An Intrusion Detection System for Suburban Ad-hoc Networks

Muhammad Mahmudul Islam, Ronald Pose and Carlo Kopp
School of Computer Science and Software Engineering, Monash University, Australia
Email: {sislam,rdp,carlo}@csse.monash.edu.au

***Abstract*— Due to the nature of the wireless media, ad-hoc wireless networks are vulnerable to various attacks. There are security protocols that prevent unauthorized nodes from accessing the network through authentication. Secrecy of information is provided through encryption. However these protocols cannot detect if any member of the network degrades the network performance due to misbehavior. Therefore an intrusion detection system (IDS) is required that monitors what is going on in the network, detects misbehavior or anomalies based on the monitored information and notifies other nodes in the network to take necessary steps such as to avoid or punish the misbehaving nodes. In this paper we propose an IDS, referred to as the SAHN-IDS, suitable for multi-hop ad-hoc wireless networks like a SAHN (Suburban Ad-hoc Network). SAHN-IDS detects misbehavior based on nodes getting an unfair share of the transmission channel. It also detects anomalies in packet forwarding, such as intermediate nodes dropping or delaying packets. Unlike most IDSs for detecting anomalies in packet forwarding, SAHN-IDS does rely on overhearing packet transmissions of neighboring nodes, since that is ineffective in networks where nodes use different transmission power, different frequency channels and directional antennas for different neighbors. Moreover, unlike most IDSs, most of the thresholds in SAHN-IDS are set dynamically. We show the effectiveness of SAHN-IDS through simulations.**

I. Introduction

Security in wireless ad-hoc networks is more difficult to achieve than a wired counterpart due to the limited physical protection of each node, unreliability of wireless links and the lack of any central infrastructure. Security protocols, such as LLSP [1] and Ariadne [2], try to secure the network from unauthorized users (i.e. intruders). However attackers may be robust enough to succeed in infiltrating the security system and compromising the members (i.e. authorized nodes) of the network possibly causing them to misbehave. A member node may also misbehave due to selfishness. Node misbehavior can result in degradation of network performance. Hence it is important to monitor the system to look for any anomalies and take necessary actions if an anomaly is detected. A system performing these tasks is known as an intrusion detection system (IDS).

An IDS aims to enhance the intrusion prevention facility of the underlying security protocol. An ideal IDS should able to detect an anomaly quickly so that the misbehaving node/nodes can be identified and appropriate actions (e.g. punish or avoid misbehaving nodes) can be taken so that further damage to the network is minimized. It should be able to set thresholds for its detection schemes dynamically so that misbehaving nodes cannot easily work around the detection scheme. For detecting anomalies in packet forwarding it should not rely on overhearing packet transmissions of neighboring nodes since limitations on transmission range may make this impossible. This lead to the following problematic situations:

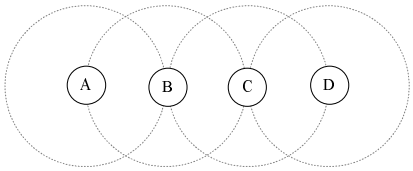


Fig. 1. Nodes in a network. Each dotted circle represents the transmission range of the corresponding node.

- **Ambiguous collisions:** A node will not be able to decode the contents of a packet by overhearing if the packet collides with other packets transmitted from other nodes. Hence a detection scheme based on overhearing may not be able to identify the nodes.
- **Receiver collisions:** In receiver collisions a node A (in Figure 1) can tell if B has forwarded a packet to C but not if C has received it or not. Thus if B wants to circumvent the detection mechanism of A, it can purposefully cause a collision at C by forwarding the packet to C when C is transmitting.
- **Limited transmission power:** A node B (in Figure 1) can limit its transmission power such that the transmitted signal is strong enough to reach the previous node A but not the actual recipient C.
- **Directional antennas:** A directional antenna prevents neighboring nodes, that are not within its direction, from overhearing its transmissions.

Moreover, nodes may employ a link layer security module, such as LLSP [1], which encrypts the packets for each neighbors with different keys. In this case A, in Figure 1, will not be able to decode the overheard packets transmitted by B for C and hence its detection mechanism will fail.

We propose an IDS, referred to as SAHN-IDS, that achieves the desirable properties of a good IDS with respect to a suburban ad-hoc network (SAHN) [1][3][4][5][6][7]. Simulation results indicate that SAHN-IDS is effective enough for the identified attacks.

