A Novel Approach for Efficient Usage of Intrusion Detection System in Mobile Ad Hoc Networks

Ningrinla Marchang, *Member, IEEE,* Raja Datta, *Senior Member, IEEE,* and Sajal K. Das, *Fellow, IEEE*

***Abstract*—Mobile Ad hoc Networks (MANET) are self-configuring, infrastructureless, dynamic wireless networks in which the nodes are resource constrained. Intrusion Detection Systems (IDS) are used in MANETs to monitor activities so as to detect any intrusion in the otherwise vulnerable network. In this paper, we present efficient schemes for analyzing and optimizing the time duration for which the intrusion detection systems need to remain active in a mobile ad hoc network. A probabilistic model is proposed that makes use of cooperation between IDSs among neighborhood nodes to reduce their individual active time. Usually, an IDS has to run all the time on every node to oversee the network behavior. This can turn out to be a costly overhead for a battery-powered mobile device in terms of power and computational resources. Hence, in this work our aim is to reduce the duration of active time of the IDSs without compromising on their effectiveness. To validate our proposed approach, we model the interactions between IDSs as a multi-player cooperative game in which the players have partially cooperative and partially conflicting goals. We theoretically analyze this game and support it with simulation results.**

***Index Terms*—Ad hoc networks, intrusion detection, energy efficiency.**

I. Introduction

A *mobile ad hoc network* (MANET) is a self-organized collection of mobile nodes which communicate with each other without the help of any fixed infrastructure or central coordinator. A node can be any mobile device with the ability to communicate with other devices. In a MANET, a node behaves as a host as well as a router. A node intending to communicate with another node that is not within its communication range, takes help of intermediate nodes to relay its message. The topology of the network dynamically changes over time as nodes move about, some new nodes join the network or few other nodes disengage themselves from the network. MANETs have distinct advantages over traditional networks in that they can easily be set up and dismantled, apart from providing flexibility as the nodes are not tethered.

Besides being operable as a stand-alone network, ad hoc networks can also be attached to the Internet or other networks, thereby extending connectivity and coverage more importantly to areas where there are no fixed infrastructures. Present and future MANET applications cover a variety of areas. One important application scenario is vehicular ad hoc network (VANET). VANET is a self-configuring network of moving vehicles (i.e., a vehicle is a node) although the movement pattern of nodes are restricted by the road course, traffic regulations, etc. VANET is a promising technology that has tremendous potential to improve vehicle and road safety, traffic efficiency and convenience ([1]-[2]).

Due to the inherent characteristics of a MANET, such as mobility, wireless communication links and lack of any centralized authority, providing security in a MANET is a challenging task. Moreover, security solutions for fixed wired networks are not easily adaptable to mobile wireless networks. One way of providing security to a MANET is intrusion detection, a process of monitoring activities in the system so as to determine whether there has been any violation of security requirements. Intrusion Detection System (IDS) is the mechanism used by the nodes of a network for detection of intrusion and has been classified into two broad categories based on the techniques adopted, viz., (a) Signature-based intrusion detection and (b) Anomaly-based intrusion detection. In signature-based detection, knowledge about the signatures of attacks is incorporated in the detection system. At the occurrence of an attack, the characteristics of the attack is matched with the signatures included in the IDS. If there is a match, then an attack associated to that signature is said to have occurred. In anomaly-based detection, the IDS does not attempt to find a signature match but searches for anomalous events or behavior. For instance, it could look out for anomalous behavior such as dropping of data packets and events such as erratic changes in the routing table. IDSs can also be categorized based on the audit data used for analysis. Host-based IDSs make use of data obtained from the host for which it checks for intrusion detection. This kind of data could be operating system or application logs on the system. On the other hand, network-based IDSs collect and analyze data from network traffic. In our work, we concentrate on network-based anomaly detection.

While a lot of research effort has been expended in designing effective IDSs, not much effort has been made on efficient employment of the IDSs. In a resource-constrained environment, this is of utmost importance. We attempt to address this issue in our work. In most of the existing IDSs for MANETs, a detection system sits on every node, which runs all the time. One common mechanism used by such IDSs is monitoring traffic in the node's neighborhood ([3]-[8]). Since

Ningrinla Marchang is with the Department of Computer Science and Engineering, North Eastern Regional Institute of Science and Technology, Nirjuli, Itanagar - 791109, Arunachal Pradesh, INDIA e-mail: ningrinla@yahoo.co.in.

Raja Datta is with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology Kharagpur, Kharagpur - 721302, West Bengal, INDIA e-mail: rajadatta@ece.iitkgp.ernet.in.

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, 500 W. 15th Street, Rolla, MO 65409-0350 USA e-mail: sajaldasuta@gmail.com

Manuscript received XXX, XX, 2015; revised XXX, XX, 2015.

a node in a MANET may have limited battery power and computational resource, running an IDS all the time may turn out to be a costly overhead. Thus, the challenge is how to reduce the duration of time an IDS needs to remain active without compromising on its effectiveness. This issue may not be much of a concern in a wired network, in which an IDS is deployed mainly in a stationary router or gateway, with virtually unlimited computational and battery power. But this is of significant concern in the case of MANETs, where the mobile nodes themselves not only behave as hosts and routers, but also have to carry out other functions such as intrusion detection either collaboratively or individually. To this end, we propose a distributed scheme for efficient usage of IDSs in a network based on probability theory.

Cooperative game theory can be used to model situations in which players coordinate their strategies and share the payoffs between them. The output of the game (individual payoffs that players receive) must be in equilibrium so that no player has incentive to break away from the coalition ([33]-[35]). The game settings in all the earlier game-theoretic work on IDS involves two sets of opposing players, the nodes/IDSs and the attacker/defaulters. In our work, we have set a game that involves players (IDSs sitting in neighboring nodes) cooperating to achieve a common goal (i.e., to monitor a single node). To the best of our knowledge, we have not come across any work on cooperating IDSs (to get a security versus energy tradeoff) that models such a situation using game theory. We have presented such a cooperative multi-player game to model the interactions between the IDSs in a neighborhood and used it to validate our proposed probabilistic scheme.

The contributions of this paper are summarized as follows:

1. We present a novel technique, based on a probabilistic model, to optimize the active time duration of intrusion detection systems (IDSs) in a MANET. The scheme reduces the IDSs' active time as much as possible without compromising on its effectiveness.
2. To validate our proposed approach, we also present a multi-player cooperative game that analyzes the effects of individual intrusion detection systems with reduced activity on the network.
3. Through simulation we show that a considerable saving in energy and computational cost is achieved using our proposed technique of optimizing the active time of the IDSs while maintaining the performance of the IDS.
4. The proposed scheme uses local information, thus making it distributed and scalable. Moreover, it works on both static and mobile networks.

The rest of the paper is organized as follows. Section II reviews the work in the existing literature. We define a problem for optimizing the active time of the intrusion detection systems in a MANET in section III. In Section IV we give a multi-player cooperative game theoretic analysis to the problem. A distributed algorithm for efficient usage of IDS is presented in section V. In section VI, we present the performance evaluation and section VII concludes the paper along with directions on future research.

II. Related Work

This section presents existing related work on energy efficient usage of intrusion detection systems in a MANET. In [9], the authors provided a formal study on optimizing network topology for edge-self monitoring in sensor networks with the objective of maximizing the lifetime of the network. The focus is on optimized selection of monitor nodes that monitor communication links so as to reduce the number of monitor nodes. Though the objective is the same, i.e., energy conservation, our work focuses on reducing the active time of the monitor nodes instead of reducing the number of monitor nodes. The existing work focus on reducing the number of monitor nodes that monitor a communication link. Hence, the active nodes bear the whole burden of monitoring communication links while the sleeping nodes sleep. While the overall energy consumption may be reduced, some nodes' energy may be depleted sooner than that of the others. In our work, instead of placing the burden of monitoring on a few selected nodes, every neighbor node chips in so that each node fairly shares the profit (energy saving) as will be illustrated in the simulation results in section VI.

The protocol SLAM [10] makes use of special nodes called *guard* nodes for local monitoring in sensor networks. Usually the guard nodes remain in sleep mode in the network. Before communicating on a link, a node awakens the guard nodes responsible for local monitoring on its next hop. The main aim of the protocol is to reduce the time a guard node remains awake for the purpose of monitoring malicious activities. We find that there is an interdependence between the nodes while carrying out network monitoring. However, in our proposed work, a node determines the probability with which its own IDS monitors and schedules its monitoring time independent of the other nodes. Moreover, when a large number of communication links are in use, almost all the guard nodes in SLAM might be awake, which is also a downside of the protocol.

