ON MODELING ENERGY-SECURITY TRADE-OFFS FOR DISTRIBUTED MONITORING IN WIRELESS AD HOC NETWORKS

Seyhun Mehmet Futaci, Katia Jaffrès-Runser and Cristina Comaniciu
Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ
Email: sfutaci, Cristina.Comaniciu, Katia.runser@stevens.edu

ABSTRACT

In this paper, we propose a game theoretic framework for distributed intrusion detection in ad hoc networks. The proposed models capture the energy-security tradeoffs for individual monitoring and propose an energy efficient design for distributed monitoring, with probabilistic guarantees for the achieved security level in the network.

1. INTRODUCTION

With the rapid advances in research on wireless technology, wireless ad hoc networks have become an attractive choice for both commercial and military applications. In these networks, security and energy efficiency are of primary concern. To ensure security, Intrusion Detection Systems (IDSs) [1]-[4],[7], are usually deployed in the network to detect malicious activity. Instead of using an end-to-end intrusion detection, better network reactivity is obtained when the IDS is deployed at the nodes to perform local monitoring of the traffic. For instance in [7], the Medium Access Control (MAC) traffic is monitored by some nodes of the network. Normal behavior in the neighborhood of a mobile node is characterized by a 'safe-mode' training stage. Misbehaving nodes are then detected through MAC layer anomalies using cross-feature analysis on feature vectors constructed from the training data.
Since these more reliable IDS systems are deployed at the node level, they yield additional energy expenditure for the nodes. For a more energy efficient network design, the monitoring burden can be shared among the nodes participating in the network. For example, several works (e.g., [1],[3]) have suggested a cluster based IDS architecture. For instance in [1], a cluster is formed among a group of nodes where every pair of members can communicate with a direct wireless link, and a cluster-head is then elected randomly from the cluster. A periodical re-election may be engaged to improve fairness. The cluster head selection described above ensures a shared responsibility for monitoring, but it may still lead to unequal energy levels across nodes, unless a more sophisticated election scheme is employed that specifically accounts for current energy levels in nodes. However, a more elaborate election mechanism, may lead to excessive communication between the nodes, which in turn will increase the energy spent on overhead. A distributed solution is thus highly desirable.

In this work, we propose such a distributed solution based on a game theoretic formulation. In this framework, each node decides to monitor or not independently, aiming to maximize a utility function which represents a balance between the gains obtained by monitoring and the energy costs involved. Since the results of the monitoring are shared with the entire neighborhood, an important issue of selfishness arises, yielding a problem similar with the classic tragedy of the commons scenario.

Our main contributions in this work are: (a) determine a practical energy cost metric for Intrusion Detection Systems based on the number and type of instructions in an IDS algorithm; (b) propose an energy efficient, distributed monitoring protocol based on a non-cooperative game framework.

The paper is organized as follows. In section 2 we summarize our findings on the energy cost estimation method. In section 3 we present a dynamic Bayesian game model for cooperative sensor networks. Section 4 summarizes our simulation results and section 5 presents related work. Conclusions are presented in Section 6.

2. ENERGY-SECURITY TRADEOFFS FOR INTRUSION DETECTION

Security and energy efficiency are key measures of performance in wireless networks. However, in order to ensure security, IDS monitoring must be deployed, which often consumes significant energy. In this section, we try to approximate the energy consumption due to IDS monitoring for a wireless node. Our approach is to quantify the energy expenditure of running a software algorithm on a microprocessor, as a function of the software complexity.

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

2.1. FIRST ORDER ENERGY ESTIMATION MODEL

Based on the CMOS Power Consumption model, given a CPU instruction, if we know the supply voltage level, the number of gate toggling for an instruction and the capacitive load being switched in each bit toggling, we can find the energy consumption resulting from the execution of this instruction. Unfortunately, the CPU instructions and the capacitive load of a particular circuitry in a CPU that is active during the execution of a specific instruction are not published in the processors' data sheets. Hence, an analytical computation of the energy consumption is not possible.

For a specific Microprocessor architecture, it is possible to measure the energy consumption of each particular instruction [6] and reuse it to order the instructions regarding their level of energy consumption. The work of Sinha et al [6] on the Intel StrongARM processor shows that to a first order approximation, the current consumption of a piece of code is independent of the code, and depends only on the operating voltage and frequency of the processor. Consequently, we can use the following first order software energy estimation model:

$$E_{tot} = V_{DD} \cdot I_0(V_{DD}, f) \cdot \Delta t \quad \textbf{(1)}$$

where, E_{tot} is the total energy consumed in executing the program, *Vdd* is the supply voltage, Δt is the program execution time and $I_0\ (V_{dd},f)$ is the supply current at the given V_{dd} level and the given operating frequency *f*.