[1]The SAHN is a multi-hop ad-hoc mesh network that has been proposed to alleviate the expensive, oversubscribed, area limited and less secured features of existing wireless broadband solutions. Provision for efficient and dynamic network management protocols at each node makes the network independent of any centralized administrator. The security scheme at each layer is particularly appealing to security conscious business users. Additionally the wireless medium makes the SAHN suited to extending the Internet infrastructure to areas of inadequate wired facilities.

Section II gives a summary of the related works and their shortcomings. Section III gives an overview of the attacks we have considered in our scheme. Sections IV, V, VI and VII describes our detection schemes. Sections VIII and IX outline the possible responses to the identified attacks. We present some simulation results in Section X that shows the effectiveness of SAHN-IDS. Finally we conclude our paper with some plans for future research.

II. Related Work

Intrusion detection systems can be classified broadly into two classes: (a) Reputation based schemes and (b) Incentive based approaches. Reputation based schemes, e.g. [8][9] [10], detect misbehaving nodes and notify other nodes of the misbehaving nodes so that misbehaving nodes can be punished or avoided in future routing. Incentive based approaches, such as [11], aims to promote positive behavior to foster cooperation instead of relying on participants to report and punish misbehaving nodes. These schemes use a virtual money concept to charge for using network resources. Since nodes need to gain virtual money inorder to request other nodes to forward their packets, resource abuse like DoS attacks by flooding becomes very expensive. However making an effective charging system may be very difficult. Since SAHN-IDS is a reputation based system, we will describe work related to this class only.

Zhang et al. [8][12] have developed a distributed and cooperative intrusion detection system (IDS) where individual IDS agents are placed on each and every node. Each IDS agent runs independently, detects intrusion from local traces and initiates response. If the evidence of anomaly from a local trace is inconclusive, neighboring IDS agents cooperatively participate to resolve the issue. The authors have detailed intrusion detection methods for the following attacks: (a) falsifying route entry in a node's route and (b) random packet dropping by intermediate nodes. The random packet dropping detection scheme relies on overhearing transmissions of neighboring nodes which has some limitations described earlier.

Bhargava and Agrawal [13] have extended the IDS model described in [8] to enhance the security in AODV (Ad-hoc On-demand Distance Vector [14]) routing protocol. The proposed scheme is claimed to identify false route request, DoS, compromise of a destination and impersonation attacks. However, the proposed scheme may not detect intentional packet delays by a misbehaving node acting as an intermediate router. It has the limitations similar to [8].

AODVSTAT [15] is a STAT (State Transition Analysis Technique [16]) based IDS designed for detecting attacks against the AODV routing protocol. The attacks that AODVSTAT can detect using state-transition diagrams are (a) Spoofing attacks where packets arrive with the same IP but different MAC addresses or vice versa, (b) Dropping of packets, (c) Resource depletion attack, (d) False propagation of sequence numbers and (e) Man-in-the-middle attack. Like [8], it is not clear how a node can monitor ongoing traffic of its neighbors if those neighbors transmit packets with limited transmission power, different frequency channels or directional antennas.

Marti et al. [10] have proposed an intrusion detection technique, known as Watchdog, to detect nodes that agrees to forward packets but fail to do so. They have used another module, known as Pathrater, that uses the information from Watchdog and helps the routing protocol to avoid misbehaving nodes. Watchdog and Pathrater are best suited to be implemented on top of source routing protocols, such as DSR (Dynamic Source Routing [17]). Similar to the previous protocols Watchdog has the limitations of relying on overhearing packet transmissions of neighboring nodes for detecting anomalies in packet forwarding.

CONFIDANT (Cooperative Of Nodes, Fairness In Dynamic Ad-hoc NeTworks) [18][19] is an extended version of Watchdog and Pathrater [10] where this scheme not only takes into account the observed or reported misbehavior of a malicious node, but also punishes the misbehaving node. Nodes with bad reputations are isolated inorder to limit their activity in the network. Thus CONFIDANT, unlike Watchdog and Pathrater, stimulates misbehaving nodes to contribute to the normal operations of the network in order to be able to get services from other nodes. But CONFIDANT suffers from similar limitations to [8].