In [11], a protocol for optimal selection and activation of intrusion detection agents for wireless sensor networks is presented. Only nodes which have the trust value above the trust requirement can activate the intrusion agent to monitor packets and send alert packet to cluster heads. It is a requirement in the protocol for each sensor node to maintain a small trust database of its neighbors and the clustering of sensor nodes. A game theoretic framework for distributed intrusion detection in ad hoc networks which maximizes the network lifetime while ensuring probabilistic guarantees for the achieved security level is presented in [12]. The authors assume that the network in divided into clusters of nodes among which some are trusted. A trusted node is equipped with a perfect IDS so that when it performs intrusion detection, it is effective for the whole cluster and no other node is involved in the monitoring process. In comparison, in our proposed approach we neither assume that some nodes are trusted nor that an IDS is perfect. The existence of the energy-security trade-off that is shown in [12] is also observed in our simulation results. More importantly, all the above work ([9]-[12]) assume the network to be static while our approach works even when the nodes are mobile. In [13], a technique

is presented which optimally selects a subset of nodes in a dynamic network, each of which manages/monitors a subset of nodes with the aim of reducing monitoring traffic or choosing nodes predicted to be long-lived. Optimal selection of m out of M sniffers and assignment of each sniffer to one of the K channels to maximize the total amount of information gathered in a multi-channel wireless network is done in [14]. However, our work does not share the same goal as the above two.

Reduction of energy consumption by intrusion detection systems is being researched in the context of wired networks too ([15]-[16]). In [15], an architecture (LEoNIDS) is presented for network-level intrusion detection system which resolves the energy-latency trade-off by providing both low power consumption and low detection latency at the same time. Packet-based selective encryption is used in [16] for reducing the energy consumption during intrusion detection for networked control systems security.

Game theory is widely used for modeling intrusion detection in wireless networks ([17]-[24]). Several other game-theoretic solutions are also found in the literature that take care of issues like cooperation and selfishness of the nodes in a network ([25]-[29]).

III. Efficient Usage of IDS As an Optimization Problem

We attempt to solve the problem of efficient usage of IDS in two phases: First, we look at the problem from the point of view of a node being monitored by its one-hop neighbors. We present an optimization problem for the same and analyze it using game theory. Second, we view the problem from the point of view of a node which monitors its neighbors. Using the solution to the optimization problem, we arrive at an efficient distributed algorithm which every node in the network employs. Let us consider a network of wireless nodes, each having an intrusion detection system (IDS) that is responsible for detecting malicious activities within its neighborhood. We assume that a mobile node is watched for malicious activities by all its neighbors (nodes within its radio range) using these IDSs. Hence, by neighbor, we mean 1-hop neighbor throughout the rest of the paper. Some level of redundancy can be observed here. Suppose, a node a has k neighbors at a particular instant. For instance, in Fig. 1(a) at a particular instant, node 4 has four neighbors, 2, 3, 5 and 6 (i.e., $a = 4$, $k = 4$). Each of the k neighbors monitors the traffic of node a all the time. At any instant of time, all or some of the k neighbors may detect the malicious activity of node a depending upon the detection rate of the IDS components on them. More importantly, the neighbors spend their valuable computational resources and energy while monitoring node a all the time. However, it may not be required to keep the IDS running on each node all the time. We attempt to reduce this redundancy, thereby saving the afore-mentioned resources. The assumptions that we make are summarized as follows:

1. Each node is equipped with an IDS component.
2. The IDS monitors the traffic of its neighbors all the time (which we wish to reduce).

Further, we make no assumptions about the detection rate of the IDS. The detection rate (and false detection rate) of an IDS depends on factors such as the design of the IDS and how the afore-mentioned characteristics affect the effectiveness of the IDS. In our work, we do not focus on designing an IDS but present a scheme for its efficient usage. The number of IDSs actively monitoring a neighborhood may depend upon the level of security that is desired in there. We define the security level as follows: A security level of l means that a node is monitored by at least l of its neighbors at any instant of time. The security level also provides a trade-off between security and energy consumption. The higher the security level, the more is the number of neighbors that monitor a node at a time, which results in higher energy consumption.

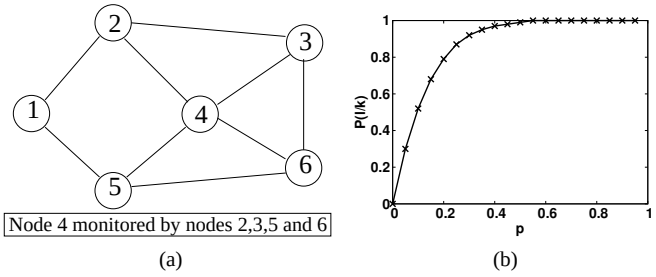
Assume that a node a has k neighbors (IDSs) at a particular instant. Each neighbor monitors independently with a probability of p. The probability that node a is monitored at security level l is:

$$P(l/k) = \sum_{i=l}^{k} \binom{k}{i} p^i (1-p)^{k-i} \tag{1}$$

We define an optimization problem as follows:

$$\textit{Minimize } p \tag{2}$$
$$\textit{subject to } \sum_{i=l}^{k} \binom{k}{i} p^i (1-p)^{k-i} \geq T$$

where $T + \epsilon = 1$ and ϵ is a very small positive number. T denotes a threshold value, which is the minimum probability with which the desired security level (l) is maintained. The value of T can be set depending on the application scenario. Hence, given T and l, an optimal solution to the optimization problem of (2) will give the minimum value of p with which each neighbor has to monitor. Here the monitored node is watched by at least l neighbor IDSs with a probability of T.



(a) (b)

Fig. 1: (a) A MANET (edge between nodes denote they are within radio range); (b) p Vs. P(l/k) using eqn. (1) [k=7 and l=1]

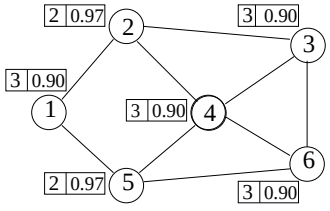


Fig. 2: Calculation of minimum monitoring probability

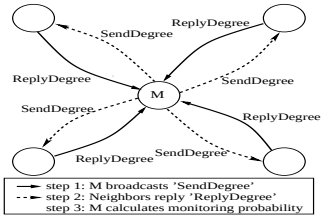


Fig. 3: Illustration of Algorithm LDK

To represent the condition when node a is monitored with probability 1 by at least l neighbor, the optimization problem of (2) has to be optimized towards 1, and the solution would be $p = 1$ irrespective of the value of k. However, we contend that if the requirement that at least l neighbors monitor always (with probability 1) is relaxed by a very small degree (the value of ϵ), we can reduce the value of p to a very large extent. Fig. 1(b) gives the *probability(p)* versus $P(l/k)$ plot using equation (1) when the security level $l = 1$ and the number of neighbors $k = 7$. We can see that the value of $P(l/k)$ increases rapidly as the value of p increases, and stabilizes at about $p = 0.60$. It means that after a certain point, even if we make the IDSs monitor more frequently, the incremental gain is minimal. Moreover, for application scenarios such as an IDS, the value of $P(l/k) = 0.9999$ would effectively mean $P(l/k) = 1$.

IV. A Game Theoretic Analysis of IDS Usage in a Network

The solution to the optimization problem of (2) must be such that it is profitable from the point of view of a cooperating IDS. In other words, the energy saving achieved by this approach should be in equilibrium. To show that, we describe a cooperative game model to represent the interactions between the IDSs in a neighborhood.

Each player's (IDS's) objective is to monitor the nodes in its neighborhood at the desired security level in order to detect any malicious activity. Another objective is to conserve its energy. Here, we would like to consider the first objective as the primary goal and the second one as the secondary goal. If the second objective, i.e., saving battery power, were the main objective, each node would independently decide to sleep all the time resulting in a totally inactive IDS. Since the nodes are independent, they have to cooperate to achieve the above goals. According to [33], cooperative game theory analyzes these situations where the participants' objectives are partially cooperative and partially conflicting. Thus our scenario can be modeled as a n-player cooperative game.

A coalitional (cooperative) game with transferable utility (a TU game) is defined [34] as a pair (N, v) where N is a set of players and v is a function that associates a real number $v(S)$ with each subset S of N. $v(\phi)$ = 0. If a coalition S forms, then it can divide its worth, $v(S)$ in any possible way among its members.

Now, to get a node monitored with the desired security level, each of its neighbors (IDSs) has to participate (cooperate) in monitoring with the minimum probability (solution of problem of (2)). This can be modeled as an n-person cooperative TU game [34] in the characteristic form denoted by $[N, v]$, where $N = \{1, 2, 3, .., n\}$ is a set of players (neighbors) and v is a real-valued characteristic function on 2^N, the set of all subsets of N. Here v assigns a real value $v(S)$ to each subset S of N, and $v(\phi)$ = 0. Assuming that the energy consumption of the IDSs is linear,

$$v(S) = \begin{cases} s(1-p_s)E & \text{if } s \geq l \\ 0 & \text{if } s < l \end{cases}$$

where E = the energy consumed by an IDS if it monitors all the time, $s = |S|$, p_s = the probability with which each player monitors (solution of the optimization problem of (2)) in a coalition consisting of s players, and l = security level. The utility of the game is the energy saved by a player. If $s \geq l$, the desired security level (l) can be achieved and thus the payoff $v(S) = s(1-p_s)E$. Otherwise, the security level cannot be achieved and $v(S) = 0$. Note that the payoff of subset S depends on the cardinality ($|S|$) of the subset and not on the identity of the players in the subset. For instance, $v(1, 2, 3) = v(2, 4, 5) = 3(1-p_3)E$.