We have conducted extensive experiments using Freescale Semiconductor's MC9S08GT60 Micro-controller, which led us to believe that the above result applies to a general class of microcontrollers used in wireless ad-hoc sensor networks as well. Consequently, we conjecture that we can approximate the energy consumption metric as being determined mainly as a function of the execution time Δt of the programs, given *Vdd* and $I_0\ (Vdd,f)$ in (1).

2.2. EXECUTION TIME (Δt) OF A PROGRAM

The execution time Δt of a specific program is directly related to the time complexity of the associated algorithm. Using the time complexity function, we can use the following equation for finding the execution time Δt of a program written in a high level language (e.g. C programming language):

$$\Delta_t = \frac{t(n).N.c}{f}, \quad \textbf{(2)}$$

where *t(n)* is the time complexity function giving the total number of step counts (a step is either an addition, a multiplication, a comparison, etc…), *n* is the instance characteristic, *N* is the average number of machine instructions per step count, *c* is the average number of machine cycles per machine language instruction and *f* is the operation frequency of the computing platform.
From (1) and (2), a complete first order energy equation can be written as:

$$E_{tot} = V_{DD} \cdot I_0(V_{DD}, f) \cdot \frac{t(n).N.c}{f} \quad \textbf{(3)}$$

Since this formula uses an average value for N, it only gives a first approximation of the energy consumption. However, to get a more precise estimation, the value of *t(n)* can be modified to account for the different number of instructions a statement is using on the targeted CPU.
When we have the parameters and the time complexity function in place we can predict the energy consumption of a program for different problem sizes without having to run the program for different instances. We can also incorporate new time complexity functions for new algorithms, and find out their approximate energy consumptions.

2.3. ENERGY CONSUMPTION FOR AN EXAMPLE IDS

The intrusion detection algorithm used as an example in the following is the Cross Feature Anomaly Detection proposed in [7]. Based on the first order energy estimate of Eq.(3) and for the typical wireless ad hoc network microcontroller MC9S08GT60 of Freescale Semiconductor, an estimate of the energy needed by a node for monitoring is determined here.

The anomaly detection part of the IDS has been implemented in C for the target platform. The program has then been compiled and disassembled to determine the total cycle counts of 80.40 Million machine cycles which corresponds to an execution time of Δt=10.05sec for *f* = 8MHz. Training and test data sets of 400 tuples each are considered in the simulated IDS. Using the values V_{dd} =3V, $I_0(Vdd,f)$=6.5mA, the total energy spent for running the algorithm once is of E_{TOT}=195.9Millijoules.

To determine the impact of the IDS on the battery life of a wireless node, we used the "Battery Life Estimation Model" [9] of a ZigBee Wireless ad-hoc network node using the same microcontroller (MC9S08GT60) and Freescale Semiconductor's MC13192 RF Transceiver. The objective is to compute the life time of an 800mAh battery, when a node transmits 3000 packets and receives 3000 packets a day. Battery supply efficiency is taken as 80% so that the system capacity is accepted as 640mAh. The run currents and the sleep currents of the microcontroller and the RF Transceiver are used in the computation.

The Intrusion Detection algorithm load is introduced into the node model using our previous figure of 10.05s. execution time. When running the intrusion detection algorithm every 30 minutes in an one hour time slot a consumption of 1.28 mAh/Day is computed versus the 0.41 mAh/Day consumption when no IDS is considered. Consequently, a node consumes roughly three times more energy when running an IDS algorithm. We use this ratio in our simulations for our proposed distributed IDS monitoring.

In this model, every time a node performing IDS is sending a report, it broadcasts a message to its cluster. Since the energy to broadcast one single packet is fairly small compared to the IDS algorithm energy expenditure, we do not account for it in our energy model.

As a matter of fact, the more frequently intrusion detection is performed, the more load is expected. The frequency of IDS needs to be increased when traffic increases in the network. Hence, the energy consumption arising from the regular network activity increases as well as the energy needed for the increased monitoring effort.