CORE (Collaborative Reputation) [20] is a reputation based system similar to CONFIDANT [18]. But unlike CONFIDANT, it does not allow negative ratings to be broadcast by other nodes to prevent false accusation. It is a generic mechanism that can be integrated with several network and application layer functions. Examples of the network function include performing route discovery, forwarding data packets etc. Similar to [8], the limitations of the detection system in networks with limited transmission power and directional antennas have not been addressed.

Like our protocol, Balakrishnan et al. [21] have proposed a way to detect packet dropping in ad-hoc networks that addresses the problems of receiver collisions, limited transmission power and directional antennas discussed earlier. This scheme (TWOACK) can be added on to a source routing protocol such as DSR. Suppose node A has discovered a route to F with a source route $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$. In TWOACK when B forwards a packet for A, C (the node two hops away from A) receives the packet and sends an acknowledgement to A indicating B has forwarded the packet properly. If A does not get an acknowledgement for the packet, it expected to be forwarded by B to C, within a certain timeout period it suspects B to be misbehaving. The same procedure is carried out by every set of three consecutive nodes along the source route. In TWOACK each forwarded packet has to be acknowledged which may contribute to traffic congestion on the routing path. S-TWOACK (Selective TWOACK) reduces this extra traffic by sending a single acknowledgement for a number of packets instead of for a single packet. Unlike our scheme TWOACK/S-TWOACK cannot detect the misbehavior of a forwarding node if it violates the fairness of the underlying packet transmission scheduling function and hence delays packet transmissions for selected nodes.

SAHN-IDS does not suffer from the aforementioned limitations. Moreover, unlike others, SAHN-IDS can select most of its thresholds dynamically.

III. Attack Models

SAHN-IDS aims to detect the following attacks:

A. Unfair use of the transmission channel (ATTACK1)

A node can prevent other nodes in its neighborhood from getting fair share of the transmission channel. This misbehavior can be considered as DoS (Denial of Service) attacks against the competing neighbors in a contention based network since the competing neighbors are deprived of their fair share of the transmission channel. The possible methods for this type of attack are as follows:

- **Not complying with the MAC protocol:** Contention based MAC (Medium Access Control) protocols, such as 802.11, use RTS and CTS to notify the immediate neighbors of the transmitters and receivers how long the transmission channel will be reserved for the successful transmission of the associated data packets. RTS/CTS and the backoff mechanism aim to minimize collisions among competing neighbors and try to ensure that all the competing neighbors can get some share of the common channel. However a node can generate RTS/CTS at an unacceptable rate by ignoring the backoff mechanism so that competing neighbors cannot get an adequate share of the transmission channel. This can cause the packets waiting at the output queues of the competing neighbors to wait for too long until they time out and get removed. Both RTS and CTS contain fields that notify neighboring nodes for how long the frequency channel will be occupied for successful transmission of the associated data packets. If the indicated duration ($\mathrm{T_i}$) is less than the actual duration ($\mathrm{T_a}$) taken for successful transmissions, the transmission channel will remain occupied for an additional period, $\mathrm{T_a} - \mathrm{T_i}$. The competing neighbors may not be aware of this additional hidden period. As a consequence neighbors trying to access the channel within the hidden period are likely to face unexpected collisions, increase their backoff intervals and hence may not get their share of the channel.
- **Jamming the transmission channel with garbage:** Garbage can consist of packets of unknown formats, MAC layer packets violating the proper sequence of a transaction (e.g. sending a data packet without exchanging RTS and CTS) or simply random bits used as static noise by misbehaving nodes. Garbage data may result in too many collisions, may consume a significant part of the available channel capacity or both. Consequently legitimate neighbors may not be able to access the channel properly when needed.
- **Not complying with the bandwidth reservation scheme:** Nodes in a multi-hop wireless network can reserve bandwidth, i.e. a portion of the transmission channel capacity, along its route before initiating a flow. If there is not enough bandwidth, new flows should not be admitted so that existing flows are not choked. A misbehaving node may not abide by this rule and try to push out packets when there is not enough bandwidth left. As a result legitimate nodes may not get fair share of the transmission channel.

B. Anomalies in Packet Forwarding (ATTACK2)

Anomalies in packet forwarding can take the following forms:

- **Drop packets:** A node may disrupt the normal operation of a network by dropping packets [22]. This type of attack can be classified into two types: (a) Black hole attack and (b) Gray hole attack. In a black hole attack a misbehaving node drops all types of packets (both data and control packets). In a gray hole attack an attacker selectively drops packets (only data packets). In this paper we consider only the gray hole attacks. We refer to this attack as **ATTACK2a**.
- **Delay packet transmissions:** A node can give preference to transmitting its own or friends' packets by delaying others' packets. As a result some flows may be not be able to meet their end-to-end delay and jitter requirements. **ATTACK2b** and **ATTACK2c** refer to the attacks related to delay and jitter requirements respectively.