A solution to every cooperative game is given by the Shapley value of a player i (refer [31, page 265]):

$$\varphi_i[v] = \sum_S \frac{(s-1)!(n-s)!}{n!}[v(S) - v(S-i)] \tag{3}$$

where n is the number of players, $s = |S|$ and the summation is taken over all subsets S of N. In equation (3), since $[v(S) - v(S-i)] = 0$ if the player $i \notin S$, the summation is effectively taken over all subsets S of which player i is a member. The value of $v(S)$ depends on the cardinality ($s = |S|$) of S. Therefore, we group the subsets depending on their cardinality and the summation is taken over these groups of subsets such that $s = 1$ to n. The number of subsets of size s of which player i is a member is given by $\binom{n-1}{s-1}$. Thus, the Shapley value of player i can be written as

$$\begin{aligned} \varphi_i[v] = & \\ & \sum_{s=1}^{n} \binom{n-1}{s-1} \frac{(s-1)!(n-s)!}{n!}[s(1-p_s) - (s-1)(1-p_{s-1})]E \\ = & \frac{E}{n} \sum_{s=1}^{n} 1 - sp_s + (s-1)p_{s-1} \\ = & (1-p_n)E \end{aligned}$$

where p_n= the probability with which each player monitors in the grand coalition consisting of all the n players.

Observation 1. The Shapley value of the game $\varphi[v] = ((1-p_n)E, (1-p_n)E, .., (1-p_n)E)$ is individually rational since $\varphi_i[v] \geq v(\{i\})$.

Here, $\varphi_i[v] = (1-p_n)E$. According to the characteristic equation, $v(i) = (1-p_1)E$, since the subset S consists of only one player. In the definition of the characteristic function, p_s is the probability with which each member player monitors in a coalition consisting of s players and p_s is obtained using the optimization problem of (2). Thus, $p_n < p_1$. Hence, $(1-p_n)E > (1-p_1)E$. In other words, every player has no problem accepting this payoff since it is better than what it would get when it has to monitor alone. (Refer [33, defn. 4])

Observation 2. The Shapley value of the game $\varphi[v] = ((1-p_n)E, (1-p_n)E, .., (1-p_n)E)$ is an imputation since $\varphi_i[v] \geq v(\{i\})$ and $\sum_{i=1}^{k} \varphi_i[v] = v(N)$.

Here, $\sum_{i=1}^{k} \varphi_i[v] = n(1-p_n)E$ and $v(N) = n(1-p_n)E$. It is an individually rational payoff that allocates the maximum amount. Thus, each player receives the maximum payoff possible. (Refer [33, defn. 5])

Observation 3. The Shapley value of the game $\varphi[v] = ((1-p_n)E, (1-p_n)E, .., (1-p_n)E)$ is collectively rational since $\sum_{i \in S} \varphi_i[v] \geq v(S)$ for all $S \subset N$.

Here, $\sum_{i \in S} \varphi_i[v] = s(1-p_n)E$ and $v(S) = s(1-p_s)E$ where $s = |S|$. And, p_n and p_s are obtained using the optimization problem of (2) and there are more number of players in N than in S (i.e., $n > s$). Thus, $p_n < p_s$. Hence, we observe that $s(1-p_n)E > s(1-p_s)E$. No player has the incentive to quit the grand coalition (i.e., the coalition consisting of all the players) and form a smaller coalition with other nodes.

The above observations show that energy saving (Shapley value) obtained by each IDS with the help of the optimization problem of (2) (which is modeled as the cooperative game) is in equilibrium. Another way of saying that a solution of the game is in equilibrium is to prove that it is in the core of the game. The core of the game is the set of all collectively rational payoffs [33, defn. 6 and 7]. In Observation 3, we have shown that the Shapley value obtained is collectively rational. Therefore, the Shapley value of the game $\varphi[v] = ((1-p_n)E, (1-p_n)E, .., (1-p_n)E)$ is in the core.

V. The IDS Usage Algorithm

Thus far, we have looked at the problem of efficient usage of an IDS from the perspective of a node monitored by its neighbors. Next, we use the optimization problem of (2) as a building block and develop a distributed scheme for the IDSs. Every node employs this scheme to determine the ideal probability with which its IDS has to remain active so that all nodes in the network are monitored with the desired security level.

Let p_i^{min} be the optimal (minimum) probability with which node i has to monitor so that its neighbors are monitored with the desired security level. We refer to p_i^{min} as the minimum monitoring probability of node i. For instance, in Fig. 2, node 5 has three neighbors (1, 4, 6). Suppose, $l = 1$. Here, 4, 1 and 6 have to be monitored by their respective neighbors with a probability of 0.85, 0.97 and 0.90 (solutions of problem (2) when T=0.995) respectively. Since node 5 is a neighbor of the nodes 1, 4 and 6, $p_5^{min} = max(0.85, 0.97, 0.90) = 0.97$.

We define the *degree* of a node to be the number of its neighbors at any instant of time. Let m_i denote the minimum degree of the neighbors of node i. We assign m_i to k in the optimization problem of (2) to obtain the following optimization problem whose solution is p_i^{min}.

$$\textit{Minimize } p \qquad (4)$$
$$\textit{subject to} \sum_{j=l}^{m_i} \binom{m_i}{j} p^j (1-p)^{m_i - j} \geq T$$

where, $T + \epsilon = 1$ and ϵ is a very small positive number. The term T, as explained earlier denotes a threshold value, which is the minimum probability with which the desired security level (l) is maintained, albeit for the whole network.

For instance in Fig. 2, $m_5 = 2$ since 2 is the least of all the degrees of node 5's neighbors, viz., 1, 4, and 6. Consequently, $p_5^{min} = 0.97$. Similarly, the corresponding (m_i, p_i^{min}) pairs for other nodes are also shown in Fig. 2. The minimum monitoring probability obtained as the solution to the optimization problem of (4) ensures that every node in the network is monitored at the desired security level. The proof follows.

Theorem: Each node in the network is monitored with the desired security level (l) when p_i^{min} of each node i is calculated using minimum degree of its neighbors (m_i) in the optimization problem of (4).

Proof: *Assumption*: For every node i, p_i^{min} is calculated using a positive integer x such that $x > m_i$ and yet every node in the network is monitored with security level l.

Let $p^{(m_i)}$ be the solution to the optimization problem of (4). Hence, $p^{(x)}$ denotes the corresponding solution when m_i is replaced by x. Without loss of generality, let node 1 be the neighbor of node i with minimum degree among all its neighbors, i.e., $m_i =$ degree of node 1. Since the L.H.S of the constraint of the optimization problem is the probability that at least l neighbors are monitoring out of all the x neighbors, the value of $p^{(x)}$ decreases as the value of x increases. Hence, $p^{(x)} < p^{(m_i)}$ since $x > m_i$. Here, m_i is the degree of node 1. Hence, $p^{(m_i)}$ is the minimum probability with which node 1 has to be monitored by its neighbors so that security level l is achieved (Refer optimization problem of (2)). Since $p^{(x)} < p^{(m_i)}$, node 1 is not monitored with security level l. This contradicts our assumption. Hence proved.

The mechanism employed by each node in the network to determine the minimum monitoring probability is best presented by the simple algorithm, called *LDK*, which stands for Least Degree for k. The *LDK* algorithm is illustrated pictorially in Fig. 3. Each node (say M) initiates this algorithm to determine the probability with which it has to monitor its neighborhood. In step 1, M broadcasts the message *Send-Degree*. This message is limited to only one hop. In step 2, the neighbors of M reply back with their respective degrees. In step 3, the least of these degrees is assigned to k in the formula, and the minimum monitoring probability of M (p_M^{min}) is calculated.

Algorithm LDK.

Step 1. Each node M broadcasts a message of type Send-Degree to its neighbors asking them to send their $degree$.

$$M -> broadcast : (SendDegree)$$

Step 2. On receipt of the $SendDegree$ message in step 1, each neighbor node, B of M replies to M a $ReplyDegree$ message.

$$B -> M : (ReplyDegree)$$

Step 3. On receipt of each $ReplyDegree$ message in step 2, M does the following:

i. For each message do
$$degree = ReplyDegree;$$

ii. $k = Minimum(degree)$;

iii. If $l > k$ then $p_M^{min} = 1$. Otherwise, p_M^{min} is assigned the minimum value of p (where l is the desired security level of the neighbor, $T + \epsilon$=1, ϵ is a very small positive number) such that

$$\sum_{i=l}^{k} \binom{k}{i} p^i (1-p)^{k-i} \geq T$$

In step 2 of *LDK*, a malicious neighbor may send a false

degree information to M and try to disrupt the algorithm. However, *LDK* is resilient to such an attack under the following assumption. We assume that a malicious neighbor of M would like p_M^{min} to be as less as possible so that the chance of being detected is reduced. It cannot change its security level and thus to be monitored with a low monitoring probability, it can only send a high degree to M in step 2. Since the minimum degree of the neighbors is chosen by M in step 3 to determine the value of p_M^{min}, the high degree sent by the malicious neighbor will most likely not be chosen. Even if several malicious neighbors collude and report an inflated high degree, if there is at least one honest neighbor which reports correctly, the honest neighbor's degree will be chosen as the minimum degree (in step 3) and p_M^{min} will be correctly calculated. We contend it is safe enough to assume that at least one neighbor is honest.

