Improving the efficiency of Monitoring & detecting Intruders in MANET

V. Pratima,
M. Tech Student, Department of CSE,
Chadalawada Ramanamma Engineering College,
Tirupati.

J. Ravikumar,
Asst. Prof, Department of CSE,
Chadalawada Rmanamma Engineering College,
Tirupati.

***Abstract*— Rely on mobile ad hoc networks so many intrusion detection techniques projected based on each node reflexively monitoring the data forwarding by its next hop. This paper projected valued evaluations of false positives and their collision on monitoring based intrusion detection for MANETs. Our investigational consequence shows that, even for a plain three-node configuration, an actual MANET suffers from high false positives; these results are validated by Markov and probabilistic models. Nevertheless, this false positive dilemma cannot be practical by simulating the similar network via admired MANET simulators, such as ns-2, OPNET or Glomosim. To cure this, Glomosim simulator proposed a probabilistic noise generator model. With the proposed noise model, the simulator exhibits the cumulative false positive actions comparable to that of the investigational tested. In simple monitoring based scheme where no resultant and more exact models are used, false positives causes the collisions in network performance in two ways, condensed throughput in standard networks with no attackers and helplessness to ease the upshot of attacks in networks with attackers.**

***Keywords*: MANETs, improving efficiency, intrusion detection.**

I. Introduction

A mobile ad hoc network (MANET) is a collection of wireless devices moving in seemingly random directions and communicating with one another without the aid of an established infrastructure. To extend the reachability of a node, the other nodes in the network act as routers. Thus, the communication may be via multiple intermediate nodes between source and destination. Since MANETs can be set up easily and inexpensively, they have a wide range of applications, especially in military operations and emergency and disaster relief efforts [9]. However, MANETs are more vulnerable to security attacks than conventional wired and wireless networks due to the open wireless medium used, dynamic topology, distributed and cooperative sharing of channels and other resources, and power and computation constraints. Intrusion detection systems (IDSs), which attempt to detect and mitigate an attack after it is launched, are very important to MANET security. Several monitoring-based intrusion detection techniques (IDTs) have been proposed in literature [7], [2]. In a monitoring-based IDT, some or all nodes monitor transmission activities of other nodes and/or analyze packet contents to detect and mitigate active attackers. Intuitively, it is easy to see that monitoring-based intrusion detection is not likely to be accurate for ad hoc networks due to varying noise levels and varying signal propagation characteristics in different directions.

An IDT uses additional mechanisms such as trust values for nodes before considering nodes to be suspicious. Even with such additional mechanisms, monitoring neighbors' transmissions is the key technique that triggers the detection process for many IDTs. Most evaluations of IDTs are based on small testbed configurations, or simulations which do not incorporate any realistic environmental noise models. More significantly, there are neither report on the extent of the false positive problem nor on the quantification of the effectiveness of monitoring.

In this paper, we quantify false positives and analyze their impact on the accuracy of monitoring-based intrusion detection. We use a combination of experimental, analytical, and simulation analyses for this purpose. First, using a linear chain of three off-the-shelf wireless routers, we show that a sender of data packets falsely suspects, based on the monitoring of transmission activities in its radio range, its next hop of not forwarding its packets (though 98 percent of its packets are delivered to its destination). We validate the experimental results by deriving a Markov chain to model monitoring and estimate the average time it takes for a sender to suspect its next hop. However, this phenomenon cannot be observed using the commonly used simulators such as ns-2, Glomosim or OPNET since they do not implement realistic models of environmental radio noise and thus cannot simulate the false positives that are seen in an actual network.

To remedy this deficiency, we use a previously proposed probabilistic noise model based on the generalized extreme value (GEV) distribution to model the noise levels seen in our experiments [23]. We incorporate the GEV noise model in the Glomosim simulator and show that net impact of false positives seen in the experimental testbed can now be recreated reasonably accurately with simulations. Finally, we use the simulator fortified with the noise model to simulate large MANETs to study the impact of noise on intrusion detection. Our results indicate that monitoring-based intrusion detection has very high false positives, which impact its capability to mitigate the effect of attacks in networks with attackers.