If these anomalies are not detected, nodes may still use the offending node/nodes in their routes to connect to the remote parts of the network and may not achieve required QoS.

IV. Detecting ATTACK1

We make the following assumption to detect ATTACK1. Traffic flows allocate bandwidth (i.e. link capacity) at each routing node before they can begin their actual transmissions. If there is not enough bandwidth, additional flows are not admitted. This enables existing flows to achieve their desired QoS. A detailed description of such an admission control protocol can be found in [5][6][7].

Let us assume that for each period T, a node X knows that $p\%$ of the available link capacity has been allocated by its neighboring nodes where $p \leq L$ where L is the total link capacity. L should be less than 100% since no system can work at 100% capacity. We have set it to 90%.

Now for each period T, X measures the percentage of link capacity $r\%$ being used by the neighboring nodes for the admitted flows. It also measures the percentage of link capacity $s\%$ being wasted due to collisions, garbage data and flows that did not reserve bandwidth. If

$$(r + s) \geq L \tag{1}$$

X assumes that a neighbor or a group of neighbors are accessing the channel unfairly. X increases a non-negative misbehavior counter MC[ATTACK1] each time X detects ATTACK1 and decrements it if there is no such misbehavior. If MC[ATTACK1] reaches a threshold, X declares its neighborhood misbehaving.

Sometimes a neighbor of X may not utilize the whole part of link capacity allocated to an admitted flow. This can happen if the flow does not send packets at a constant bit rate. Hence r can be less than p. Therefore, $r < p$ does not mean that neighbors are not getting fair share of the channel. However, $r < p$ can also be true if a neighbor does not get fair share of the channel due to any of the reasons mentioned in Section III-A.

So far we have discussed the technique to detect if the neighborhood of a node is under ATTACK1. Detecting the node or direction responsible for ATTACK1 has not been discussed. Now we will extend our scheme to identify the offending node or the direction of misbehavior. Note that the following schemes will only work when a single neighbor misbehaves.

To identify the neighbor causing ATTACK1 by sending packets that can be decoded (i.e. not garbage), we can use the following technique. X measures the percentage of link capacity s[N]% being wasted by a neighbor N for the flows originating/passing through it without reserving bandwidth for each period T. Now if

$$s[\mathrm{N}] \geq (L - p) \tag{2}$$

X assumes that N is responsible for ATTACK1. Similarly X maintains a misbehavior counter MC[ATTACK1, N] for N. If MC[ATTACK1, N] reaches a threshold, X declares N to be misbehaving.

To identify the neighbor causing ATTACK1 by jamming the transmission channel with garbage, we need to use a separate directional receiver along with the existing omnidirectional receiver. The directional receiver points to a particular direction for a period T and is then rotated to a different direction. For each direction and period T if

$$s \geq (L - p) \tag{3}$$

X suspects that a neighbor in that direction may be jamming the transmission channel with garbage data. X maintains a misbehavior counter MC[ATTACK1, direction] for each direction. If the misbehavior counter for a particular direction reaches a threshold, X declares that direction to be misbehaving.

V. Detecting ATTACK2a

For a flow f each node, h hops away from the source in the routing path, measures the rate R[f, h] at which it processes packets. At the source, intermediate and destination nodes processing of packets refers to sending, forwarding and receiving packets respectively. h of the source is 0 and it increases for subsequent nodes towards the destination. At the end of each period T, the destination puts R[f, h=destination] in a packet called the route status packet (RSP) and sends it to the source through all the intermediate nodes of f. Each intermediate node also appends R[f, h] to RSP before sending to the next node. R[f, h] can be digitally signed by its respective node to prevent other nodes from modifying it. When RSP reaches the source node, it contains R[f, h] values of all the downstream[2] nodes of f. Now we can estimate the forwarding ratio[3] of a node h hops away from the source by the following expression:

$$\text{Forwarding ratio, } \mathrm{F}[f, h] = \frac{\mathrm{R}[f, h+1]}{\mathrm{R}[f, h-1]} \tag{4}$$