However, the afore-mentioned assumption may not hold for some other kind of malicious neighbor. It may send a low degree (e.g., 1) to force M to use a high monitoring probability and consequently consume more energy. This attack can be handled in two ways. First, in step 2, a neighbor B may send the identity (e.g., IP address) of its neighbors along with its degree. In step 3, M can perform some validation check which we illustrate using an example. Suppose, in Fig. 2, node 5 is malicious. When node 4 requests for the degrees, node 3 replies with (degree:3, IDs:2,4,6), i.e., its degree is 3 and the identities of its neighbors are 2, 4 and 6. Similarly, nodes 2 and 6 reply with (degree:3, IDs:1,3,4) and (degree:3, IDs:3,4,5) respectively. However, suppose node 5 reports a false reply with (degree:1, IDs:4). On receipt of these replies, node 4 now can determine that node 5 has sent a false reply. The degree reported by node 5 has to be at least 2 since node 6 has already reported that it is a neighbor of node 5. Thus, node 4 discards the reply of node 5 and considers only the rest. However, this check will not work in the extreme case when none of the neighbors of node 4 are also neighbors of node 5.

The drawback of the above mechanism is the increase in the size of the reply message. An alternative technique is to request for the IDs of the neighbors only when there is suspicion. The function (Minimum(degree)) in step 3(ii) checks for suspicion before returning the minimum degree. A suspicion is said to have been aroused if there is an outlier (at the lower end) of the received degrees (considering the degrees as a data set), or an unusual value is reported (e.g., 1). Since the neighbors share a neighborhood, it is unlikely that there will be huge differences in their degrees. Thus, an outlier at the lower end could be used to identity a false degree of very low value. In case of a suspicion, the node M can again request for the degrees and the identities and perform the validation check as mentioned above.

A. Message Complexity of LDK algorithm

In step 1 of the *LDK* algorithm, a message is broadcast and in step 2 a message is received from each of its neighbors. Each node executes this algorithm using only local information. Therefore, the worst case message complexity of the algorithm is ***O(d)***, where d is the highest degree of any node in the network at any instant of time. Moreover, provisions in the underlying routing protocol can be exploited. For instance, if the routing protocol is AODV [32], the HELLO packets which are periodically broadcast by each node can be appended with the degree of the node before broadcasting it. Hence, step 1 and step 2 of the algorithm would not be needed. Since the nodes are mobile, the degrees of the nodes may change. So this algorithm must be run at periodic intervals and the value of p_i^{min} recalculated for each node i. This period is a configurable parameter, which has to be set judiciously. It should be noted that when the period is shorter, a more accurate state of the topology will be obtained although a higher communication cost will be incurred as each node has to obtain the degrees of its neighbors through some messages (steps 1 and 2). On the other hand, a longer period may cause the algorithm to use inaccurate information about the topology (depending on how quickly it changes), while reducing the communication cost.

B. Security Level

In *LDK*, the probability with which a node has to monitor depends on the value of the security level. It is defined as the minimum number of neighbors that monitor a node's behavior at any instant. Firstly, the concept of the security level is introduced so that the algorithm *LDK* can be used in a wide range of application scenarios with varying security requirements. Secondly, the level of security provides a tradeoff between intensity of security and energy saving that takes place in the network, (refer Fig. 8(a)). The higher the security level, the higher the number of neighbors monitoring a node at any instant and consequently the lesser the energy saved. Thirdly, the concept provides a mechanism by which we can overcome the inherent challenges posed by distributed intrusion detection. Thus, security level is a critical parameter as illustrated by the following scenario:

In a cooperative IDS, a neighbor of a node cooperate with other neighbor nodes for validation of the observed data. We can see that the proposed probabilistic efficient IDS usage scheme limits monitoring of a node. This is because at any instant of time, all the neighbors of a node are not monitoring its behavior. Here an IDS component may observe only a portion of its neighbor's behavior. This may lead to inconsistency w.r.t. to the observed data in different IDSs. By setting the security level, one can limit the number of nodes observing a node's behavior at any instant of time. Thus, there will be no inconsistency in the observed data of at least these number of nodes. Further, if the validation requires consensus of more number of neighbors, the security level can be raised. The use of the security level for cooperatively detecting a malicious node is demonstrated using simulation experiments in the next section (refer Figs. 4-5).

Additionally, the effect of using *Algorithm LDK* is that a node (IDS component) samples the behavior of a neighbor node instead of monitoring it all the time. It has been found that the sampling rate of an IDS affects its performance [30]. However, in the case of a cooperative IDS, these components

cooperate and share their observations to finally detect any anamolous behaviour. Even if a neighbor may not observe a portion of the node's behavior, some other neighbor(s) observe the said portion. As the number of neighbors that monitor the said portion can be tuned using the security level, no portion goes un-monitored (unobserved). Thus, the performance of the IDS (whose components are on the neighbors) will not be affected.

VI. Performance Evaluation

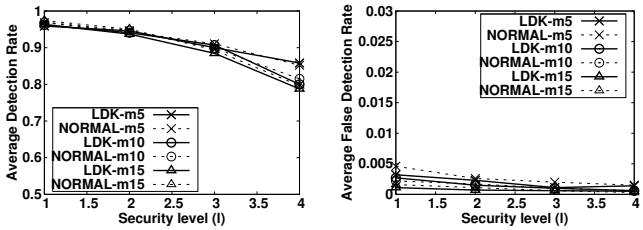
In this section we present simulation results for the *Algorithm LDK* and discuss its performance. We design a cooperative IDS and deploy it in a MANET simulated using ns2.32 simulator [36] and compare its performance under two scenarios:

1. We keep IDSs running on mobile nodes in a network throughout the simulation time.
2. We use the *Algorithm LDK* to reduce the active time of IDS in each node of the network.

The focus is not on the design of the cooperative IDS but on how integrating *LDK* in it helps reduce the active time of individual IDSs while attempting to maintain its effectiveness. The performance metrics are detection rate, false detection rate, and the saving of energy and computational resource. We compare these metrics when *LDK* is in use as compared to when it is not. Additionally, we show the comparison of energy depletion of the individual nodes in the network. We consider a square area of $1000m \times 1000m$ and deploy nodes randomly in this area. Nodes move within this area using the random waypoint movement model [39]. The pause time taken is $0sec$ and each node has a transmission range of $250m$.

For our experimentation, we design an IDS, which detects the dropping of data packets, i.e., *grayhole* attack. A malicious node drops every data packet that comes its way instead of forwarding it. An IDS sits on every node of the network, where the routing protocol used is AODV [32] and the MAC protocol is 802.11. A node obtains the degree of its neighbors with the help of HELLO messages [32]. Each node monitors its neighbors for malicious activities, which we assume here as dropping of data packets. A fixed-size interval, called *IDS-interval* is used by all nodes. Each node divides the simulation time into slots of *IDS-interval* ($2sec$ in our case) independently. There is no synchronization of the nodes. At the start of each interval, each node implements *LDK* and determines the probability with which it has to monitor. Depending on the probability thus obtained, it either monitors during that interval or does not do so. At the end of each interval, a node broadcasts a VOTE message that a neighbor is suspected to be malicious if it drops data packets beyond a predefined threshold. This threshold is configurable. Since the transmission range of a node cannot be changed dynamically in ns2.32, we employ a 1-hop broadcast. Thus, votes about a node are aggregated at that node. We assume a tamper-resistant module which does the aggregation. We also assume the use of a broadcast authentication mechanism for broadcasting the votes. To aggregate the VOTE messages, the simulation time is also divided into slots of *RA-interval* (Result Aggregation interval) which is an integral multiple of IDS-interval. In the simulation, we have taken RA-interval to be equal to IDS-interval. If the number of VOTE messages about a node during an RA-interval reaches a predefined threshold, then the node is said to be detected as malicious during that RA-interval. Otherwise, it is not detected as malicious. In our design of the IDS, we have used the security level (l) as the threshold value. Thus, the detection process of the IDS is strict in the sense that in the scenario when at least l nodes are monitoring, at least l votes are required to convict a malicious node as detected. One may choose a more lenient measure by choosing a value less than (l) for the threshold. False vote messages can be taken care of by increasing the security level and setting (increasing) the threshold so that even if some neighbors collude to send false VOTE messages about a node, it will not be detected as malicious.

The nodes monitor the traffic of their neighbors by operating in the promiscuous mode. A node can set its configuration to 'promiscuous mode' to listen to network traffic within its radio range. For generating each point in the graphs, 3 topology scenes and 5 constant bit rate (over UDP) traffic scenarios of four 64-byte packets/sec were taken, i.e., each point is the average of the results of 15 runs. The simulation time is $500sec$ for each scenario. For deployments of 100, 75 and 50 nodes, the average number of connections established are 70, 50 and 35 respectively. Each connection starts at the beginning of the simulation and continues till the end of it.