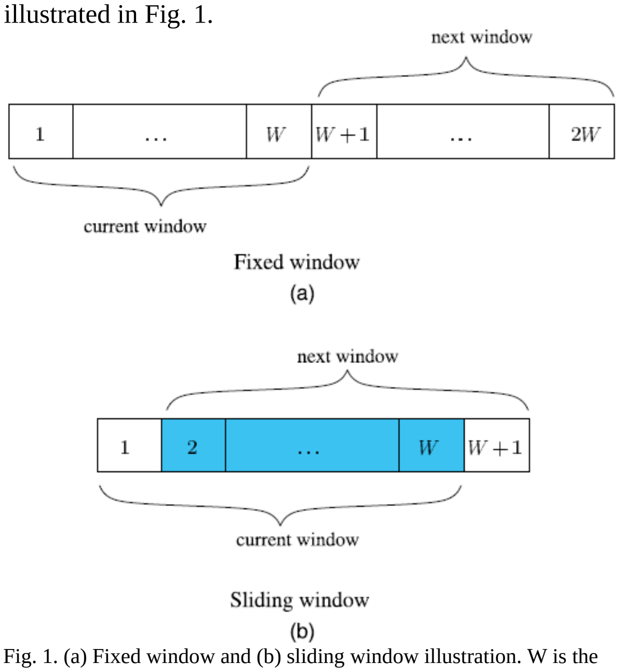
The rest of the paper is organized as follows: Section 2 describes the effect of false positives in monitoring using experiments on a three-node testbed. Section 3 presents analytical models to validate the experimental results. Section

4 presents the measurement and modeling of background noise for wireless devices. Section 5 incorporates proposed GEV noise model and evaluates monitoringbased approaches in large networks. Section 6 presents related work and Section 7 concludes the paper.

II. TESTBED EVALUATION OF FALSE POSITIVES

In monitoring-based intrusion detection, each node monitors the forwarding behavior of its neighboring nodes. In most cases, a node only monitors its next hop in a route. Consider a three-node segment of a route (with at least two hops) being used to send data packets. If the three nodes are denoted as node 1 (source or the node closer to source), node 2, and node 3 (destination or the node closer to destination), then node 2 is the next hop of node 1 and node 3 is the next hop of node 2. When node 1 transmits a data packet to node 2, it expects to hear node 2's transmission of this packet to node 3 within some specified amount of time. If the fraction of packets not overheard by node 1 exceeds a specified threshold, then node 1 concludes that node 2 is dropping too many data packets and suspects it to be a malicious node.

For monitoring purposes, node 1 keeps track of a window of packets that it sent recently to its next hop. Two types of windows can be used to keep track of monitoring: fixed window or sliding window. Let W be the monitoring window size. Also, assume that each packet is given a sequence number, starting at 1. Let j be the sequence number of the most recent packet sent to the next hop. With fixed window monitoring, packets numbered bðj _ 1Þ=WcW þ 1; . . . ; j are monitored. The size of the monitoring window varies from 1 to W. With sliding window, packets j _W þ 1; . . . ; j for j > W or 1; . . . ; j, for j _ W, are monitored. Both types of windows are illustrated in Fig. 1.



Fixed window
(a)

Sliding window
(b)

Fig. 1. (a) Fixed window and (b) sliding window illustration. W is the window size.

Let us consider a detection scenario with a threshold of T; so if L ¼ dWTe packets are not overhead within the current window, then the next hop is suspected. To understand the similarities and differences between the fixed and sliding windows, let us assume that noise does not impact the overhearing of transmissions within a node's radio range. In such a scenario, a malicious node can drop up to L _ 1 packet out of W on the average without risking suspicion by neighbors. However, the temporary drop rates can be different.

Therefore, with the fixed windows approach, a malicious node can afford to drop packets at a faster rate, at times. The drawback of the sliding windows approach is that it can lead to higher false positives in noisy environments.

A. *Testbed Experiments*

To understand the extent of false positives in monitoring, we used a wireless network testbed of three Linksys wrt54g Wi-Fi routers [10]. The wrt54g routers have a built-in fourport 100 Mbps Ethernet switch, an 801.11g access point, two standard omnidirectional antennas, a 200 MHz MIPS processor, and 16 MB of RAM and 4 MB of flash memory, which serves as the disk memory. We reprogrammed the routers using OpenWrt Linux [4]. This testbed was set up as a linear chain in a long corridor in a building with adjacent routers 20' apart. All three routers use the same ssid (which is different from the other Wi-Fi devices in the building-wide 802.11b/g production network) so that they can communicate among themselves only. To minimize the interference, these three routers use a different (noninterfering) channel from those used by other access points. Also, to minimize interference from moving objects and signals from cell phones, we carried out our experiments early mornings from 2:00 am to 5:00 am.

In each experiment, node 1 transmits at a rate of 200 Kbps (fifty 500 byte packets/s) for up to 80 seconds. A single CBR over UDP connection is used. Node 2 transmits every packet it receives from node 1 to node 3. Every node records the ID of each packet it receives, transmits, or overhears. The packet trace from each router is sent to a desktop machine via the Ethernet connection of the routers. After the experiment, we analyzed the packet traces obtained from the three nodes. We removed the traces for the first 500 packets, which were considered to be part of the network warmup. With the MAC level ACK mechanism in the 802.11 protocol, node 1 can determine if a packet it transmitted is received successfully by node 2. Therefore, we considered only the packets that were successfully received by node 2 in our analysis of false positives.