If

$$\text{Delivery ratio, } \frac{\mathrm{R}[f, h = \text{destination}]}{\mathrm{R}[f, 0]} < \mathrm{R}^{\mathrm{thres}}[f] \tag{5}$$

where $\mathrm{R}^{\mathrm{thres}}[f]$ is the allowable minimum end-to-end delivery ratio[4] for the flow f, the source suspects the intermediate node, h hops away from the source with the highest F[f, h], is dropping packets at an intolerable rate. The source maintains a misbehavior counter MC[ATTACK2a, f, h] for each downstream node. If MC[ATTACK2a, f, h] reaches a threshold for a downstream node, the source declares the node to be misbehaving.

Note that the threshold value in Eq.(5) has been selected based on the requirement of the flow, i.e. dynamically. The reason to include Eq.(5) with Eq.(4) is to prevent any false alarm when some intermediate nodes may drop packets but the end-to-end delivery ratio requirement will still be met.

VI. Detecting ATTACK2b

We present two schemes to detect ATTACK2b. The first scheme relies on time synchronization among all nodes. One way to achieve time synchronization in an ad-hoc network is to install GPS (Global Positioning System) on each node. Before sending packets of a flow f the sender puts time-stamps in each packet. Time-stamps can be digitally signed to prevent other nodes from modifying them.

When a node in the route of f receives a packet from the source, it subtracts the time-stamp in the packet from its current time. Thus a downstream node of f computes the delay each packet has encountered since it was sent from the source. Each node averages such delays for a period T. Let the average value be $\mathrm{T}^{\mathrm{avg}}[f, h]$ at a downstream node, h hops away from the source. Similar to the detection scheme for ATTACK2a, the destination periodically sends a RSP containing $\mathrm{T}^{\mathrm{avg}}[f$, h=destination] to the source through all the intermediate nodes of f. Each intermediate node also appends $\mathrm{T}^{\mathrm{avg}}[f, h]$ to the RSP with a digital signature. When RSP reaches the source node, it contains $\mathrm{T}^{\mathrm{avg}}[f, h]$ values of all the downstream nodes of f. With this information the source can estimate the average delay $\mathrm{T}_{\mathrm{d}}^{\mathrm{avg}}[f, h]$ the flow has encountered at an intermediate node h hops away from the source, i.e.

$$\mathrm{T}_{\mathrm{d}}^{\mathrm{avg}}[f, h] = \mathrm{T}^{\mathrm{avg}}[f, h+1] - \mathrm{T}^{\mathrm{avg}}[f, h-1] \tag{6}$$

Now if

$$\mathrm{T}^{\mathrm{avg}}[f, h = \text{destination}] > \mathrm{T}^{\mathrm{thres}}[f] \tag{7}$$

where $\mathrm{T}_{\mathrm{ete}}^{\mathrm{thres}}[f]$ is the allowable maximum end-to-end delay of f, the source assumes the intermediate node, h hops away from the source with the highest $\mathrm{T}_{\mathrm{d}}^{\mathrm{avg}}[f, h]$, is not complying with the end-to-end delay requirement. If the misbehavior counter MC[ATTACK2b, f, h] reaches a threshold, the source declares the node h hops away from the source is misbehaving.

The second scheme is based on measuring round trip delays with probe packets for each intermediate node. This scheme relies on the following assumptions: (a) links are bidirectional, (b) transmission and queuing delays in both directions should be almost the same, and (c) probe packets should be encrypted on end-to-end basis so that intermediate nodes cannot detect their types and hence cannot treat them differently to remain undetected.

After initiating a flow f, the source node sends periodic probe packets to each of the associated downstream nodes. Once the probe packet reaches its destination X with hop count h, it is sent back to the source with the time $\mathrm{T}^{\mathrm{process}}[f, h]$ indicating the processing delay at X. The source also measures the round trip delay of the probe packet sent to X. Let the average round

[2]Nodes towards the destination of a flow are called the downstream nodes.

