Intrusion Detection System in Wireless Sensor Networks

Sarfraz Ali

Student of M.Tech. PCE Nagpur India

Abstract: ***Wireless sensor nodes have wide range of applications like monitoring data to surveillance ,healthcare to agriculture etc..In all these above monitoring system there is no any physical line of defense against data hacking and to stop malicious nodes, data stealing or destroying .So Intrusion Detection System(IDS) protects data from malicious nodes. In this paper we study to implement Intrusion Detection in sensor networks and propose a scheme that can be applied to such networks. Its basic concept is that nodes monitor their adjacent node and cooperate with their nearest neighbors to make the network back to its normal operational condition. We start in a simple approach in which, even though nodes don't have a global view, they can still detect an intrusion. We apply our design principles for the black hole and selective forwarding attacks by defining appropriate rules that characterize malicious behavior.***

Keywords: IDS WSN Malicious node Blackhole, Selective forwarding.

1. Introduction

Wireless sensor network (WSN) is a network is popular and simple sensor nodes that can sense temperature, humidity etc. and can communicate with each other using a wireless radio device. WSNs require the operation of a large number of sensor nodes. This raises problems for good maintenance and utilization. Sometimes it is not possible to reach the difficult a area at all, like for example in hostile, dangerous environments ,high low temperature areas or military applications. So, sensor networks should become self monitoring and exhibit response and adaptability to upcoming changes with time, without any user or human instructions..

Its utility is even more useful system when it comes to security threats. The routing nature shortest path of WSNs and the limited resources of their nodes make them prone to attacks. Any useful mechanism that can protect and provide their normal operation should be based on autonomous mechanisms within the network ih andtself.

Currently, research on providing security solutions for WSNs has focused mainly in three categories:

1) Signature management: It is the way for establishing cryptographic keys between nodes to enable encryption and authentication.
2) Authentication and Secure Routing: protocols are [2] Proposed and used to protect information from being revealed to an unauthorized party and guarantee its integral delivery to the base station.
3) Secure services: Service used to transmit data safely from source to destination has been made in providing Specialized secure services, like secure localization [3], secure collection[4] and secure time synchronization [5].

All security protocols are based on definite assumptions about the nature of attacks to the network. If the attacker is “weak”, the steps and protocols will achieve its security goal. This means that an enemy is prevented from breaking into a sensor network and stops its proper operation. If the attacker is “strong” there is a non-negligible probability that the intruder will break in. Because of their resource constraints, sensor nodes usually cannot deal with very strong consequences. So we need a second line of defense: An Intrusion Detection System (IDS) that can detect a third party's attempts of making possible insecurities and warn for malicious attacks, even if these attacks have not been experienced before.

Related Work

Intrusion detection is a vital aspect within the broader area of computer security. IDS systems, where certain monitor nodes in the network are responsible for monitoring their neighbors, looking for intruders. They listen to messages in their radio range and store in a buffer specific fields that may be useful to an IDS system running within a sensor node.

Two more IDSs for routing attacks in sensor networks. Some papers assume that routing protocols for ad hoc networks can also be applied to WSNs: the AODV (Ad hoc On-Demand Distance Vector) protocol, DSDV and DSR protocols. More productive work has been done in intrusion detection for ad hoc networks [10]. In such networks, distributed and cooperative IDS system is also used. Detailed distributed designs, actual detection techniques and their performance have been studied in more depth. While also being ad hoc networks, WSNs are much more resource oriented. In this paper attempts has been made to move towards direction, defining the requirements, studying the appropriate design choices and proposing a specific architecture appropriate for IDSs in WSNs.

Contributions

While completely take over nodes and extract their cryptographic keys [11], we assume that such an IDS cannot remove legitimate nodes by replicating captured nodes or introducing new ones in many parts of the network. This assumption is needed because an IDS for WSNs should exploit its nature in such a network to detect intrusion nodes to detect are the blackhole and selective forwarding attacks
The contribution of this paper is as follows -

1) First, we see the architectures of IDS systems and the details of which is the most appropriate for sensor networks. This is important as it will enable further work in this field that will also take into consideration the special properties of such networks.
2) Second, designing an IDS to detect blackhole and selective forwarding attacks [12], based on specification-based detection, requiring only small amounts of communication and calculation resources.
3) Finally, we show the effectiveness of our scheme by measuring the detection accuracy in a real simulated model.