We used these preprocessed packet traces to compute the percentage of packets received by node 2, but not overheard by node 1. If l is the number of packets in the monitoring window that is not overheard, then

$$q_w = \frac{l}{W} \qquad (1)$$

is the fraction of successfully transmitted packets, but not overheard for the current window. (Though the size of the window varies from 1 to W when fixed window monitoring is used, using W rather than the current window size in the denominator results in the correct calculation of qw.) If qw _ T (equivalently, l _ dWTe), where T is the threshold to suspect next hop, then node 1 suspects node 2 of dropping data packets. An IDT uses additional mechanisms such as trust values to actually suspect the nodes. Even in such cases, not overhearing is a key event that triggers the detection process.

III. Analytical Models

We now present an analytical model to validate the experimental results. Let ti; ri, and oi denote, respectively, the number of packets transmitted, received from previous hop, and overheard retransmissions of next hop by node i, for i = 1; 2; 3. It is clear that r1= o2 = t3 = o3 = 0 for the three-node setup we used in the experiments. Also, t1_r2 _ t2 and o1 _ t2. If node 2 is not malicious and no packets are lost due to congestion (which is the case in our experiments), then r2=t2. We calculate the overall not-overheard rate due to environmental noise, denoted q, as follows:

$$q = \frac{r_2 - o_1}{r_2}, \tag{2}$$

$$\begin{aligned} p_{i,i-1} &= \mathcal{P}[\text{The oldest packet in current window is not} \\ &\quad \text{overheard } \cap \text{The newest packet in next} \\ &\quad \text{window is overheard} | \text{current state} = s_i] \\ &= \mathcal{P}[\text{The oldest packet in current window is not} \\ &\quad \text{overheard} | \text{current state} = s_i] \cdot \mathcal{P}[\text{The newest} \\ &\quad \text{packet in next window is overheard} | \text{current} \\ &\quad \text{state} = s_i] \\ &= \frac{i}{W}(1-q), \quad if \quad 0 < i < L, \end{aligned} \tag{3}$$

$$\begin{aligned} p_{i,i+1} &= \mathcal{P}[\text{The oldest packet in current window is} \\ &\quad \text{overheard} \cap \text{The newest packet in next window} \\ &\quad \text{is not overheard} | \text{ current state} = s_i] \\ &= \mathcal{P}[\text{The oldest packet in current window is} \\ &\quad \text{overheard} | \text{ current state} = s_i] \cdot \mathcal{P}[\text{The newest} \\ &\quad \text{packet in next window is not} \\ &\quad \text{overheard} | \text{current state} = s_i] \\ &= \left(1 - \frac{i}{W}\right) q, \quad if \quad 0 \le i < L. \end{aligned} \tag{4}$$

It is noteworthy that node 1 knows r2 due to MAC level ACKs from node 2. The not-overheard rate can also be considered as the probability that a packet received by node 2 was not overheard by node 1. The not-overheard rate q is a key parameter in the development of the analytical model.

A. Sliding Window Model

We model the state of sliding-window-based monitoring using a discrete-time Markov chain. More specifically, we use the number of not-overheard packets in the monitoring window as the state of the monitoring by node 1. The window slides to the right with each packet received by node 2. Therefore, packet receptions of node 2 are the time steps in the Markov chain.

To complete the Markov model, we need to derive the state transition probabilities. Let pi;j denote the transition probability from state si to state sj, i.e., the probability that the number of packets not overheard in next sliding window will be j given the value i in the current sliding window. Only transitions si ! siþ1 for 0 _ i < L; si ! si_1 for 0 < i < L, and si ! si for 0< I< L are feasible since with any new transmission, the number of not-overheard packets can increase by 1, decrease by 1, or remain the same. So pi;j = 0, if ji<j<2. Since sL is an absorbing state, pi,j=0, for j 6=L. Assuming that the not-overheard packets in a sliding window are uniformly distributed, pi;i_1; pi;iþ1, and pi;i are given in (3), (4), and (5).

$$p_{i,i} = \begin{cases} 1 - p_{i,i-1} - p_{i,i+1} & \text{if } 0 < i < L \\ \quad = 1 - \frac{i}{W}(1-q) - \left(1 - \frac{i}{W}\right)q & \\ \quad = 1 - q - \frac{i(1-2q)}{W} & \\ 1 - q & \text{if } i = 0 \\ 1 & \text{if } i = L. \end{cases} \tag{5}$$