(a) Detection Rate (b) False Detection Rate

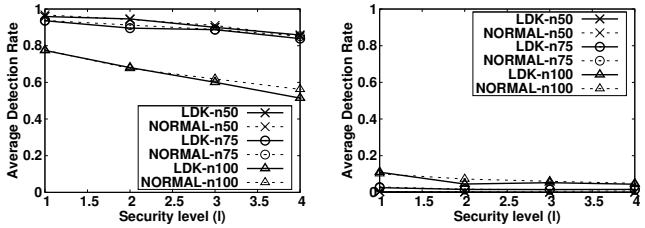
Fig. 4: Effectiveness (varying sec. level & no. of mal. nodes); N=50, P=1m/sec

A. Effectiveness

First, we show the effectiveness of *LDK*. Let N, M, P and l denote the number of nodes, number of malicious nodes, maximum speed of nodes, and security level respectively. Fig. 4(a) shows the average detection rate with respect to l for different values of M $(5, 10, 15)$ when $N = 50$ and $P = 1m/sec$. Detection rate (DR) is calculated as the ratio of the number of times malicious nodes are detected to the total number of times they should have been detected. Similarly, false detection rate (FDR) is calculated as the ratio of the number of times benign nodes are detected to the total number of times malicious nodes should have been detected. The plot labeled *LDK-m5* denotes one when *LDK* is in use and $M = 5$. Similarly, the plot labeled *NORMAL-m5* denotes the plot when *LDK* is not in use (i.e., IDSs are active all the time) and $M = 5$. Other similar labels carry similar meaning. In all the

three cases, we observe that the detection rate decreases as the security level increases. A higher security level represents a stricter IDS in a way that more number of votes are required to finally *convict* a node as malicious. In all three cases, IDS with *Algorithm LDK* gives almost the same detection rates as IDS without *LDK*.

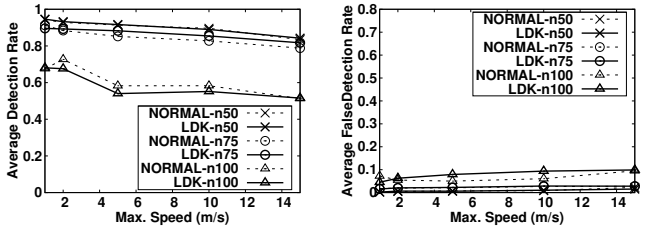
The average false detection rate versus the security level (l) is given in Fig. 4(b). As expected, the average false detection rate decreases as l increases. As explained earlier, at higher security levels, the IDS is said to be stricter and so false detection of a non-malicious node requires more number of votes. Once again, we observe that the IDS with *LDK* gives almost the same false detection rate as IDS without *LDK*.



(a) Detection Rate (b) False Detection Rate

Fig. 5: Effectiveness (varying security & no. of nodes); M=5, P=1m/sec

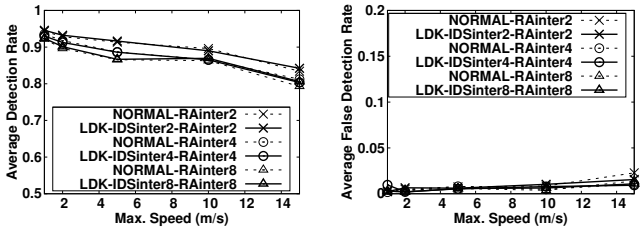
Fig. 5 shows the detection rate and the false detection rate versus l when N varies by 50, 70, and 100 respectively. Here, $M = 5$ and $P = 1m/sec$. *LDK-n50* denotes the plot while *LDK* is in use and $N = 50$. Similarly, the plot labeled *NORMAL-n50* denotes one when *LDK* is not in use and $N = 50$. Other similar labels carry similar meaning. We see the highest detection rate when N is the least (i.e. 50). The higher the number of nodes, more packet collisions occur, which naturally affects the effectiveness of the IDS. However, we find that the IDS with *LDK* gives a detection rate, which is close to that of IDS without *LDK*. We also see that the false detection rate increases as N increases (refer Fig. 5(b)). When more packets are dropped due to collision, false detection increases as the IDS is not able to distinguish between malicious packet dropping and dropping due to collision. So it is quite likely that innocent nodes are *convicted*.



(a) Detection Rate (b) False Detection Rate

Fig. 6: Effectiveness (varying speed & no. of nodes); M=5, l=2

We next observe how the detection rate varies as P varies when $M = 5$ and $l = 2$. Fig. 6 illustrates that IDS with *LDK* gives almost the same detection rate and false detection rate as IDS without *LDK* when $N = 50$. We further find that *LDK* gives better performance when $N = 75$. This is because when *LDK* is not is use, more number of votes are broadcast, and thus more collision of votes occur. Overall, the detection rate tends to decrease as P increases. For $N = 100$ nodes, the plots in Fig. 6 are not smooth. This is because of high packet drop due to the high network density and considerable amount of traffic in the network. Thus, we conclude that the effectiveness of *LDK* is not affected by speed variations.



(a) Detection Rate (b) False Detection Rate

Fig. 7: Effectiveness (varying speed & IDS intvl.); N=50, M=5, l=2

Next, we observe the effect of the length of the of IDS-interval on the detection rate and the false detection rate. Fig. 7 shows the results for varying speed when $l = 2$, $M = 5$ and $N = 50$. The label *NORMAL-RAinter2* denotes the plot when *LDK* is not in use and the RA-interval is $2sec$. The results from the neighbors are aggregated once every RA-interval. The label *LDK-IDSinter2-RAinter2* denotes the plot when every node implements *LDK* at the start of every $2sec$ (IDS-interval) to obtain the probability with which it has to monitor during that interval and the RA-interval=$2sec$. Similar labels carry similar meaning.

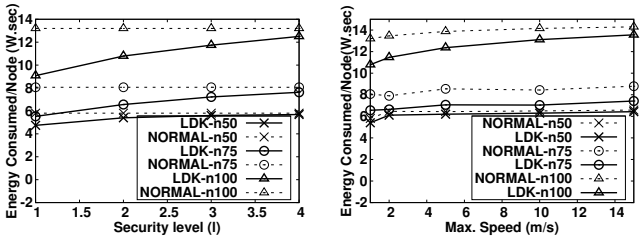
From Fig. 7(a) we see that even when the IDS-interval is increased, the detection rate of IDSs with *LDK* is almost the same as IDSs without *LDK*. We also observe from Fig. 7(b) that the false detection rate remains almost the same for both cases even when the length of the IDS-interval varies. Therefore, we conclude that the length of the IDS-interval has no adverse effect on the effectiveness of the *LDK*. Moreover, the detection and false detection rate improve in both cases (NORMAL and LDK) as the IDS-interval decreases. This is because in a mobile network, a lesser IDS-interval value helps better in adapting to the changing topology. We have set the same value for both the RA-interval and the IDS-interval. This is not mandatory. For instance, one could set the RA-interval to be twice that of IDS-interval. However, in that case, the vote count after each RA-interval must be greater than or equal to twice that of the security level for detection. To summarize, from the above results and discussion we observe that effectiveness of IDS in the network is not compromised while using the proposed *LDK* algorithm in the system.

B. Energy Consumption

In this subsection we show how energy consumption is minimized when *Algorithm LDK* is used in a mobile network. The energy model given in ns2.32 has no provision for

calculating the energy spent in receiving a packet in the promiscuous mode. Hence, we use the energy consumption model given in ([37], [38]) and show the amount of energy consumed in each node during a simulation time of $500sec$. The given consumption model is for the Lucent IEEE 802.11 WaveLAN PC card(2.4 Ghz direct sequence spread spectrum). We compare the energy spent in the two cases viz., when *LDK* is in use and when it is not. We calculate the energy spent for receiving packets in the promiscuous mode only. The two cases differ only in the amount of time spent in monitoring neighbors, which is implemented by sniffing packets in the promiscuous mode. Thus, energy spent by the nodes in performing other functions are the same for both the cases. However, using the results given in [37] about energy consumption in different MANET routing protocols, we show that there is considerable reduction in energy consumption due to *LDK*. A node in the promiscuous mode listens to the traffic of other nodes. We consider the situations when the node is within radio range of only the sender of the packet and when it is within range of both the sender and the receiver.

The energy consumption due to receipt of packets in the promiscuous mode is shown in Fig. 8(a) for varying l and N. Here, $M = 5$ and $P = 1m/sec$. For all the three cases ($N = 50, 75, 100$), energy consumed by IDS without *LDK* is same even if l increases. This is because the IDS is active all the time irrespective of the security level. The energy saving when *Algorithm LDK* is run in the network is clearly seen to be about $4.2W.sec$ per node when $l = 1$ and $N = 100$. When *LDK* is not in use, the energy consumed is about $13.2W.sec$; whereas it is about $9W.sec$ when it is in use. Thus in the best case, our approach reduces the energy consumption due to promiscuous receiving of packets by almost one third. Hence, it is quite evident that a considerable amount of energy will be saved during the entire lifetime of the network. For all the cases, the energy consumption increases as the security level increases. Therefore, it is obvious that to maintain a higher level of security, comparatively more amount of energy needs to be expended.