The remainder of this paper is organized as follows. In Section II elaborate on the requirements that an IDS in sensor networks should have and in Section III consist of build step-by-step our proposed architecture. Then we present the overall design in a more modular and generalized form in Section IV. The performance of the proposed IDS is evaluated in Section V through simulations. Finally, Section VI concludes the paper.

2. Requirement for in Sensor Network

This section we focus on the requirements that an IDS system for sensor networks should satisfy. To do so, one has to look at the characteristics of these networks. Each sensor node has small communication and computational resources and a short radio range. Each node is a weak unit that can be easily compromised by an adversary [11], who can then load malicious software to launch an insider attack. We require that an IDS system for sensor networks must satisfy the following properties:

1) **Specified auditing:** IDS for sensor networks must work with specified and partial audit data. In sensor networks there are no centralized points that can collect global audit data, so this approach fits the sensor network module.
2) **Minimize resources:** IDS for sensor networks should utilize a small amount of resources. The wireless network does not have stable connections, and physical resources of network and devices, such as bandwidth and power, are limited. Disconnection can happen at any time. Communication between nodes for intrusion detection purposes should not take too much of the available bandwidth.
3) **Trust no node:** IDS cannot assume any single node is secure. Sensor nodes can be very easily compromised.
4) **Distributed model:** That means data collection and analysis is done on a number of locations. The distributed approach also applies to execution of the detection algorithm.
5) **Be secure:** IDS should be able to tolerate an attack within itself. Compromising a monitoring node and controlling the behavior of the IDS agent should not enable an adversary to start a node from the network, or keep another intruder node undetected.

3. Intrusion Detection in WSN

IDS architecture based on the above design goals. We break this into three parts. First, is about auditing mechanisms, then about detection algorithms and finally about decision making techniques. For each part we present the available solutions and we elaborate on which is more appropriate for sensor networks. Then we apply our findings to detect blackhole and selective forwarding [12] attacks.

A. Intrusion Detection Architecture

In sensor networks, most intruder targets the routing layer, since that allows them to take control of the information flowing in the network. Besides, sensor networks are mainly about reporting data back to the base station, and deviating this process makes an attack a successful one. So, for such networks, the most appropriate architecture for IDS would be network-based. A network-based IDS uses raw network packets as the data source. It listens on the network and captures and examines individual packets in real time. As all communication in the WSN is conducted over the air and a node can overhear traffic passing from a neighboring node, nodes can mutually check network traffic. Example, in [13] architecture for ad-hoc networks is shown

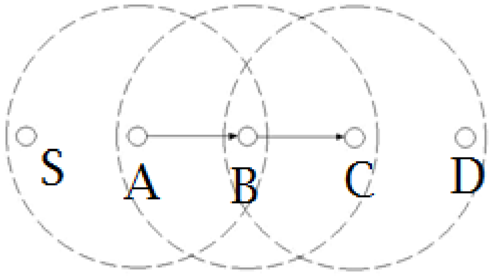

Figure 1: Node B is selectively forwarding packets to node C. Node A promiscuously listens to node B's transmissions.

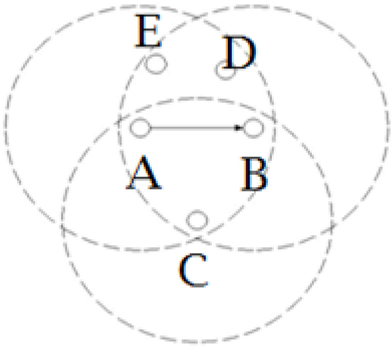

Figure 2: Nodes A, C , D and E can be watchdogs of the link A → B.