The transition probability matrix of the Markov chain is given by

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & 0 & \cdots & \cdots & 0 & 0 \\ p_{1,0} & p_{1,1} & p_{1,2} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & p_{L-1,L-2} & p_{L-1,L-1} & p_{L-1,L} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

It can be partitioned so that

$$P = \left(\begin{array}{c|c} Q & \mathbf{C} \\ \hline \mathbf{0} & 1 \end{array}\right) \tag{6}$$

Where Q is(L-1)*(L-1) sub stochastic matrix describing the probabilities of transition only among the transient states. C is a column vector and 0 is a row vector of (L-1) zeros.
Now, the n-step transition probability matrix Pn has the for

$$P^n = \left(\begin{array}{c|c} Q^n & \mathbf{C'} \\ \hline \mathbf{0} & 1 \end{array}\right) \tag{7}$$

For each experimental data set, we calculated the not overheard rate q using (2) from the end of warmup to the instant of suspecting the next hop and compared the time to suspect value obtained from the analysis of experimental data with the model's estimate for the same q value. Once again, we used window size W =50. These figures show that experimental results match the results from the Markov model, especially for thresholds < 10%.

B. Fixed Window Model

Let X denote the random variable for the number of not overheard packets in a fixed window of size W and q denote the overall not-overheard rate. Then,

$$\mathcal{P}[X = i] = \binom{W}{i} q^i (1-q)^{W-i}$$

In a fixed window, the probability that less than L=WT packets are not overheard is given by

$$\mathcal{P}[X < L] = \sum_{i=0}^{L-1} \binom{W}{i} q^i (1-q)^{W-i}$$

Then, the average number of fixed windows that need to be checked before a fixed window has L or more packets not overheard is

$$N = \frac{1}{1 - \mathcal{P}[X < L]} = \frac{1}{1 - \sum_{i=0}^{L-1} \binom{W}{i} q^i (1-q)^{W-i}}. \tag{8}$$

A truncated negative binomial distribution, is given by

$$R_{fw} = \frac{\sum_{k=L}^{W} k \cdot \left[\binom{k-1}{L-1} q^L (1-q)^{k-L} \right]}{\sum_{k=L}^{W} \left[\binom{k-1}{L-1} q^L (1-q)^{k-L} \right]}. \qquad (9)$$

Once again, we compare the time to suspect values from the experiments with those given by the model for the same q value.

IV. Noise Model for Simulators

In this section, we describe a parameterized noise model that we originally developed in an earlier work to study the impact of noise on the performances of routing protocols [12]. We used an expanded testbed of eight Linksys wrt54g Wi-Fi routers to measure the background noise. We obtained the noise levels using the wl program that came with the driver supplied by the router manufacturer. This noise information is sent to a specified desktop machine via the Ethernet. Due to clock resolution, each router could provide the noise level it sees once in every 100 ms.

A. *GEV Noise Model*

We used MATLAB to analyze and model the noise levels captured in our measurements. MATLAB has an extensive library of distributions including Gaussian, gamma, and lognormal. In such diverse fields as image processing, architectural acoustics, and electronic music, it is often assumed that noise conforms to Gaussian distribution. But, we found that neither Gaussian nor any of the commonly used distributions, such as gamma and lognormal model, the noise levels seen by wireless routers accurately. Further investigation revealed that the GEV distribution [14] models this background noise fairly accurately. If X is a GEV random variable, then its cumulative distribution function (CDF) and probability density function (PDF) are given as follows:

$$F(x;\mu,\sigma,\xi) = \exp\left\{ -\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right) \right]^{-1/\xi} \right\}$$

$$\text{for } 1 + \xi\left(\frac{x-\mu}{\sigma}\right) > 0, \qquad (10)$$

$$f(x;\mu,\sigma,\xi) = \frac{1}{\sigma}\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right) \right]^{-1/\xi-1}$$

$$\exp\left\{ -\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right) \right]^{-1/\xi} \right\}$$

$$\text{for } 1 + \xi\left(\frac{x-\mu}{\sigma}\right) > 0, \qquad (11)$$

We also drew the quantile-quantile (Q-Q) plot of the empirical data and the corresponding GEV random variates. If the theoretical distribution (GEV, in this case) accurately models the empirical data (sampled noise, in our case), then the Q-Q plot would be linear [13]. Fig. 8 shows that the points from GEV distribution fall closer along their reference line than the points from Gaussian distribution. Therefore, GEV distribution models the measured noise data better than the Gaussian distribution. The estimated parameters of GEV distribution for the sampled data are: The sampled data from other routers can also be modeled using GEV distribution with slightly different parameter values. This is to be expected since the environmental noise changes slightly for different labs or offices even on the same floor of a building. See

B. *Three-Node Network Simulations*

We simulated the three-node experimental network using the Glomosim simulator. When the default noise model is used, node 1 never suspects node 2. However, when GEV noise model is used, node 1 tends to suspect node 2. The time to suspect value depends on the q value, which depends on the GEV parameters as seen in the experiments. In GEV, *E* only affects the tail behavior of the distribution slightly, and has more impact on the distribution. So, we adjusted to simulate network environments with different q Values.