3. DISTRIBUTED MONITORING: NETWORK MODEL

Suppose that we need to secure a dense ad hoc (sensor) network. We assume that a secure, fast and reliable node to node communication is available. Each node in the network has limited battery power. Clusters are formed according to the clustering approach that is defined in [1]. The network can be partitioned into K non-overlaping clusters. Each node is aware of its fellow cluster members. We consider a cluster k of N_k nodes of which M_k of them are trusted nodes. These trusted nodes are labeled i = 1,2,…, M_k. All M_k nodes are equipped with a perfect IDS. When a node performs intrusion detection, this is effective for the whole cluster and no other node is involved in the monitoring process. The following application proposes a game model for secured military networks where a strong emphasis is put on extending the network's lifetime and consequently accounting for the energy level of a node.

3.1 GAME MODEL

Let W_k denote the overall security value of the cluster k, and let w_i denote the security value of each node i, where $0<w_i<W_k$. The value w_i can be interpreted in the game as the reward for node i to be protected. For each node i,the cost for monitoring is defined by:

$$c_i = E_M / E_{T_i} \quad \textbf{(4)}$$

where E_M is the energy consumed by the IDS and E_{Ti} is the remaining energy level of node i.

Each player i (i.e. node) has two pure strategies: *Contribute* (s_i=1) and *Don't Contribute* (s_i=0) to the distributed monitoring task. We refer to the set of oponents of the player i with the notation s_{-i}.

The payoff function u_i is defined as follows:

$$u_i(s_i, s_{-i}) = \max(s_i, s_{-i})w_i - c_i s_i \quad \textbf{(5)}$$

All the M_k nodes of the cluster are separated into two sets: the set of *low energy* nodes and the set of *high energy* nodes. The first set considers the nodes with a high remaining energy level and second one the nodes with a low remaining energy level with respect to a pre-defined energy threshold E_{th}. Consequently, each node i has a type denoted by $\theta_i \in \{0,1\}$. For θ_i = 0 (resp. θ_i = 1), node i is a low (resp. high) energy node. In the following, the mixed strategy equilibrium of the game is obtained for the case where the high energy nodes (resp. the low energy nodes) have the same probability of contribution p_H (resp. p_L). These contribution probabilities will be defined with the section 3.3, which introduces equilibrium analysis for the game.

The game is repeated at a regular time interval $\Delta t_k=[t_{k+1}- t_k]$ at the beginning of which each trusted node i chooses an action $a_i(t_k)$ which can be "*Contribute"* or "*Don't contribute"*. This action depends on the defending contribution of the other trusted nodes as stated by Eq.(5). Consequently, a player should be able to estimate the contribution probabilities of the other players. In our model, since two types of nodes are defined, a player needs also to know the type of the other nodes to estimate their contribution probabilities. Therefore, each node could simply advertise its own level of remaining energy. Such a solution is not beneficial since it increases the traffic in the network and wastes too much energy. Furthermore, for security purposes, we have to consider this energy-related information as private.

Our proposed game is an incomplete information game where the nodes are uncertain about the energy level of the other nodes. Under the assumption that the energy level depletes slowly in time, a Bayesian game formulation is suitable for modeling such an incomplete information game. Our formulation provides a framework for the node to choose whether to defend the cluster according to its belief about the type (remaining energy level) of all of its opponents.

Under the assumption of slow energy depletion, the type of the users can be assumed to be piecewise constant. In our formulation we assume the type θ_i as being constant at the time scale of the game. We define $\mu_i^j(\theta_j)$ as the belief a node i has about the type of another node j. This belief represents a probability distribution over the possible types of node j.

We assume that players in our game are rational at each stage game. Consequently, each player's optimal strategy is to maximize its own payoff according to his updated beliefs. The update rule for the beliefs is defined in the following.

3.2. BELIEF UPDATE RULE

In the first time t_0, node i's belief about the type of node j is determined by an apriori probability μ_0 (in the following, we consider μ_0=0.5). During the time period Δt_0 , the nodes play the game. The nodes that decide to contribute broadcast their IDS reports to all the trusted nodes. Then, at time t_1, a node i can update its belief about node j based on his observation of the node j's action. Node i's observation about node j's action at previous time period Δt_{k-1} depends on whether node j has sent an intrusion detection report at the end of the time slot in its predefined communication slot. This report indicates that node j was monitoring in Δt_{k-1} and if it detected an anomaly or not.