Where nodes are grouped in clusters and only the cluster-heads are responsible for monitoring the traffic within their clusters. However, a single monitor node fails to meet the "trust no node" requirement, since it could be captured and force the network to isolate another legitimate node. Instead, a certain fraction of nodes in an area should agree on an observation. If the number of nodes that can form such a detection quorum is larger than the number of nodes that can be captured by an adversary in the specific area, a simple majority vote can be used to form a decision. Let's consider again the example of Figure 1 and suppose that B is malicious. There are three cases, arising from the wireless nature of communications, where having a node A monitoring node B cannot result in a successful detection of node B:

1) At the same time that node B forwards its packet, another node S sends a packet to A, causing a collision at A. Node A cannot be certain which packets caused this collision, so it cannot conclude on B's behavior.
2) At the same time that node B forwards its packet to node C, node D also makes a transmission, causing a collision at C. Node A thinks that B has successfully forwarded its packet, since it doesn't know about the collision. Therefore, node B could skip retransmitting the packet, without being detected.
3) Node B waits until C makes a transmission, and then transmits its packet causing a collision at C. Again, node C never receives the packet, but node A cannot accuse B of anything.

From the above cases we can conclude that the watchdog approach should involve information from more than one node. So, for our intrusion detection system we require that any other neighbor of B that can listen to the packets this node is sending or receiving will participate in the intrusion detection procedure. In particular, for a link A → B, the watchdog nodes will be all the nodes that reside within the intersection of A's and B's radio range, including node A. For example, in Figure 2, the nodes A, C, D and E can be watchdogs for the communication between A and B.

We have simulated random topologies of 1000 uniformly distributed nodes and calculated the average number of watch-dogs for different network densities. What we have found is that for any communication link between two nodes and for any network density, the number of watchdogs on the average is approximately half the neighborhood size. So, for example, in a network where nodes have 8 neighbors the average number of watchdogs for any link is close to 4.

B. Intrusion Detection Techniques

Intrusion detection systems must be able to distinguish between normal and abnormal activities in order to discover malicious attempts in time. There are three main techniques that an intrusion detection system can use to classify actions [15]; misuse detection, anomaly detection and specification-based detection. In misuse detection or signature-based detection systems, the observed behavior is compared with known attack patterns (signatures). Action patterns that may pose a security threat must be defined and stored to the system. Then, the misuse detection system tries to recognize any “bad” behavior according to these patterns. It is already concluded from research in ad hoc networks that severe memory constraints make ID systems that need to store attack signatures relatively difficult to build and less likely to be effective [10].

Anomaly detection systems focus on normal behaviors, rather than attack behaviors. First these systems describe what constitutes a “normal” behavior (usually established by automated training) and then flag as intrusion attempts any activities that differ from this behavior by a statistically significant amount.

Specification-based detection systems are also based on deviations from normal behavior in order to detect attacks, but they are based on manually defined specifications that describe what a correct operation is and monitor any behavior with respect to these constraints. This is the technique we use in our approach. It is easier to apply in sensor networks, since normal behavior cannot easily be defined by machine learning techniques and training.

Since we follow the specification-based approach, we need to define which norms are going to be used to describe normal operation. These specifications for detecting black hole and selective forwarding attacks can simply be a rule on the number of messages being dropped by a node. Each of the watchdog nodes will apply that rule for itself to produce an intrusion alert. The naive approach would be to increment a counter every time a packet is dropped and produce an alert when this value reaches a threshold. However, we should take under consideration loss of messages due to other reasons, as those described in Section III-A. So, this approach will cause the counter of the watchdog nodes to increment and eventually reach the threshold value. Then the node would be charged without being malicious. If we consider a rate at which packets are being lost not by a selective forwarding attack, but because of other legitimate factors in the network, then in case of an attack, the packets will be dropped at a higher rate than they normally do. So, we need to set a threshold of the rate at which packets are dropped, and when this is reached an alarm can be generated. For that, we substitute the counter criterion with a rate criterion. To measure a rate we need to keep track of time duration. Therefore we require each watchdog node to keep track of the packets not being forwarded within a fixed amount of time, let's say w units, and we modify the intrusion detection rule as follows:

Rule 1: “For each packet that a node A sends to node B, temporally buffer this packet and wait to see if node B forwards it. If not, increment a counter corresponding to that node B. Else removes the packet from the buffer. If after w units the node has dropped more than t percent of the packets, produce an alert. ”

So, each watchdog node has a window of w units, during which it creates statistics on the overheard packets. At the end of each window an alert may be produced according to the threshold criterion, which is broadcasted by that node. Then the next window is started, and the same process is repeated periodically, for all watchdog nodes. We do not require that the nodes are synchronized, so the windows in each node are also not synchronized. They may have any time difference between 0 and w units.