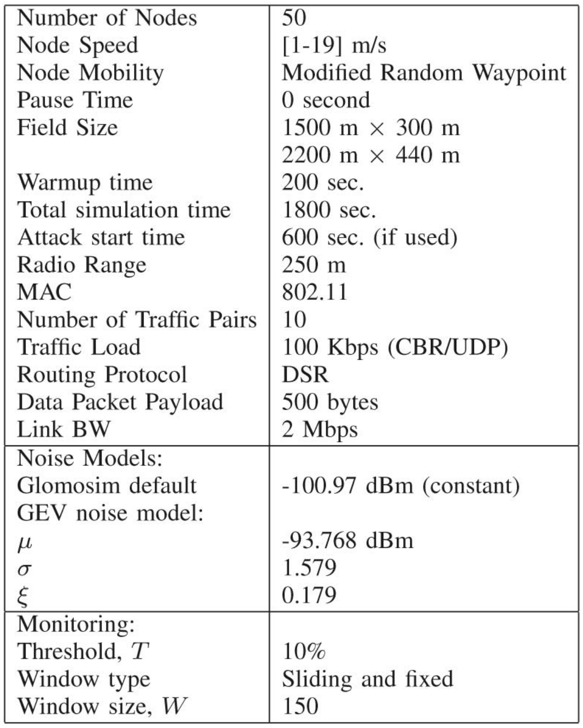
We compared the time to suspect values obtained from simulations with those from the experiments and the analytical model estimates. Owing to space considerations, we show, in Fig. 9, the comparisons for the case of 10 percent threshold only. The RMSE calculations, for the simulated and analytical model values indicate that there is a close agreement between the analytical models and the simulations.

GEV model is similar to the naive sampling indicated by Lee et al. [15]. They propose the closest-fit pattern matching (CPM) approach to generate noise from sampled noise traces to efficiently and accurately simulate packet delivery. Although the CPM technique captures the autocorrelation effects better, we choose GEV model for the following reasons: 1) GEV model is a simple parametric model which can be easily adjusted to simulate different background noise profiles, while CPM requires a new noise trace files in each case; 2) CPM is computationally expensive; 3) GEV model gives reasonably accurate results based on our evaluation of simulation and experimental results.

V. Simulation of Mobile Ad Hoc Networks

We used the Glomosim simulator, v2.03 [11] to evaluate the effectiveness of monitoring in larger mobile ad hoc networks using both GEV noise model. The actual not-overhead rate is higher due to interference from competing transmissions in an ad hoc network. Each node maintains a monitoring window for each traffic flow (connection) though it. In each traffic flow, each data packet sent from the source node is assigned an increasing ID. Only when current node overhears next hop forwarding packets j, it will consider packets with ID between i and j as not-overheard, where i is ID of the last overheard packet and i < j. Therefore, it can avoid false positives due to random back offs at the MAC layer.

We implemented the Watchdog intrusion detection technique (denoted, WD) proposed in [9] as a representative monitoring-based IDT. Following the description given in [12], our implementation has three components: watchdog, pathrater, and sending extra route request when all routes contain one or more suspicious nodes. In the watchdog component, each node that sends or forwards data packets monitors its next hop. When a node suspects its next hop, it will sends an ALARM message to the source node (if it is not the source). When a route break occurs, the monitoring windows in the broken route path are cleared. In the pathrater component, nodes that are not suspected are given a small positive value, less than 1, as their initial rating, which is increased gradually with passage of time. When an alarm message is received by the source node of a route, it will assign a rating of 100 to the suspected node. The rating of a path is the average of the ratings of the nodes on the path. The source chooses the highest rated path if there are multiple positive paths to the same destination. If all paths to its destination have negative ratings, then a new route discovery is initiated (the third component of the IDT) to find a path with positive rating. Although WD is a simple IDT, its primary element monitoring may be used as the key step to initiate the detection process in more elaborate IDSs.