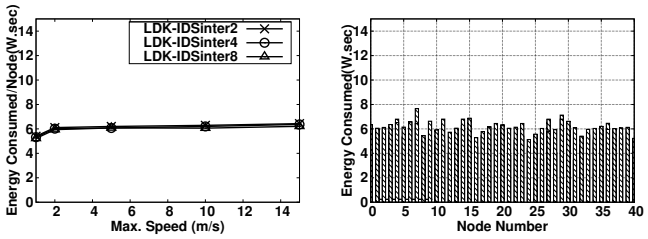
(a) M=5, P=1m/sec (b) M=5, l=2

Fig. 8: Energy consumption per node

Again from [37] it can be seen that the energy consumption due to promiscuous receiving is a substantial part of the total energy consumption of a node. L. M. Feeney [37, figure 4] has given a comparison of the total energy consumption specifying components attributable to traffic sent and received, packets dropped due to collision, and packets discarded or received in the promiscuous mode for routing protocols AODV[32], DSR[39] and DSR-np. In normal AODV, a node does not operate its network interface in the promiscuous mode unlike in DSR. Every packet that is not meant for the node is discarded and the energy spent for this is also shown. Feeney in [37, figure 4] has also shown that the energy spent in discarding packets (that is not meant for it) contributes to more than two-third of the total energy consumed (by all the component attributes) for all possible mobility scenario. While this result may depend on the simulation environment (although the authors have used the same CBR bit rate as we have), it however gives a general idea that discarding of packet contributes substantially to the total energy consumption.

In our simulation, to incorporate the *Algorithm LDK*, we have made AODV to operate in the promiscuous mode. Thus, every node instead of discarding a packet, which is not meant for it, will receive and process it contributing substantially to the total energy consumption as discussed above. Besides, in [37, table 1], it is also shown that receiving a packet in the promiscuous mode is much more costly than discarding it. Therefore we conclude that the energy spent by a node in promiscuous receiving of packets is a substantial part of the total energy spent. Thus reduction of energy spent due to such promiscuous receiving is of much significance.

Fig. 8(b) shows the energy consumed by a node in promiscuously receiving packets as P varies. Here, $M = 5$ and $l = 2$. We see that the energy spent tends to increase as the speed increases for the varying number of nodes. The higher the node mobility, the greater is the number of transmission of packets that are overheard by a node. Here again, we observe that use of *LDK* in the network results in saving of energy.



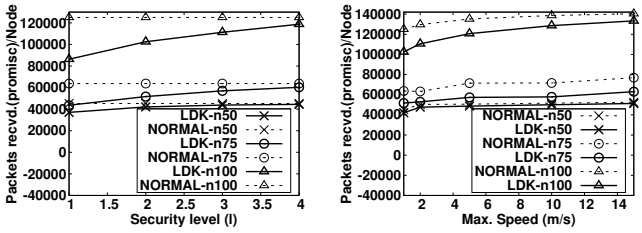
(a) Varying IDS interval (b) For each node

Fig. 9: Energy consumed per node; N=50, M=5, l=2

Next, we evaluate the effect of the IDS-interval on the energy consumption. Fig. 9(a) shows the energy consumption per node for varying speed when $l = 2$, $M = 5$ and $N = 50$. The label *LDK-IDSinter2* denotes the plot when every node implements *LDK* at the start of every 2 seconds (IDS-interval) to obtain the probability with which it has to monitor during that interval. Similar labels carry similar meaning. We observe that even when the IDS-interval varies, the energy consumption per node is almost the same. This shows that the IDS-interval has no effect on the energy consumption of the nodes.

One factor that lends to increasing the lifetime of a network is that there should not be much disparity between the energy depletion rate of the nodes in the network. Instead of placing the burden of monitoring on a few selected nodes, in *LDK* every node involves in monitoring thus ensuring a more

uniform energy depletion rate due to monitoring. Fig. 9(b) shows the energy consumed by each node in promiscuously receiving packets during the simulation. Here, $N = 50$, $M = 5$ and $l = 2$. Each bar on the chart is an average of the results of different values of P (1,2,5,10,15m/sec). The nodes are numbered from 0 to 49. We observe that the there is not much disparity between the energy spent by the different nodes on monitoring during the simulation time. Every node contributes in monitoring, thus spends between $5W.sec$ and $6.7W.sec$. However, some nodes spend a little more energy than the others (e.g., node 0 vs. node 2). This is expected since the network is mobile and the degree of a node keeps changing. Nodes which find themselves in a denser area of the network will monitor with a higher probability than those in less dense areas, and consequently will expend more energy.



(a) M=5, P=1m/sec (b) M=5, l=2

Fig. 10: No. of packets recvd.(promisc) per node

C. Computational Cost

Fig. 10(a) depicts the number of packets received by a node in the promiscuous mode for varying l and N, when $M = 5$ and $P = 1m/sec$. The reduction in the number of packets received when *LDK* is in use is clearly seen in all cases. The number of packets received increases as l increases as expected since the IDS works more to maintain a higher security level.

We therefore conclude here that the computing power spent by the IDS (node) reduces considerably when *Algorithm LDK* is invoked. The computing power spent by the IDS of a node is directly proportional to the number of packets received by it in the promiscuous mode. The reason being that when a packet is thus received, the IDS processes it, which may include buffering of the packet, searching for it in the buffer, incrementing some counter, etc. Assuming c is the cost of computing resources (CPU time, space required, etc.) incurred for processing one packet thus received, the total cost of computational resource required is $125000c$ for $l = 1$ and $N = 100$ for IDS without *LDK*, whereas it is $85000c$ for IDS with *LDK* (refer Fig. 10(a)). The reduction in cost is $40000c$. Hence, the computational cost saved here is almost one-third. For $N = 75$ and $N = 50$, at security $l = 1$, the saving is about one-third and one-fourth respectively (refer Fig. 10(a)).

The number of packets received by a node in the promiscuous mode for varying node speed is given in Fig. 10(b) where $M = 5$ and $l = 2$. We find that the number of packets received in the promiscuous mode tends to increase as the speed increases. The higher the node mobility, the more is the eavesdropping of packets by a node. We observe that the number of packets that are received when *Algorithm LDK* is run in the network is less than when it is not for all speed values. Thus, we see that our approach results in saving of computational power.

In our simulation we have used the HELLO packets for AODV for sending the *SendDegree* and *ReplyDegree* messages. However, they may be generated independently. Moreover, *VOTE* messages are also generated. These messages require extra communication (and hence energy) overhead. However, we contend that since these messages are only generated periodically, the overhead is negligible as compared to the energy spent due to monitoring traffic all the time.

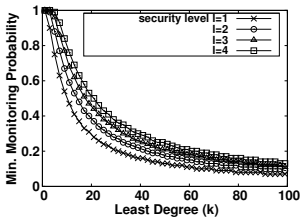


Fig. 11: Effect of least degree of neighbors (k)[T=0.9999]

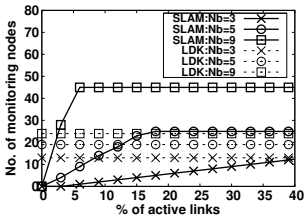


Fig. 12: Comparison of LDK and SLAM

Fig. 11 shows how the minimum monitoring probability (p_M^{min} in step 3[iii] of LDK) varies for increasing values of the least degree (k). We show the plots for varying security levels when $T = 0.999$. It is observed that the higher the security level, the higher is the value of the minimum monitoring probability. It is intuitive that nodes in a network must work harder to achieve higher level of security. Another significant observation is that the value of the monitoring probability decreases rapidly as the value of k increases. The higher the value of k, the denser is the network. As seen from the graph, employing the LDK algorithm will reduce the minimum monitoring probability better in a dense network than in a sparse one. This is expected as the proposed approach exploits the inherent redundancy in monitoring a neighborhood. Hence, if redundancy is minimal, employing LDK will consequently yield minimal benefits. Some of the important simulation results are given in tables I, II and III.

TABLE I: Effectiveness: Detection rate (DR) and False detection rate (FDR) [N=50, max. speed=1m/sec]

M	l	DR(normal)	DR(LDK)	FDR(normal)	FDR(LDK)
5	1	0.968	0.959	0.0046	0.0032
	2	0.943	0.945	0.0026	0.0023
	3	0.910	0.900	0.0020	0.0011
	4	0.852	0.858	0.0015	0.0014
10	1	0.969	0.962	0.0022	0.0027
	2	0.946	0.940	0.0017	0.0015
	3	0.898	0.907	0.0010	0.0010
	4	0.815	0.800	0.0006	0.0006
15	1	0.973	0.963	0.0016	0.0011
	2	0.950	0.937	0.0011	0.0007
	3	0.893	0.884	0.0006	0.0006
	4	0.796	0.787	0.0003	0.0005

Finally, we compare our proposed algorithm (LDK) with an existing algorithm, SLAM [10], which we found to be closest in terms of their goals, although the approaches are entirely different. While SLAM seeks to reduce the energy spent by guard nodes using local monitoring in sensor networks, LDK reduces the active time of the IDSs running in each node of

TABLE II: Effectiveness / Energy saving at varying security level [mal. nodes=5, max. speed=1m/sec, energy used (engy.) in W.sec]