C. Decision Making Techniques

The next design issue we need to solve is who is going to make the final decision that a node is indeed an intruder and actions should be taken. There are two approaches for this. Either we could use a cooperative mechanism or let nodes decide independently.

In an *independent decision-making* system, there are certain nodes that have the task to perform the decision-making functionality. They collect intrusion and anomalous activity evidences from other nodes and they make decisions about network-level intrusions. The rest of the nodes do not participate in this decision. For example, reviewing the architecture proposed in [13] for ad-hoc networks, the cluster-heads gather information from their cluster members and maintain a state machine for each one of them. Then the cluster-head can decide with a certain confidence that a node has been compromised by looking at reports regarding that node.

In such architectures, the decision-making nodes can attract the interest of an attacker, since compromising them would leave the network undefended. Another drawback of such an approach is that they restrict computation-intensive analysis of the overall network security state to just a few key nodes. This special mission of processing the information from other nodes and deciding on intrusion attempts results in an extra processing overhead, which may quickly lead to energy exhaustion.

In a *cooperative* IDS system, if an anomaly is detected by a node, or if the evidence is inconclusive, then a *cooperative* mechanism is initiated with the neighboring nodes in order to produce a global intrusion detection action. Even if a node is certain about the guiltiness of a suspicious node, still the decision should be cooperative, because, the node taking a decision could be malicious itself.

In our approach, we use a cooperative decision making approach, where the watchdog nodes of a link A → B cooperate in order to decide whether node B is launching a selective forwarding attack and take appropriate actions. In Section III-A we explained why a node cannot make such a decision on its own. So, we require that each node makes its final decision based on the alerts produced by all other watchdogs of the same link.

In order to build a cooperative decision mechanism, we take advantage of the fact that all watchdog nodes of a link are within communication range of each other. That means any watchdog node can listen to the messages broadcasted by the rest. So, it is easy for these nodes to announce their alerts to each other, by making a single message broadcast. With this knowledge, each node can make a safer conclusion by applying a majority rule:

Rule 2: *"If more than half (i.e., the majority) of the watchdog nodes have raised an alert, then the target node (i.e. node B) is considered compromised and should be revoked, or the base station should be notified."*

In particular, for the link A → B, we will define node A as the responsible node to gather the alerts from the rest of the watchdogs and apply the majority rule. We call that node the collector.

The rest of the watchdogs do not need to activate their cooperative detection engines for that link. So, the above majority rule states that for n watchdogs of a link A → B, if at least N_2 + 1 alerts are received by the collector A, including its own local alert, then a decision is made that node B is compromised. The problem that arises next is how long the collector should wait for the alerts.

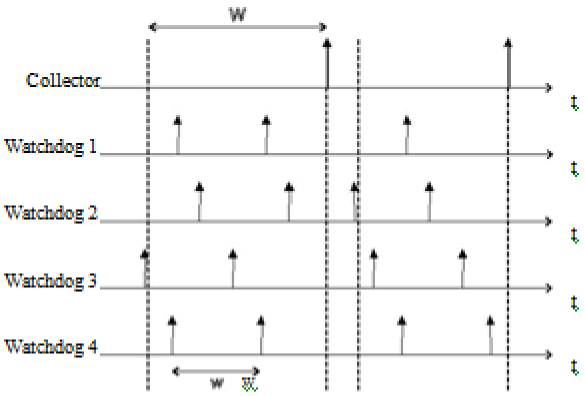

Figure 3: Cooperative detection mechanism applied by the collector. Each window **W** starts at the reception of the first alert from any watchdog, including the collector itself. In this example, **W** = 2**w**.

As we described in Section III-B, each watchdog node needs w units to decide whether a node is dropping packets at a higher rate than the normal. So, in order for the collector to receive the alerts from the rest of them, it has to wait for a longer interval of W units. Since we do not require the Watchdog nodes to be synchronized (see Figure 3), W must be long enough in order to ensure that any possible alerts from other watchdogs are received. In the worst case it has to be a little longer that w, but in the experimental section we show how other values of W affect the success of detection. Also note that if during that period, a second alert from the same watchdog arrives, then that alert is ignored in the application of the majority rule.