Number of Nodes	50
Node Speed	[1-19] m/s
Node Mobility	Modified Random Waypoint
Pause Time	0 second
Field Size	1500 m × 300 m 2200 m × 440 m
Warmup time	200 sec.
Total simulation time	1800 sec.
Attack start time	600 sec. (if used)
Radio Range	250 m
MAC	802.11
Number of Traffic Pairs	10
Traffic Load	100 Kbps (CBR/UDP)
Routing Protocol	DSR
Data Packet Payload	500 bytes
Link BW	2 Mbps
Noise Models:	
Glomosim default	-100.97 dBm (constant)
GEV noise model:	
μ	-93.768 dBm
σ	1.579
ξ	0.179
Monitoring:	
Threshold, T	10%
Window type	Sliding and fixed
Window size, W	150

Fig 2. Simulation parameters

We used both sliding and fixed window monitors in our simulations since the type of window used in [8] is not specified. However, both produced nearly identical results (often, the curves for both cases are superimposed on each other when plotted). Therefore, we present the results for the sliding window case only.

The simulation parameters are listed in Fig. 11. With 50 nodes, the node densities (the average number of nodes in a radio transmission area) are about 10 for the larger fields and 22 for the smaller fields. We chose long corridor type fields so that routes are likely to have multiple hops. (If one-hop paths are used most of the time or if the network is disconnected most of the time, then the impact of the attacks and the effectiveness of Watchdog IDT cannot be seen clearly.)

A. False Positives in Normal Mobile Ad Hoc Networks

First, we ran a set of simulations to see the extent of false positives in MANETs. We used only monitoring of next hops; there were no malicious nodes in these simulations. When GEV noise model is used, nodes are suspected much faster and more false positives occur. If the simulation is run for long enough time, all nodes in the network will be suspected. Even when default constant background noise is used, there are many false positives due to interference noise from competing transmissions. It is interesting to note that false positives are higher in low-density networks than in high-density networks though the interference noise is likely to be less in the former networks. The reason is, in low-density networks, the hop distances are larger and signals overheard during monitoring are weaker correspondingly.

Also, since there are more hops in each route in the low-density network, there are more chances that nodes will be suspected. Although fewer false positives occur when the threshold is higher (e.g., 15 percent), malicious nodes can take advantage of it and drop more packets without being detected. Therefore, in the remaining of this paper, we choose 10 percent as the detection threshold.

B. Impact of Intrusion Detection Technique on Normal Networks

There are too many false positives when monitoring is used in normal mobile ad hoc networks, especially when the background noise is simulated using the GEV noise model. However, it is not clear if the false positives have any impact on the network performance: since there may be multiple paths between a source and its destination, when a node is suspected, an alternate path that does not involve the node may be used without any loss of performance. Therefore, in this set of simulations, we used the overall network throughput as the performance metric. We measured the network throughput with and without GEV noise model. Then, we turned on the Watchdog IDT (explained above), reran the same configurations and measured the network throughput. We measured the number of delivered data bytes every 100 seconds after the warmup time. The network throughput at any time is given by dividing the total bytes delivered up to that point since warmup by the time elapsed since the warmup.

C. Effectiveness of Intrusion Detection Technique in Networks with Attackers

Next, we evaluated the effectiveness of WD when there are attackers, who participate in the route discovery process as normal nodes but drop all received data packets. Wesimulated the case where 10 of the network nodes are malicious and drop all data packets. This is the attack model used in [12]. It is noteworthy that it is much harder to detect malicious nodes when they drop selectively. Therefore, the simulated attack presents an easier challenge for an IDT.

Fig 3 shows the network throughput when the 10 malicious nodes drop all received data packets starting at simulation time of 600 seconds. Without WD, the network throughput degrades to 40 percent in the high-density network, and to 46 percent in the low-density network. In the high-density network, with WD active, the network throughput is improved from 53.3 to 90.1 kbps with default background noise, and from 54.7 to 65.8 kbps with GEV background noise. In the low-density network, however, WD does not mitigate the impact of the attacks, especially when GEV noise model is used.

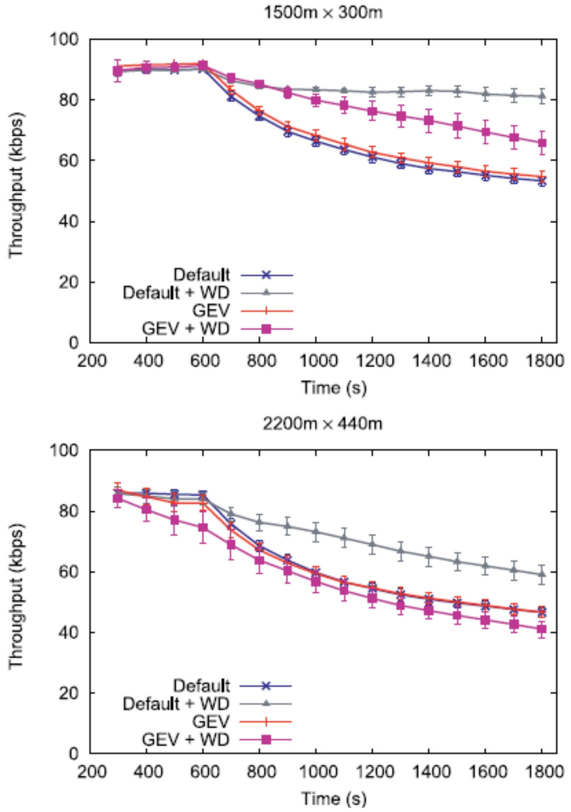


Fig 3. Throughput in networks with attackers. Ten nodes are malicious nodes which drop all received data packets.