[3]forwarding ratio = $\frac{\text{Data received by the neighboring downstream node}}{\text{Data sent from the neighboring upstream node}}$

[4]delivery ratio = $\frac{\text{Data received successfully}}{\text{Data Sent}}$

trip delay of probe packets sent to X be $\mathrm{T}^{\mathrm{roundTrip}}[f,h]$. Let the average end-to-end delay up to node X be denoted by $\mathrm{T}^{\mathrm{avg}}[f,h]$. $\mathrm{T}^{\mathrm{avg}}[f,h]$ can be computed as follows:

$$\mathrm{T}^{\mathrm{avg}}[f,h] = \frac{\mathrm{T}^{\mathrm{roundTrip}}[f,h] - \mathrm{T}^{\mathrm{process}}[f,h]}{2} \quad (8)$$

Now we can substitute Eq.(8) into Eq.(6) and Eq.(7) in order to detect ATTACK2b.

VII. Detecting ATTACK2c

Detection of the jitter requirement violation does not require any time synchronization among the network nodes. Each node, on the routing path of a flow f, can measure jitter based on the average inter arrival times of the associated packets. Intermediate nodes of a flow can use the method used in RTCP (Real-Time Control Protocol, RFC1889) for jitter estimation. Let the jitter estimated by a node, h hops away from the source on the routing path, for the flow f and for a period T be J[f, h]. At the end of each period T, a RSP collects J[f, h] values of all the downstream nodes of f and reaches the source. Let the end-to-end jitter tolerance for the flow f be $\mathrm{J}^{\mathrm{thres}}[f]$. If

$$\mathrm{J}[f, h = \text{destination}] > \mathrm{J}^{\mathrm{thres}}[f] \quad (9)$$

the source node suspects the node, h hops away with the highest J[f, h], is responsible for ATTACK2c. If the misbehavior counter MC[ATTACK2c, f, h] reaches a threshold, the source declares the downstream node, h hops away, is misbehaving.

VIII. Responding to ATTACK1

If the detection module of X succeeds in identifying the misbehaving node N or the direction of misbehavior (DoM), it takes the following actions:

- X notifies the routing module that it should avoid N or the neighbors towards the DoM in the route discovery process for a certain duration.
- X notifies the underlying security module to re-authenticate N or all the neighbors towards DoM.
- If X forwards any packet for an ongoing flow to N or towards the DoM, it notifies the source of the flow to find an alternate route that avoids N or the neighbors of X towards DoM.

The detection module of X may detect ATTACK1 but may not identify the misbehaving node or the direction of misbehavior. In this case the responses are as follows:

- X notifies the routing protocol to avoid all its neighbors in the route discovery process a certain duration.
- X notifies the underlying security module to re-authenticate all its neighbors.
- If X forwards any packet for an ongoing flow to any of its neighbors, it notifies the source of the flow to find an alternate route that avoids all the neighbors of X.

IX. Responding to ATTACK2

If a node X detects a downstream node Y is responsible for ATTACK2 for a flow f, it takes the following actions:

- Notifies the routing module to find an alternate route avoiding Y for the flow f.
- Notifies its routing module and those of the neighbors of Y to avoid Y for a certain period for any new flow that has similar QoS requirements as f.