Node i updates its belief at time t_k about the type of node j by calculating its posterior beliefs, defined as $\mu_i^j(\theta_j \mid a_j(t_{k-1}), t_k)$ where $a_j(t_{k-1})$ represents node j's action at the previous time t_{k-1}. The beliefs of node i can be updated from time period t_{k-1} to t_k using Bayes' rule following:

$$\mu_i^j(\theta_j|a_j(t_{k-1}),t_k) = \frac{p(\theta_j)\cdot P(a_j(t_{k-1})|\theta_j)}{\sum_{\theta'_j} p(\theta'_j)\cdot P(a_j(t_{k-1})|\theta'_j)} \quad \textbf{(6)}$$

where $p(\theta_j)$= $\mu_i^j(\theta_j \mid a_j(t_{k-2}), t_{k-1})$ the value of the belief at time t_{k-1} and $P(a_j(t_{k-1})|\theta_j)$ is the probability that action $a_j(t_{k-1})$ is observed at time period t_{k-1} given the type of the node j.

3.3. EQUILIBRIUM OF THE GAME

Our game model is a Dynamic Bayesian game which follows a multi-stage game model with observed actions and incomplete information. Perfect Bayesian Equilibrium (PBE) extends sub-game perfection to games with incomplete information. In a PBE, every stage game admits a Bayesian equilibrium. Therefore, actions of the players are the best response actions in every stage game given the beliefs of the players at the beginning of each stage game. PBE requires Bayesian updating of beliefs whenever it is applicable. In order to show that our game model has a PBE, Bayesian conditions **B1** to **B4** and the equilibrium condition **P** need to be satisfied.

Bayesian Conditions:

B1: The posterior beliefs are independent, and all types of players have the same beliefs, and even unexpected observations do not make player i believe that his opponents' types are correlated.

B2: Bayes' rule is used to update beliefs from $\mu_i(\theta_j \mid a_j(t_{k-1}), t_k)$ to $\mu_i(\theta_j \mid a_j(t_k), t_{k+1})$ whenever possible.

B3: The players do not signal what they do not know.

B4: All the players have the same posterior belief about the type of another player.

PROPOSITION 1: Our game satisfies the Bayesian condition B1 to B4.

PROOF*:* In our game model, the beliefs of a node i about the type of the other nodes do not depend on its own type θ_i. Hence, the beliefs are independent, all players have the same beliefs and **B1** is satisfied. Condition **B2** is also satisfied based on our proposed belief updating system of Eq.(6). The intrusion detection report is the result of the node's contribution for the cluster in the previous time slot. Therefore, players signal what they already know and **B3** is satisfied. Condition **B4** is also satisfied since the prior beliefs about each node in the first time slots are equal (μ_0=0.5) and the belief updates do not depend on the type of the node i that calculates its beliefs about its set of opponents $-i$ ■

Equilibrium condition P:

For each player i of type θ_i, knowing player i's alternative mixed strategy σ_i' and the global history $h(t_k)$, the expected payoff achieved by employing the mixed strategy σ has to satisfy the following condition:

$$u_i\left[\sigma|h(t_k),\theta_i,\mu_i(.|h(t_k))\right] \geq u_i\left[(\sigma'_i,\sigma_{-i})|h(t_k),\theta_i,\mu_i(.|h(t_k))\right] \quad \textbf{(7)}$$

where $h(t_k)$ is the global history defined by the set of all the actions performed by all the players until time t_k, expressed as:

$$h(t_k) = h_i(t_k), \forall i \quad \textbf{(8)}$$

The action history $h_i(t_k)$ of a player i at the time t_k is a binary vector that contains the actions of the player i at each stage of the game $t_0,\ldots t_{k-1}$ which is given by:

$$h_i(t_k) = (a_i(t_0), \ldots a_i(t_{k-1})) \quad \textbf{(9)}$$

Here, the global history is known by all the players of the game.

Equilibrium condition **P** states that the strategy chosen by player i is optimal for each stage of the game.

PROPOSITION 2: Our game satisfies condition **P**.