With this majority rule, if a watchdog is compromised and trying to revoke a legitimate node, or issues no alarms for another malicious node that launches an attack, it would have no effect because the majority would still prevail. However, if the collector itself is compromised, then the adversary can gain the control of the intrusion result. To avoid this scenario, we could have the rest of the watchdog nodes apply the majority rule over the alerts they receive and check their conclusions with the collector's report. Alternatively we could use a probabilistic version of verifiable agreement [16] in which the majority vote contains a cryptographic proof that it was formed based on real alarms of the watchdogs.

4. Building Blocks of IDS Client

In our discussion so far we have described all the opera-tions a sensor network IDS system should perform to detect blackhole and selective forwarding attacks. In this section, we formalize our approach by presenting a more modular architecture of the IDS system. We require that each node in the network has an IDS client with the following functionality:

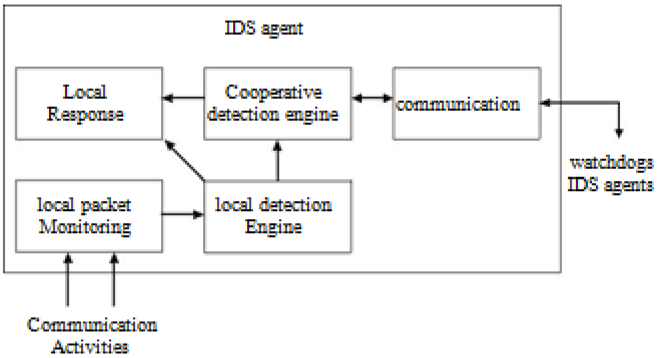

Figure 4: The building blocks of the IDS client existing in each sensor node.

- Network Monitoring: Every node performs packet monitoring in their immediate neighborhood collecting audit data.
- Decision Making: Using this audit data, every node decides on the intrusion threat level on a host-based basis. Then they publish their findings to their neighbors and make the final collective decision.
- Action: Every node has a response mechanism that allow it to respond to an intrusion situation.

Based on these functions we build the architecture of the IDS client based on five conceptual modules, as shown in Figure 4. Each module is responsible for a specific function, which we describe in the sections below. The IDS clients are identical in each node and they can broadcast messages for clients in neighboring nodes to listen. The communication amongst the clients allows us to use a distributed algorithm for the final decision on the intrusion threat.

A. Local Packet Monitoring

This module gathers audit data to be provided to the local detection module. Audit data in a sensor network IDS system can be the communication activities within its radio range. This data can be collected by listening promiscuously to neighboring nodes' transmissions.

B. Local Detection Engine

This module collects the audit data and analyzes it according to given rules. As we said in Section III-B, specification based detection is most appropriate for sensor networks, so the local detection engine stores and applies the defined specifications that describe what is a correct operation and monitors audit data with respect to these constraints.

C. Cooperative Detection Engine

If there is an evidence of intrusion, this module broadcasts the state information of the local detection process to the neighboring nodes. The same module in each node collects this information from all the neighboring nodes and applies a majority rule to conclude whether there is an intrusion or not. The input from the local detection engine is also counted in for this conclusion.

D. Local Response

Once the network is aware that an intrusion has taken place and have detected the compromised area, appropriate actions are taken by the local response module. The first action is to cut off the intruder as much as possible and isolate the compromised nodes. After that, proper operation of the network must resume. This may include changes in the routing paths, updates of the cryptographic material or restoring part of the system using redundant information distributed in other parts of the network. Autonomic behavior of sensor networks means that these functions must be performed without human intervention and within finite time.

Depending on the confidence and the type of the attack, we categorize the response to two types:

- **Direct response:** Excluding the suspect node from any paths and forcing regeneration of new cryptographic keys with the rest of the neighbors.
- **Indirect response:** Notifying the base station about the intruder or reducing the quality estimation for the link to that node, so that it will gradually lose its path reliability.