VI. RELATED WORK

Many IDTs for MANETs have been proposed in literature. They can be classified as: signature-based detection, anomaly detection, and specification-based detection. A survey of intrusion detection techniques is given in [15]. Based on how the data needed for intrusion analysis are gathered, IDTs for MANETs can be divided into three approaches: monitoring-based, probing-based, or explicit feedback among intermediate nodes in routes. (Explicit feedback among end nodes is commonly used for both security and performance tuning. We do not specifically review the literature on this technique.)

Watchdog and pathrater [13] are the first monitoring based technique proposed for ad hoc networks. In this approach, nodes monitor transmission activities of neighboring nodes and analyze packet contents to detect and mitigate an attack after it is started. When a node suspects its next hop, it will send an alarm message back to source node. Pathrater is used to punish suspicious nodes by not including them in routing. However, monitoring-based intrusion detection is not likely to be accurate for ad hoc networks due to varying noise levels, varying signal propagation characteristics in different directions, and interference due to competing transmissions within the network. In this paper, we showed monitoring gives very high false positives when environmental noise effects are considered. We tried to complement the existing results by quantifying the benefits and overheads of watchdog in more realistic noise conditions.

The Watchdog technique has been extensively studied for its deficiency, false positives and has been modified or supplemented it with other mechanisms to make it more accurate. Specific results include CONFIDANT [7], [5], CORE [11], and LARS [12]. These results use different policies to propagate monitored information (trust) to others in order to mitigate misbehavior and enforce cooperation. In particular, Buchegger and Le Boudec [5] present a Bayesian approach to assign trust and reputation ratings the CONFIDANT system. Their simulation results (with the default noise model of a constant value) show that incorporating secondary trust information gathered from other nodes with the primary trust information directly gathered (by monitoring) can significantly speed up the detection of misbehaved nodes. The effectiveness of these approaches needs to be carefully evaluated with more realistic noise simulation models or experiments.

There are several other papers on using a reputation/ trust system for MANETs [11], [10], [12]. Luo et al. [15] describe a localized trust model in which multiple nodes are collaboratively used to provide authentication services. Eschenauer et al. [11] describe a trust framework which encompasses Pretty Good Privacy (PGP) [12] like trust models. Liu et al. [11] present a dynamic trust model to address packet drops by selfish and malicious nodes. In general, a trust system requires propagation and dissemination of trust. Also trust evidence must be distributed redundantly to handle the unreliable connectivity in MANETs [11]. Trust propagation is complex, not well understood in the context of ad hoc networks, in which trust collection and dissemination may be incomplete and problematic and has high computational requirements (e.g., collaborative authentication [12]) and communication overhead (requiring localized or limited distance network floods).

In probing-based approaches [1], [12], [13], [14], nodes query other nodes and receive their reception and transmission of data. Analyzing this information, they can detect intruders. However, probing-based approaches have different issues. First, it will incur more delay to detect malicious nodes since an anomalous activity needs to be suspected/identified prior to probing for relevant data from other nodes. Second, malicious nodes can give false probe data to avoid detection. Third, malicious nodes can also collude to avoid detection, or frame up legitimate nodes, or deceive legitimate nodes to send incorrect information. The explicit feedback approach by Liu et al. [15] requires downstream nodes (toward the destination) send explicit ACKs to upstream nodes two hops away from them. This achieves the intended effect of monitoring with explicit ACK packets between nodes two hops away from each other. This method overcomes the limitations of monitoring at the cost of additional packet transmissions, book keeping, and computational overhead. We have used a similar approach, denoted p-hop crosscheck, p _ 2, to detect control packet falsification in on-demand route discoveries in another paper [14]. In addition to the computational and book-keeping overhead, this approach works only for isolated and noncolluding malicious nodes. For example, if there are two malicious nodes in a row in a route that always send the anticipated feedback ACKs upstream and ignore any ACKs from downstream nodes, then the 2ACK scheme for data packets as well as the two-hop crosscheck for control packets do not work [14].