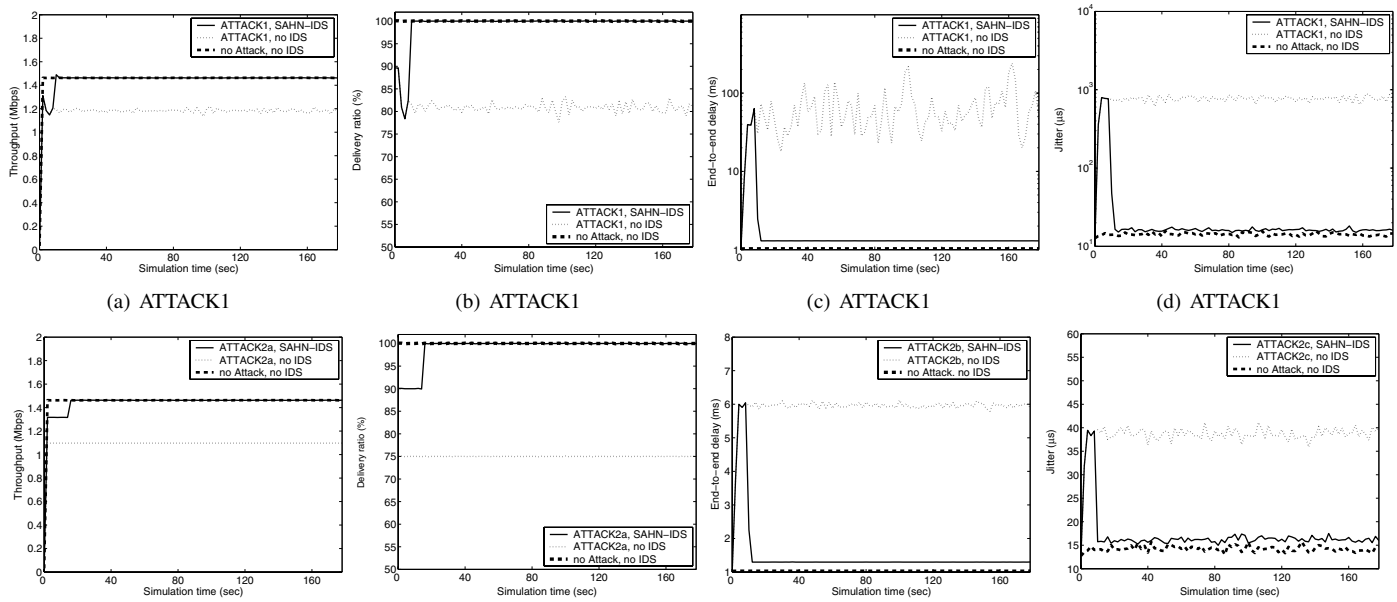
X. Simulation

We have used GloMoSim (version 2.02) for simulating various layers and wireless media. 70 nodes are placed on a 3000 meter by 3000 meter flat terrain where they are separated by at most 240 meters, use the same transmission power with an transmission range of maximum 240 meters, share the same frequency channel and use IEEE 802.11 in the link layer. The physical layer modulates/demodulates signals using OFDM (Orthogonal Frequency Division Multiplexing) with a transmission rate of 54 Mbps and uses a single network card with a single omnidirectional antenna. The routing has been done using DSR. To investigate the effectiveness of SAHN-IDS, we have integrated the detection schemes (except the directional antenna variation of ATTACK1 and scheme 2 of ATTACK2b) across the application, network and link layers.

For each simulation run, we have randomly selected twelve pairs of source and destination where each node pair is separated at least by four intermediate nodes and creates a 1.46 Mbps UDP type CBR (Constant Bit Rate) flow at random time. Once a flow is initiated, it is executed until the end of each simulation run. Simulation time for each run was set to 180 seconds. We have logged the values of various performance metrics (i.e. End-to-end delivery ratio, throughput, delay and jitter) every 2 seconds. The source and destination node pairs were evenly distributed over the whole network in order to reduce the effect of network congestion on the network performance for overlapping sessions so that only the misbehavior related effects become prominent. For each type of attack we have randomly selected four misbehaving nodes where each of them can initiate misbehavior at any time but continues to misbehave till the end of the simulation run. Each type of attack was performed in ten simulation runs to obtain average values of the associated performance metrics.

The threshold for the misbehavior counts, i.e. MCs, of various attacks was set to four. ATTACK1 was created by initiating a flow at the misbehaving node without reserving bandwidth where the channel capacity required for the flow was more than the available channel capacity. In this paper the response module only notifies the routing protocol to find an alternate route bypassing the misbehaving node. Other functionalities of the response system will be implemented in future.

Figure 2 shows the effectiveness of SAHN-IDS thorough various performance metrics. The sudden improvements in network performance indicate that SAHN-IDS is able to detect various attacks and can take necessary actions, i.e. find an alternate route, to achieve the desired network performance. The desired network performance has been shown by “no Attack, no IDS” line in each graph which was obtained by performing the simulations without any attack. However, the improvements in network performance did not come for free. The communication overhead in SAHN-IDS (i.e. messages of SAHN-IDS, route request and route reply for finding alternate routes) was increased by atmost 6%. But we believe that this increase in communication overhead will be compensated by



(a) ATTACK1 (b) ATTACK1 (c) ATTACK1 (d) ATTACK1

(e) ATTACK2a (f) ATTACK2a (g) ATTACK2b (h) ATTACK2c

Fig. 2. Effect of SAHN-IDS in networks that are under ATTACK1 and ATTACK2 but do not have any intrusion detection system. In each graph "no Attack, no IDS" acts as a benchmark to indicate the maximum possible performance that can be achieved if the networks were not attacked with ATTACK1 and ATTACK2.

performance improvements if there is any of the identified attacks in the network.