PROOF: In our game model, node *i*'s optimal behavior strategy σ_i* with respect to its beliefs about the types of its opponents $\mu_i^j(\theta_{-i} \mid a_{-i}(t_{k-1}), t_k)$ at stage game t_k satisfies the inequality below:

$$u_i((\sigma_i^*, \sigma_{-i})|\theta_i, \mu_i^j(\theta_{-i}|a_{-i}(t_{k-1}), t_k) \geq u_i((\sigma_i', \sigma_{-i})|\theta_i, \mu_i^j(\theta_{-i}|a_{-i}(t_{k-1}), t_k) \quad \textbf{(10)}$$

where σ'$_i$ is an alternative strategy of node *i* and u_i(.) is the expected payoff of user *i* under strategy profile (σ*$_i$, σ-$_i$). Eq.(10) holds because the players are rational and therefore, the equilibrium condition **P** is satisfied ■
Since conditions B1 to B4 and equilibrium condition P are satisfied, our game model admits a PBE.

Mixed Strategy Equilibrium:
Using the indifference principle and the payoff function of Eq. (5), the mixed strategy equilibrium is given by:

$$E_v(Contribute, i) = E_v(Don't\ contribute, i)$$

with $E_v(Contribute, i) = w_i - c_i$ the expected payoff of player *i* for selecting the strategy '*Contribute*' and E_v(*Don't contribute*, *i*) = the expected payoff of player *i* for selecting the strategy '*Don't contribute*' that is given by:

$$E_v(Don't\ contribute, i) = w_i \left(1 - \left[\prod_{j=1, j\neq i}^{M_k} \left(1 - (\mu_i^j(\theta_j = 1|a_j(t_k)) \cdot p_H + \mu_i^j(\theta_j = 0|a_j(t_k)) \cdot p_L)\right)\right]\right) \quad \textbf{(11)}$$

which is the product of w_i and the probability that at least one oponent of node *i* is contributing. In this equation, we have:

$$p_H = P(a_i(t_k) = \{Contribute\}|\theta_i = 1) \quad \textbf{(12)}$$

the probability that a high energy node contributes and

$$p_L = P(a_i(t_k) = \{Contribute\}|\theta_i = 0) \quad \textbf{(13)}$$

the probability that a low energy node contributes.

In order to increase the lifetime of the network, we want to increase the expected number of high energy nodes contributing to save the power of the low energy nodes. The expected number of high energy nodes *contributing* is given by E[N_H]. p_H, where $E[N_H]$ is the expected number of high energy nodes in the cluster. Similarly, we have E[N_L]. p_L the expected number of *contributing* low energy nodes using $E[N_l]$, the expected number of low energy nodes in the cluster.

To control the relative importance of E[N_H]. p_H compared to E[N_L]. p_L, we introduce the constraint:

$$E[N_H].p_H = m.E[N_L].p_L \quad \textbf{(14)}$$

where *m* is defined as the control factor of the model. By increasing *m*, the network lifetime is extended since low energy nodes are less likely to contribute for higher values of *m*.
Following equation (14), p_L is given by:

$$p_L = \frac{E[N_H].p_H}{m.E[N_L]} \quad \textbf{(15)}$$

When substituing p_L in equation (11) using equation (15), we obtain the expected payoff for the strategy '*Don't contribute'* as a function of p_H.
Using the indifference principle, we get the mixed strategy equilibrium p_H* as a function of E[N_H], E[N_L], *m*, w_i, c_i and the beliefs.

Adjusting the security level:
The overall security level is defined by the probability of successfully detecting an intrusion attempt. In our model, this probability of detection $P_{detection}$ is given by the probability that at least one trusted node is contributing in the cluster. It is given by:

$$P_{detection} = 1 - \left[\prod_{j=1, j\neq i}^{M_k} \left(1 - (\mu_i^j(\theta_j = 1|a_j(t_k)) \cdot p_H^* + \mu_i^j(\theta_j = 0|a_j(t_k)) \cdot p_L^*)\right)\right] \quad \textbf{(16)}$$

Substituing p_L* of equation (15) into equation (16), we obtain an equation that only depends on p_H*. Hence, the beliefs and the equilibrium probability p_H* determines the security level of the cluster. Since p_H* depends on the security values w_i, a desired security level can be achieved by appropriately choosing w_i.
Hence, the w_i values are re-calculated at the beginning of the each time slot having the values of the updated beliefs and the desired security level $P_{detection}$ in hand.

Adapting the energy thresholds E_{th}:
Since the energy of a node depletes with time, the number of high energy node is decreasing with time, too. Once all the nodes become low energy nodes, the nodes are newly distributed among the low and high energy node sets by reducing the energy threshold E_{th}.
In our implementation, each subsequent energy threshold is obtained by dividing the current threshold by 2.