This paper is based on our earlier conference paper [3]. Compared to the conference version, this paper presents the fixed window model, additional analysis for different threshold values, more details on RMSE analysis, description of the GEV noise model, and significantly more simulation results including the simulation analyses of a second network with higher node density. The GEV noise model and the experiments to capture the noise samples are originally described in another paper [15], which investigates the impact of noise on the performances of routing protocols. This paper uses the noise model to investigate passive monitoring and its effectiveness.

VII. CONCLUSIONS

Rely on mobile ad hoc networks so many intrusion detection techniques projected based on each node reflexively monitoring the data forwarding by its next hop. This paper projected valued evaluations of false positives and their collision on monitoring based intrusion detection for MANETs. Our investigational consequence shows that, even for a plain three-node configuration, an actual MANET suffers from high false positives; these results are validated by Markov and probabilistic models. Nevertheless, this false positive dilemma cannot be practical by simulating the similar network via admired MANET simulators, such as ns-2, OPNET or Glomosim. To cure this, Glomosim simulator proposed a probabilistic noise generator model. With the proposed noise model, the simulator exhibits the cumulative false positive actions comparable to that of the investigational tested. In simple monitoring based scheme where no resultant and more exact models are used, false positives causes the collisions in network performance in two ways, condensed throughput in standard networks with no attackers and helplessness to ease the upshot of attacks in networks with attackers.

REFERENCES

[1] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, “An On-Demand Secure Routing Protocol Resilient to Byzantine Failures,”Proc. ACM WiSe, pp. 21-30, Sept. 2002.

[2] S. Bansal and M. Baker, “Observation-Based Cooperation Enforcement in Ad Hoc Networks,” Research Report cs. NI/0307012, Standford Univ., 2003.

[3] R.V. Boppana and X. Su, “An Analysis of Monitoring Based Intrusion Detection for Ad Hoc Networks,” Proc. IEEE Globecom:Computer and Comm. Network Security Symp., Dec. 2008.

[4] R.V. Boppana and S. Desilva, “Evaluation of a Stastical Technique to Mitigate Malicious Control Packets in Ad Hoc Networks,” Proc. Int’l Symp. World of Wireless Mobile and Multimedia Networks (WoWMoM)/Workshop Advanced Experimental Activities on Wireless Networks and Systems, pp. 559-563, 2006. BOPPANA AND SU: ON THE EFFECTIVENESS OF MONITORING FOR INTRUSION DETECTION IN MOBILE AD HOC NETWORKS 1173

[5] S. Buchegger and J.Y. Le Boudec, “A Robust Reputation System for Mobile Ad-Hoc Networks,” Proc. Workshop Economics of Peerto-Peer Systems (P2PE ’04), 2004.

[6] S. Buchegger, C. Tissieres, and J.Y. Le Boudec, “A Test-Bed for Misbehavior Detection in Mobile Ad-Hoc Networks – How MuchCan Watchdogs Really Do?” Proc. IEEE Workshop Mobile Computing Systems and Applications (WMCSA ’04), 2004.

[7] S. Buchegger and J.-Y. Le Boudec, “Performance Analysis of the Confidant Protocol: Cooperation of Nodes Fairness in Dynamic Ad-Hoc Networks,” Proc. IEEE/ACM MobiHoc, 2002.

[8] R. Burchfield, E. Nourbakhsh, J. Dix, K. Sahu, S. Venkatesan, and R. Prakash, “RF in the Jungle: Effect of Environment Assumptions on Wireless Experiment Repeatability,” Proc. IEEE Int’l Conf. Comm. (ICC ’09), pp. 1-6, 2009.

[9] I. Chlamtac, M. Conti, and J.J.-N. Liu, “Mobile Ad Hoc Networking: Imperatives and Challenges,” Ad Hoc Networks, vol. 1, no. 1, pp. 13-64, 2003.

[10] Cisco Systems Inc., Linksys WRT54G v2.2 Wireless-G Broadband Router, http://www.linksys.com, 2004.

[11] [11] L. Eschenauer, V.D. Gligor, and J. Baras, “On Trust Establishment in Mobile Ad-Hoc Networks,” Proc. Security Protocols, pp. 47-66,2003.

[12] J. Hu, “Cooperation in Mobile Ad Hoc Networks,” Technical Report TR-050111, Dept. of Computer Science, Florida State Univ.,2005.

[13] R. Jain, The Art of Computer Systems Performance Analysis:Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons, 1991.

[14] M.R. Leadbetter, G. Lindgreen, and H. Rootze, Extremes and Related Properties of Random Sequences and Processes. Springer-Verlag, 1983.

[15] H. Lee, A. Cerpa, and P. Levis, “Improving Wireless Simulation through Noise Modeling,” Proc. ACM Int’l Conf. Information Processing in Sensor Networks (IPSN ’07), pp. 21-30, Apr. 2007.