N	l	DR (normal)	DR (LDK)	FDR (normal)	FDR (LDK)	engy. (normal)	engy. (LDK)	pkts rcd. (normal)	pkts rcd. (LDK)
50	1	0.968	0.959	0.0046	0.0032	5.804	4.743	45268	37007
	2	0.943	0.945	0.0026	0.0023	5.804	5.401	45268	42078
	3	0.910	0.900	0.0020	0.0011	5.804	5.603	45268	43779
	4	0.852	0.858	0.0015	0.0014	5.804	5.688	45268	44458
75	1	0.937	0.936	0.0291	0.0257	8.065	5.527	63730	43609
	2	0.912	0.895	0.0173	0.0159	8.065	6.560	63730	51701
	3	0.888	0.887	0.0148	0.0152	8.065	7.221	63730	56998
	4	0.851	0.838	0.0130	0.0159	8.065	7.626	63730	60269
100	1	0.776	0.774	0.1016	0.1099	13.199	9.080	125027	86282
	2	0.675	0.680	0.0725	0.0453	13.199	10.790	125027	102505
	3	0.618	0.600	0.0603	0.0528	13.199	11.738	125027	111298
	4	0.561	0.514	0.0493	0.0444	13.199	12.501	125027	118748

TABLE III: Effectiveness / Energy saving at varying speed [mal. nodes=5, sec. level=2, energy used (engy.) in W.sec, P in m/sec]

N	P	DR (normal)	DR (LDK)	FDR (normal)	FDR (LDK)	engy. (normal)	engy. (LDK)	pkts rcd. (normal)	pkts rcd. (LDK)
50	1	0.943	0.945	0.0026	0.0023	5.804	5.401	45268	42078
	2	0.928	0.932	0.0049	0.0065	6.462	6.108	50376	47647
	5	0.913	0.916	0.0076	0.0061	6.435	6.199	50653	48708
	10	0.896	0.890	0.0084	0.0103	6.500	6.2914	51649	50020
	15	0.834	0.842	0.0228	0.0155	6.581	6.429	52526	51277
75	1	0.912	0.895	0.0173	0.0159	8.065	6.560	63730	51701
	2	0.884	0.892	0.0198	0.0205	7.902	6.638	63231	52931
	5	0.852	0.882	0.0209	0.0225	8.548	7.066	71477	57333
	10	0.827	0.854	0.0267	0.0285	8.437	7.044	71640	57826
	15	0.788	0.817	0.0274	0.0279	8.793	7.417	76891	63004
100	1	0.675	0.680	0.0725	0.0453	13.199	10.790	125027	102505
	2	0.726	0.675	0.0545	0.0627	13.460	11.466	129579	110551
	5	0.582	0.540	0.0501	0.0791	13.872	12.365	135176	120639
	10	0.582	0.551	0.0606	0.0938	14.151	13.114	138645	128585
	15	0.512	0.515	0.0950	0.0987	14.296	13.556	140397	133262

mobile ad hoc networks through a probabilistic scheme. In SLAM, the guard (monitoring) nodes monitor the one-hop traffic links in the network. The term guard node is used for a node that connects two nodes on either side of a link. On the other hand, in LDK the monitoring nodes monitor activities of all its neighbors. Therefore, we can compare the two algorithms only in terms of the expected number of monitoring/guard nodes that are active at any point of time. Assuming a uniform distribution of nodes, let A, d, r and N_b denote the area of the network, node density, transmission range of a node and the number of neighbors of a node respectively. From [40] we know that $N_b = \pi r^2 d$ and the total number of nodes $N = dA = \frac{N_b}{\pi r^2} A$. Further, the average number of guard nodes over a link is $0.51N_b$ [40]. As the total number of links in the network can be given by $\frac{nN_b}{2}$, the total number of guard nodes $g = \frac{0.51nN_b^2}{2}$. Here, g may not denote the total number of distinct guard nodes in the network since a node can be a guard node for more than one link. In [10], a guard node is woken up only when traffic is sent over the link which it is guarding. Thus, we see that SLAM is sensitive to network traffic, whereas LDK has nothing to do with the underlying traffic. We measure the number of IDSs that are active in the whole network at a given time while using LDK as compared to the number of active guard nodes (i.e., g) if SLAM is used. Fig. 12 shows the comparison under varying percentage of active links when $A = 1000m \times 1000m$ and $r = 250m$. The plot with label *SLAM:Nb=3* is for SLAM when $N_b = 3$. Other labels carry similar meaning. We observe that initially when the network is sparse and the traffic is less, the active guard nodes (in case of SLAM) are fewer in number. However, as the network becomes denser and more links are involved in carrying traffic, the number of active guard nodes increases and overtakes the number of active IDSs (in case of LDK) under similar environments. Further, as network traffic is unpredictable when nodes are mobile, it can be concluded that SLAM is not suitable for MANET.

VII. Conclusions

In this paper we have proposed an efficient way of using intrusion detection systems (IDSs) that sits on every node of a mobile ad hoc network (MANET). We first present the minimization of the active duration of the IDSs in the nodes of a MANET as an optimization problem. We then described a cooperative game model to represent the interactions between the IDSs in a neighbourhood of nodes. The game is defined in such a way that the primary goal of the IDSs is to monitor the nodes in its neighbourhood at a desired security level so as to detect any anamolous behaviour, whereas, the secondary goal of the IDSs is to conserve as much energy as possible. To achieve these goals, each of the nodes has to participate cooperatively in monitoring its neighbour nodes with a minimum probability. We then develop a distributed scheme to determine the ideal probability with which each node has to remain active (or switched on) so that all the nodes of the network are monitored with a desired security level. The evaluation of the proposed scheme is done by comparing the performances of the IDSs under two scenarios: (a) keeping IDSs running throughout the simulation time and (b) using our proposed scheme to reduce the IDS's active time at each node in the network. From the simulation results we observe that the effectiveness of the IDSs in the network is not compromised while using the proposed scheme, rather, there is considerable reduction of energy consumption in each of the nodes that increases the network lifetime significantly. Here we have assumed a homogeneous network in a way that all the nodes have the same capacities in terms of their computational and energy resources. In future we wish to extend our model to accommodate a heterogeneous network.

Acknowledgment

The work of N. Marchang and R. Datta has been partially supported by Department of Information technology (DIT), Govt. of India, sponsored project No. 12(44)/05-IRSD. The work of S. K. Das is partially supported by NSF grants under award numbers CNS-1355505, CNS-1404677, CNS-1545050, and CNS-1545037.

References

[1] S. Zeadally , R. Hunt, Y-S. Chen, A. Irwin and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," Telecommunication Systems, vol. 50, no. 4, pp. 217-241, 2012.

[2] S. K. Bhoi and P. M. Khilar, "Vehicular communication: a survey", IET Networks, vol. 3, no. 3, pp. 204 - 217, 2014.

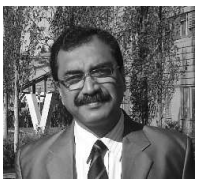
[3] S. Marti, T. J. Giuli, K. La and M. Baker, "Mitigating Routing Misbehavior in a Mobile Ad-hoc Environment," Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 255-265, August 2000.

[4] C. Manikopoulos and L. Ling, "Architecture of the Mobile Ad-hoc Network Security (MANS) System," Proc. IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3122- 3127, October 2003.

[5] K. Nadkarni and A. Mishra, "Intrusion Detection in MANETs - The Second Wall of Defense," Proc. IEEE Industrial Electronics Society Conference '2003, pp. 1235-1239, Roanoke, Virginia, USA, Nov. 2-6, 2003.