4. SIMULATION RESULTS

We have simulated a cluster with M_k=10 static trusted nodes and compared the results in terms of the probability of detection and the network lifetime in time units. We target a probability of detection $P_{detection}$ of 90%. The lifetime of the network is given by the time where the first node dies.

The initial battery levels of the nodes are Gaussian distributed (with mean 1 and variance 5) and the ratio between the energy consumption of a contributing and a

non contributing node follows the results of section 2.2. Several values for the control factor m=[50, 100, 1000] are investigated. For each value of m, the results are averaged over 500 simulation instances. Results are summarized in Table 4.1. The higher m, the longer the network stays alive. However, we can notice a slight decrease in the probability of detection since the low energy nodes contribute less for higher values of m. Nevertheless, this loss is small (~1.2%) compared to the gain (8%) in terms of lifetime when m varies from 50 to 1000.

Table 4.1 Probability of detection and lifetime as a function of *m*

	***m*=50**	***m*=100**	***m*=1000**
$P_{detection}$	**0.8423**	**0.8381**	**0.8317**
Lifetime (time slots)	**216**	**225.5**	**234**

For one of the simulated network instances obtained for *m=50*, we observed a lifetime of 274 time slots with average of 1.58 nodes contributing to the detection throughout the whole network life. The histogram of the IDS node count versus the time slots is shown in Figure 4.1 presented below.

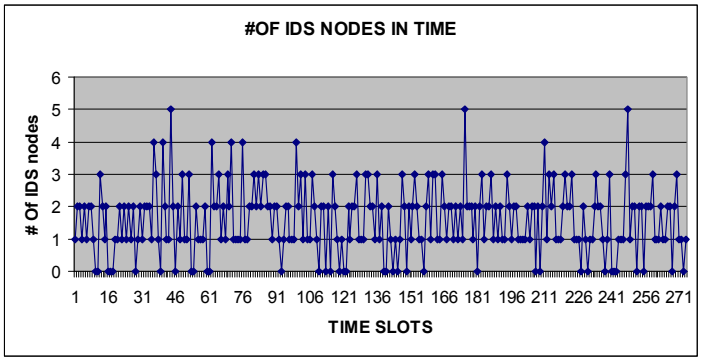

Figure 4.1 Number of contributing nodes vs. time.

In total, there are 38 time slots for which no node is monitoring out of 274 time slots, providing an achieved probability of detection of 86.1%, which is close to the target of 90%. The target is not achieved every time since we introduce beliefs which may not always represent the real status of the network.

The 3-D graph in figure 4.2 compares the energy depletion of each trusted node in a particular simulation run to give idea about how energy consumption varies among the nodes. We can see on this graph that the IDS burden is nicely distributed among the nodes with higher energy.

5. RELATED WORK

Otrok et al, [11] model a distributed mechanism for electing a leader IDS in an IDS election mechanism on MANET, aiming to balance the energy among all the nodes and increase the overall lifetime of an IDS. In the model, incentives are given in the form of reputation to encourage the nodes to cooperate in the leader election process. The reputation is used to track the cooperative behavior of nodes where miss-behaving nodes are punished by withholding the cluster's services. Reputations are calculated based on the Vickrey, Clarke, and Groves truth-telling mechanism. Most other game-theoretic solutions previously proposed for ad hoc networks focus on modeling cooperation and selfishness of the network (e.g. [12, 13, 14, 15, 16]). In these games, each node choose whether to forward or not forward a packet based on the concern about his cost (energy consumption), his benefit (network throughput), and the collaboration offered to the network by the neighbors. Each of these works try to show that by enforcing cooperation mechanisms, a selfish node not abiding the rules will have low throughput in return from the network. For example, in [12], each node uses the normalized acceptance rate (NAR) to evaluate what action he will choose (i.e. forward or not forward) when he receives a packet. NAR is defined as the ratio of the number of successful relay requests generated by a node, to the number of relay requests made by the node. In [17] authors use dynamic Bayesian game to model the interactions between attacker and defender in ad hoc networks. This allows the two players to choose their optimal strategies according to the action history profile and their beliefs about the types of their opponents. In [17] a new Bayesian hybrid detection approach is suggested for the defender, in which a lightweight monitoring system is used to estimate his opponent's actions, and a heavyweight monitoring system acts as a last resort of defense. Authors have shown that the dynamic game produces energy-efficient monitoring strategies for the defender, while improving the overall hybrid detection power.