XI. Conclusion

We have proposed an IDS, referred to as SAHN-IDS, for SAHN like networks. SAHN-IDS can detect if nodes are getting their fair share of the transmission channel. It also detects packet drops or delays that violates the respective flow requirements. Unlike most IDSs for detecting packet drops or delays, SAHN-IDS does rely on overhearing packet transmissions of neighboring nodes which makes it an effective system in networks where nodes use different transmission power and directional antennas for different neighbors. SAHN-IDS does not require setting up various thresholds manually, rather it can select them dynamically. We have also shown the effectiveness SAHN-IDS through simulations with the expense of tolerable communication overhead. In future we want to implement all the causes of ATTACK1 and ATTACK2. We would also like to implement a fully functional response system that would incorporate all the features outlined in Sections VIII and IX.

References

[1] M. M. Islam, R. Pose, C. Kopp. Link Layer Security for SAHN Protocols. In *IEEE PerCom Workshops on PWN*, Mar 2005.

[2] Yih-Chun Hu, Adrian Perrig and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad-hoc networks. In *8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 23–28, Sep 2002.

[3] R. Pose, C. Kopp. Bypassing the Home Computing Bottleneck: The Suburban Area Network. *3rd Australasian Computer Architecture Conference*, pages 87–100, Feb 1998.

[4] M. M. Islam, R. Pose, C. Kopp. A Router Architecture to Achieve Link Rate Throughput in Suburban Ad-Hoc Networks. *LNCS, Springer-Verlag, Berlin, Germany*, pages 395–407, 2003.

[5] M. M. Islam, R. Pose, C. Kopp. Effects of Directional Antennas on 802.11e. In *IEEE and IFIP WOCN*, Mar 2005.

[6] M. M. Islam, R. Pose, C. Kopp. MAC Layer Support for Real-Time Traffic in a SAHN. In *ITCC*, Apr 2005.

[7] M. M. Islam, R. Pose, C. Kopp. Making SAHN-MAC Independent of Single Frequency Channel and Omnidirectional Antennas. In *IASTED NCS*, Apr 2005.

[8] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Mobile Computing and Networking*, pages 275–283, 2000.

[9] R. Ramanujan, A. Ahamad, J. Bonney, R. Hagelstrom and K. Thurber. Techniques for Intrusion-Resistant Ad hoc Routing Algorithms (TIARA). In *MILCOM*, volume 2, pages 660–664, Oct 2000.

[10] S. Marti and T. J. Giuli and K. Lai and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, 2000.

[11] L. Buttyán and J. P. Hubaux. Enforcing service availability in mobile ad-hoc WANs. In *IEEE/ACM MobiHOC*, pages 87–96, Aug 2000.

[12] Y. Zhang and W. Lee and Y. Huang. Intrusion detection techniques for mobile wireless networks. In *ACM/Kluwer Mobile Networks and Applications (MONET)*, volume 8, pages 545–556, Sep 2003.

[13] S. Bhargava and D. P. Agrawal. Security Enhancements in AODV protocol for Wireless Ad Hoc Networks. In *VTC*, volume 4, pages 2143–2147, Fall 2001.

[14] C. Perkins and E. Royer. Ad-hoc On-Demand Distance Vector Routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.

[15] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer and R. A. Kemmerer. An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks. In *Annual Computer Security Applications Conference*, Dec 2004.

[16] K. Ilgun, R. A. Kemmerer, and P. A. Porras. State Transition Analysis: A Rule-Based Intrusion Detection Approach. *IEEE Transactions on Software Engineering*, 21(3):181–199, 1995.

[17] D.B. Johnson and D.A. Maltz. *Dynamic Source Routing in Ad-hoc Wireless Networks*, chapter 5, pages 153–181. 1996.

[18] S. Buchegger, J-Y. L. Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness and Robustness in Mobile Ad Hoc Networks. In *Euromicro Workshop on Parallel, Distributed amd Network-based Processing*, pages 403–410, Jan 2002.

[19] S. Buchegger, J-Y. L. Boudec. Performance Analysis of the CONFIDANT Protocol. In *ACM MobiHoc*, pages 455–465, June 2002.

[20] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *IFIP CMS*, Sep 2002.

[21] K. Balakrishnan, J. Deng, P. K. Varhney. TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks. In *IEEE WCNC*, Mar 2005.

[22] Siddhartha Gupte and Mukesh Singhal. Secure routing in mobile wireless ad hoc networks. *Elsevier*, 1(1):151–174, July 2003.