[6] A. Partwardan, J. Parker, A. Joshi, M. Iorga and T. Karygiannis, "Secure Routing and Intrusion Detection in Ad-hoc Networks," Proc. 3rd IEEE International Conference on Pervasive Computing and Communications, Hawaii Island, Hawaii, March 8-12, 2005.
[7] N. Marchang and R. Datta, "Lightweight Trust-based Routing Protocol for Mobile Ad Hoc Networks," IET Information Security, vol. 6, no. 4, pp. 77-83, 2012.
[8] N. Marchang and R. Datta, "Collaborative Techniques for Intrusion Detection in Mobile Ad-hoc Networks," Elsevier Ad Hoc Networks, vol. 6, no. 4, pp. 508-523, June 2008.
[9] D. Dong, X. Liao, Y. Liu, C. Shen and X. Wang, "Edge Self-Monitoring for Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems," vol. 22, no. 3, March 2011, pp. 514-527.
[10] I. Khalil, S. Bagchi and N. B. Shroff, "SLAM: Sleep-Wake Aware Local Monitoring in Sensor Networks," Proc. 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007 (DSN 2007), 565-574.
[11] T. Hoang Hai and E-N. Huh, "Optimal Selection and Activation of Intrusion Detection Agents for Wireless Sensor Networks," Proc. Future Generation Communication and Networking (FGCN 2007), vol.1, no., pp.350-355, 6-8 Dec. 2007.
[12] S. M. Fitaci, K. Jaffres-Runser and C. Comaniciu, "On modeling energy-security trade-offs for distributed monitoring in wireless ad hoc networks," Proc. Military Communications Conference, 2008. MILCOM 2008. IEEE , vol., no., pp.1-7, 16-19 Nov. 2008.
[13] R. G. Clegg, S. Clayman, G. Pavlou, L. Mamatas and A. Galis, "On the Selection of Management/Monitoring Nodes in Highly Dynamic Networks," IEEE Transactions on Computers, vol.62, no.6, pp.1207-1220, June 2013.
[14] R. Zheng, T. Le and Z. Han, "Approximate Online Learning Algorithms for Optimal Monitoring in Multi-Channel Wireless Networks," IEEE Transactions on Wireless Communications, vol.13, no.2, pp.1023-1033, February 2014.
[15] N. Tsikoudis, A. Papadogiannakis and E. P. Markatos, "LEoNIDS: a Low-latency and Energy-efficient Network-level Intrusion Detection System," IEEE Transactions on Emerging Topics in Computing, Vol. PP, no. 99, 2014.
[16] R. Muradore and D. Quaglia, "Energy-Efficient Intrusion Detection and Mitigation for Networked Control Systems Security," IEEE Transactions on Industrial Informatics, Vol. 11, no. 3, pp. 830-840, 2015.
[17] S. Shen, "A game-theoretic approach for optimizing intrusion detection strategy in WSNs," Proc. 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), pp.4510-4513, 8-10 Aug. 2011.
[18] A. Afgah and S. K. Das and K. Basu, "A Non-cooperative Game Approach for Intrusion Detection in Sensor Networks," Proc. VTC 2004, Fall 2004.
[19] T. Alpcan and T. Basar, "A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection," Proc. 43rd IEEE Conference on Decision and Control, December 2004.
[20] Y. Liu, H. Man and C. Comaniciu, "A Game Theoretic Approach to Efficient Mixed Strategies for Intrusion Detection," Proc. IEEE International Conference on Communications (ICC 2006), 2006.
[21] Y. Liu, C. Comaniciu and H. Man, "Modeling Misbehavior in Ad Hoc Networks: A Game Theoretic Approach for Intrusion Detection," International Journal of Security and Networks, vol. 1, no. 3-4, 2006.
[22] L. Chen and Jean Leneutre, "A Game Theoretical Framework on Intrusion Detection in Heterogeneous Networks," IEEE Transactions of Information Forensics and Security, vol. 4, no. 2, June 2009.
[23] A. Patcha and J. Park, "A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks," International Journal of Network Security, vol. 2, no. 2, pp. 146-152, March 2006.
[24] N. Zhang, W. Yu, X. Fu and S. K. Das, "Maintaining Defender's Reputation in Anomaly Detection Against Insider Attacks," IEEE Transactions on Systems, Man, and Cybernetics-Part B:Cybernetics, vol 40, no. 3, June 2010, pp. 597-611.
[25] P. Michiardi and R. Molva, "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad Hoc Networks," Proc. WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, March 2003.
[26] A. Afgah, S. K. Das and K. Basu, "A Game Theory based Approach for Security in Wireless Sensor Networks", Proc. International Performance Computing and Communications Conference (IPCCC), April 2004.
[27] S-K. Ng and W. K. G. Seah, "Game-Theoretic Approach for Improving Cooperation in Wireless Multihop Networks," IEEE Transactions on Systems, Man, and Cybernetics-Part B:Cybernetics, vol 40, no. 3, June 2010, pp. 559-574.
[28] M. Féleyházi, J-P. Hubaux and L. Buttyán, "Nash Equilibria of packet Forwarding Strategies in Wireless Ad Hoc Networks," IEEE ransactions on Mobile Computing, vol 5, no. 5, May 2006, pp. 463-476.
[29] F. Li, Y. Yang and J. Wu, "Attack and Flee: Game-Theoretic-Based Analysis on Interactions Among Nodes in MANETs," IEEE Transactions on Systems, Man, and Cybernetics-Part B:Cybernetics, vol 40, no. 3, June 2010, pp. 512-622.
[30] D. Brauckhoff, K. Salamatian and Martin May, "A Signal Processing View on Packet Sampling and Anomaly Detection," Proc. INFOCOM 2010.
[31] G. Owen, *Game Theory*, 3rd Edition, Academic Press, 2001.
[32] C. E. Perkins and E.M. Royer, "Ad-hoc On-Demand Vector Routing," Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, New Orleans, LA, February 1999.
[33] J. Lemaire, "Cooperative Game Theory and its Insurance Applications," Astin Bulletin, Vol 21. No. 1.
[34] B. Peleg and P. Sudholter, "Introduction to the Theory of Cooperative Games," Second Edition, Springer, 2007.
[35] E-Y. Gura and M. B. Maschler, "Insights into Game Theory," Cambridge University Press, 2008.
[36] "The Network Simulator - ns2," www.isi.edu/nsnam/ns/.
[37] L. M. Feeney, "An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks," Mobile Networks and Applications 6, pp. 239-249, 2001, Kluwer Academic Publishers.
[38] "L. M. Feeney, "Investigating the Energy Consumption of an IEEE 802.11 Network Interface," Technical Report, ISRN: SICS-T-99/11-SE, ISSN 1100-3154, Swedish Institute of Computer Science, www.sics.se/ lmfeeney.
[39] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing (T. Imielinski and H. Korth, editors), Kluwer Academic Publishers, 1996, pp. 153-181.
[40] I. Khalil, S. Bagchi and N. B. Shroff, "LiteWorp: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks," Proc. IEEE International Conference on Dependable Systems and Networks (DSN'05), pp. 612-621, 2005.

Ningrinla Marchang received her B.Tech. degree in Computer Science and Engineering from North Eastern Regional Institute of Science and Technology (NERIST), Itanagar, Arunachal Pradesh, India in 1993, M.Tech degree in Computer Science and Engineering from Indian Institute of Technology (I.I.T.), Delhi, India in 1995 and Ph.D. in Computer Science and Engineering from NERIST in 2010. From 1995 to 1996, she worked as a research engineer in the department of Computer Science and Engineering in Indian Institute of Technology, Delhi. From 1996 to 2001, she taught in the Department of Computer Applications in Sathyabama Engineering College, Chennai, India. Since 2001, she has been a faculty member of NERIST where she is an associate professor in the Department of Computer Science and Engineering. She is a member of IEEE. Her research interests include mobile ad hoc networks, sensor networks and cognitive radio networks.



Raja Datta received his B.E. in Electronics and Telecommunications Engineering from National Institute of Technology Silchar in the year 1988. He did his M.Tech. in Computer Engineering and Ph.D. in Computer Science and Engineering, both from Indian Institute of Technology Kharagpur, India. Earlier he was a faculty member in the North Eastern Regional Institute of Science and Technology (NERIST), Arunachal Pradesh, India, where he had been the Head of the Department of Computer Science and Engineering. Presently he is a Professor with the Department of Electronics and Electrical Engineering in Indian Institute of Technology Kharagpur. He is also the Professor In-Charge of Technology Telecom Center of IIT Kharagpur. Prof. Datta is a Senior Member of IEEE and was the General Secretary, Vice Chairman and Chairman of IEEE, Kharagpur Section in 2012, 2013 and 2014 respectively, during which IEEE Kharagpur Section received the best small section award in Region 10. He has a number of publications in National and International Journals and Conferences in the broad area of Computer Communication Networks and has guided a number of doctoral students towards PhD degree. Presently he is guiding 6 research scholars and 4 MS students at IIT Kharagpur. Prof. Datta has carried out and is presently handling several sponsored projects on mobile ad hoc and sensor networks funded by several organizations like Department of Information Technology (DIT), Indian Railways and Indian Space Research Organizations (ISRO), MHRD, Govt. of India. He has also completed consultancy projects with Defense Research and Development Organization (DRDO), Govt. of India and Haldia Dock Complex, WB, India. His main research interest is Computer Communication Networks and Distributed Processing that includes but not limited to Mobile Ad-hoc and Sensor Networks, WDM Optical Networks and Distributed Computing.

Sajal K. Das is the Chair of Computer Science Department and Daniel St. Clair Endowed Chair at the Missouri University of Science and Technology, Rolla. During 2008-2011, he served the US National Science Foundation as a Program Director in the Division of Computer Networks and Systems. Prior to 2013 he was a University Distinguished Scholar Professor of Computer Science and Engineering and founding director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington. His current research interests include theory and practice of wireless and sensor networks, mobile and pervasive computing, big data, cyber-physical systems, smart healthcare, distributed and cloud computing, security and privacy, biological and social networks, applied graph theory and game theory. Dr. Das directed numerous funded projects in these areas totaling over $15M and published extensively with more than 600 research articles in high quality journals and refereed conference proceedings. He holds 5 US patents, coauthored 51 book chapters and four books titled Smart Environments: Technology, Protocols, and Applications (2005), Handbook on Securing Cyber-Physical Critical Infrastructure: Foundations and Challenges (2012), Mobile Agents in Distributed Computing and Networking (2012), and Principles of Cyber-Physical Systems (2016). His h-index is 70 with more than 20,000 citations according to Google Scholar. Dr. Das received 10 Best Paper Awards in prestigious conferences such as ACM MobiCom'99, IEEE PerCom'06 and IEEE SmrtGridComm'12. He is a recipient of numerous awards for research, teaching and mentoring including the IEEE Computer Society's Technical Achievement Award for pioneering contributions to sensor networks and mobile computing, Lockheed Martin Teaching Excellence Award, and Graduate Dean's Award of Excellence for mentoring doctoral students. Dr. Das serves as the founding Editor-in-Chief of the Pervasive and Mobile Computing journal, and as Associate Editor of IEEE Transactions on Mobile Computing, ACM Transactions on Sensor Networks, and several others. Dr. Das is a co-founder of the IEEE PerCom, IEEE WoWMoM, and ICDCN conferences, and served on numerous conference committees as General Chair, Program Chair, or Program Committee member. He is an IEEE Fellow.