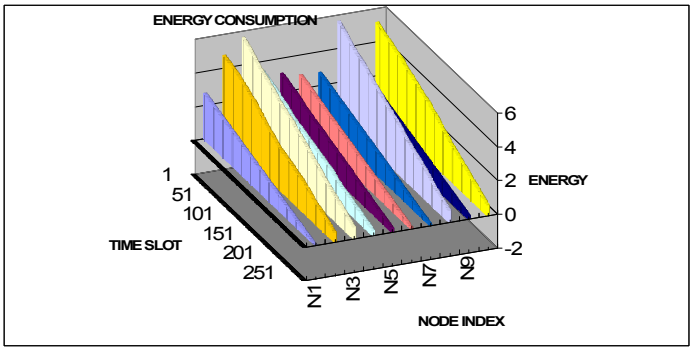

Figure 4.2 Energy depletion of each trusted node in a particular simulation run

6. CONCLUSION

In this work, we propose an efficient mechanism for distributed intrusion detection monitoring in wireless ad hoc networks, when energy and network lifetime are important performance measures.
Our contributions are: 1) we propose a practical energy consumption evaluation model for IDS monitoring based on microprocessor characteristics and software implementation complexity; 2) we propose a mechanism design for efficient distributed monitoring based on a Bayesian game theoretic framework, which maximizes the network lifetime while ensuring probabilistic guarantees for the achieved security level.

ACKNOWLEDGEMENTS

This work is supported in part by ONR grant #N00014-06-1-0063 and by Picatinny ARDEC.

REFERENCES

[1] Y.Huang and W.Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks" In proceeding of the 1st ACM workshop on Security of ad hoc and sensor networks, pp.135-147, October 2003.

[2] H. Deng, Q. Zeng, and D.P. Agrawal, "SVM-based intrusion detection system for wireless ad hoc networks" In Proceedings of the IEEE Vehicular Technology Conference (VTC'03), volume 3, pages 2147-2151, October 2003.

[3] O. Kachirski and R. Guha, "Intrusion detection using mobile agents in wireless ad hoc networks" In Proceedings of the IEEE Workshop on Knowledge Media Networking, pages 153-158, July 2002.

[4] C. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specification-based intrusion detection system for AODV. In Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, pages 125-134, October 2003.

[5] "MC9S08GB60 MC9S08GB32 MC9S08GT60 MC9S08GT32 MC9S08GT16 Data Sheet", Freescale Semiconductor - MC9S08GB60/D Rev. 2.3 /2004

[6] A. Sinha, P.A.Chandrakasan, "JouleTrack - A Web Based Tool for Software Energy Profiling", ACM Design Automation Conference, June 2001.

[7] Y.Liu, Y.Li. and H.Man, "MAC Layer Anomaly Detection in Ad-Hoc Networks", In Proceedings of the 6th IEEE Systems, Man and Cybernetics (SMC), Information Assurance Workshop. pp 402-409, June 2005.

[9] Seminar notes, "ZigBee Technical Training Seminar", Freescale Semiconductor and EBV Electronics- Istanbul Turkey. 22 February 2005.

[10] D. Fudenberg and J. Tirole, "Game theory", The MIT Press Cambridge, Massachusetts. ISBN 0-262-06141-4

[11] H.Otrok, N.Mohammed, L.Wang, M.Debbabi, and P.Bhattacharya, "An Efficient and Truthful Leader IDS Election Mechanism for MANET" In Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)

[12] V. Srinivasan, P. Nuggehalli, C-F. Chiasserini, and R.R. Rao, "An analytical approach to the study of cooperation in wireless ad hoc networks" In IEEE Transactions on Communications, 4(2):722-733, March 2005.

[13] Y. Xiao, X. Shan, and Y. Ren, "Game theory models for IEEE 802.11 DCF in wireless ad hoc networks" in IEEE Radio Communications, 43(3):S22-S26, March 2005.

[14] J. Cai and U. Pooch, "Allocate fair payoff 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.

[15] 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.

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

[17] Y.Liu, C.Comaniciu, and H.Man, "A Bayesian Game Approach for Intrusion Detection in Wireless Ad Hoc Networks" In ACM International Conference Proceeding Series; Vol. 199 , Proceeding from the 2006 Workshop on Game Theory for Communications and Networks ACM 2006.