same matter. Including these two parameters improves the accuracy of the model by accounting for the instability of wireless networks and the difficulty of identifying attacks. We refer to this improved model as the fault tolerant dynamic Bayesian game.

VIII. Conclusion

In this paper we have discussed the importance of identifying both malicious and selfish nodes in wireless ad-hoc networks. We have developed a model based on Bayesian games with incomplete information that is capable of modeling interactions of any type of node found in an ad-hoc environment. Additionally, we have found four pure-strategy Bayesian Nash Equilibrium in which neither attacker or defender can change their strategy to improve their payoff. As well, we establish a mixed-strategy Bayesian Nash Equilibrium in which players choose actions with a particular probability based on the likely probability distribution of their opponent. In either case we were able to verify the existence a Nash Equilibrium between the attacking and defending players in which no player can advance their payoff. We proved through rigorous simulation that our proposed benevolent node is capable of determining the types of nodes in the network within a reasonable timeframe while conserving power. Additionally, we showed that throughput can be improved by coexisting with malicious nodes that seldom attack. We also developed a Fault Tolerant model that better represents the uncertainty of wireless networks. In the future we would like to improve the accuracy of the model by allowing nodes to dynamically move throughout the network. We would also like to explore the additional strategy of malicious nodes to flee from the network to avoid punishment.

REFERENCES

[1] A. Blanc, Y. Liu, and A. Vahdat. Designing incentives for peer-to peer routing. In Proc. of IEEE INFOCOM, 2005.

[2] L. Buttyan and J. Hubaux. Stimulating cooperation in self-organizing mobile ad-hoc networks. ACM Mobile Networks and Applications, 8(5), 2003.

[3] M. Felegyhazi, J. Hubaux, and L. Buttyan. Nash equilibria of packet forwarding strategies in wireless ad-hoc networks. IEEE Transactions on Mobile Computing, 5(5), 2006.

[4] V. Srinivasan, P. Nuggehalli, C-F. Chiasserini, and R.R. Rao. An analytical approach to the study of cooperation in wireless ad-hoc networks. IEEE Transactions on Communications, March 2005.

[5] Y. Xiao, X. Shan, and Y. Ren. Game theory models for IEEE 802.11 DCF in wireless ad-hoc networks. IEEE Radio Communications, March 2005.

[6] J. Cai and U. Pooch. Allocate fair payo® for cooperation in wireless ad-hoc networks using Shapley value. In Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), page 219, April 2004.

[7] P. Nurmi. Modelling routing in wireless ad-hoc networks with dynamic Bayesian games. In Sensor and Ad-hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference, pages 63{70, October 2004.

[8] A. Urpi, M. Bonuccelli, and S. Giordano. Modelling cooperation in mobile ad-hoc networks: A formal description of selfishness. In WiOpt'03 Workshop: Modeling and Optimization in Mobile, Wireless Networks, March 2003.

[9] F. Li and J. Wu. Hit and Run: A Bayesian game between regular and malicious nodes in MANETs. In Sensor, Mesh and Ad-hoc Communications and Networks, June 2008.

[10] Y. Liu, C. Camaniciu, and H. Man. A Bayesian game approach for intrusion detection in wireless ad-hoc networks. In GameNets '06 Proceedings from the 2006 workshop on Game theory for communications and networks, 2006.

[11] W. Wang, M. Chatterjee, and K. Kwiat. Coexistence with malicious nodes: a game theoretic approach. In GameNets '09 Proceedings from the International Conference on Game Theory and Networks, May 2009.

[12] Z. Bian, J. Luo. A Cooperative Game Model for Agent Negotiation in Network Service. In Proceedings of the 10th International Conference on Compuer Supported Cooperative Work in Design, 2006.

[13] J. Su, W. Liu, and K. Yue. A Network Routing Algorithm Based on the Coalitional Game Theory. In International Conference on Computational Intelligence and Natural Computing, 2009.

[14] W. He, C. Xia, C. Zhang, Y. Ji, and X. Ma. A Network Security Risk Assessment Framework Based on Game Theory. In Second International Conference on Future Generation Communication and Networking, 2008.

[15] S. Buchegger and J. Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In the National Competence Center in Research on Mobile Information and Communication Systems, 2004.

[16] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandily, and Q. Wu. A Survey of Game Theory as Applied in Network Security. In Proceedings of the 43rd Hawaii International Conference on System Sciences, 2010.

[17] D. Charilas, O. Markaki, and E. Tragos. A Theoretical Scheme for Applying Game Theory and Network Selection Mechanisms in Access Admission Control. In 3rd International Symposium on Wireless Pervasive Computing, 2008.

[18] Z. Li and Z. Jin. A Wireless Ad-hoc Network Congestion Control Algorithm based on Game Theory. In the International Conference on Future Computer Science and Applications, 2011.

[19] S. Agrawal and R. Lingawar. Applications of NS2 to Overcome Computer Network Attacks. In World Research Journal of Computer Architecture, 2012.

[20] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang. Coping with a Smart Jammer in Wireless Networks: A Stackelberg Game Approach. In IEEE Transactions on Wireless Communications, 2013.

[21] M. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J. Hubaux. Game Theory Meets Network Security and Privacy. In ACM Computing Surveys, 2011.

[22] P. Liu and W. Zang. Incentive-Based Modeling and Inference of Attacker Intent, Objectives, and Strategies. In ACM Transactions on Information and System Security, 2005.

[23] G. Theodorakopoulos and J. Baras. Malicious Users in Unstructured Networks. IN IEEE International Conference on Computer Communications, 2007.

[24] R. Komali, A. MacKenzie, and P. Mahonen. On Selfish, Local Information, and Network Optimality: A Topological Control Example. In Proceeding of 18th International Conference on Computer Communications and Networks, 2009.

[25] J. Marden and M. Effros. The Price of Selfishness in Network Coding. In IEEE Workshop on Network Coding, Theory and Applications, 2009.

[26] V. Srivastava, J. Neel, A. MacKenzie, R. Menon, L. DaSilva, J. Hicks, J. Reed, and R. Gilles. Using Game Theory to Analyze Wireless Ad Hoc Networks. In IEEE Communications Surveys and Tutorials, 2005.

[27] J. Watson. Strategy: An Introduction to Game Theory. In W. W. Norton & Company, 2008.

[28] J. Douceur. The Sybil attack. In First International Workshop on Peer-to-Peer Systems, 2002.

[29] L. Wu. DS Simulator. Available *at* http://sourceforge.net/projects/wrss