5. Experimental Evaluation

I have simulated a sensor network of 100 nodes placed uniformly random in order to test our proposed intrusion detection system. The network density was chosen so that each node has 8 neighbors on the average. Each time, we chose at random one link A → B and programmed node B to launch a selective forwarding attack, while node A was sending packets to it, at a given rate. This way we could have the watchdogs of that link A → B apply the intrusion detection and monitor the behavior of node B. With probability pD , node B was dropping the packets that were forwarded to it. Finally, we set the threshold value for the percentage of packets dropped over a period w to t = 20%. Above this threshold, each watchdog was generating an alert. Packets dropped at a lower rate were calculated to other factors, such as collisions or node failures, and did not produce an intrusion alert.

First we tested how the ratio of W and w effects the accuracy on intruder identification. The results are depicted in Figure 5, for 100 repetitions of the experiment. As we said in Section III-C, W must be bigger than w, so we did not simulate the case of $W/w < 1$. False negative rate represents the rate at which events are not flagged intrusive by the collector although the drop rate is higher than the threshold and the attack exists. If packets are dropped at a rate higher than the threshold t, then ideally, all windows W at the collector should give an alarm. However, since packets are dropped probabilistically, there might be the case that during a window w of some watchdogs, the dropped packets are less than t = 20%, and no alert is produced by those nodes. Then, the majority rule over a window W will not be satisfied, which will give no final alarm, producing a false negative.

This is less probable to happen as pD increases compared to t. In this case, the probability that during a window w the

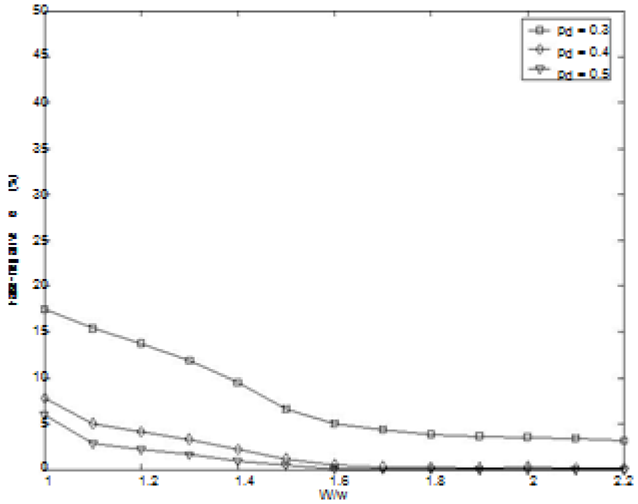

Figure 5: False-negative rate for different ratios of window length **W** to **w**

dropped packets are less than t resulting in a false negative is lower, and hence the better accuracy in detecting the attack.

We see from Figure 5 that as the window length W increases, the false negative probability decreases. This is because the collector can have a more accurate estimation as it gives more time to the watchdogs to produce their alarms. However, we cannot take W to be a very large quantity, since that would delay the detection of a compromised node. Therefore, for the rest of the experiments we fixed W = 2w.

Next we tested how the window length w affects the accuracy on intruder identification. All watchdogs are required to have the same window length. Given a steady packet rate, we measure this length in number of packets. Figure 6 shows the false negative rate for different number of packets monitored by the watchdogs. Then, for a fixed simulation time, we measured the number of final intrusion alerts produced by the cooperative engine at the collector. For the given window W (= 2w), each watchdog gathers the alerts broadcasted by the rest of them and applies the majority rule to produce a final decision.

Figure 6 shows that the false negative rate is reduced as the window length w is increased. For bigger w, more packets are monitored, and therefore, each watchdog has a better estimation of the drop rate and alerts are more successfully produced resulting in a cooperative detection at the collector. In the rest of the cases, the drop rate over the time period w for a watchdog may be statistically below the threshold, and no alert is produced. If this is true for more than half of the watchdogs, the majority rule fails and no detection is made.

Figure 7 depicts the number of alerts from the collector as a function of the drop probability pD . Two thresholds of 20% and 10% have been assumed for the local detection at the watchdogs. In all experiments we took W = 2w. The simulation time is fixed for 1000 repetitions and we set w to be long enough for 30 messages to be monitored at each watchdog.

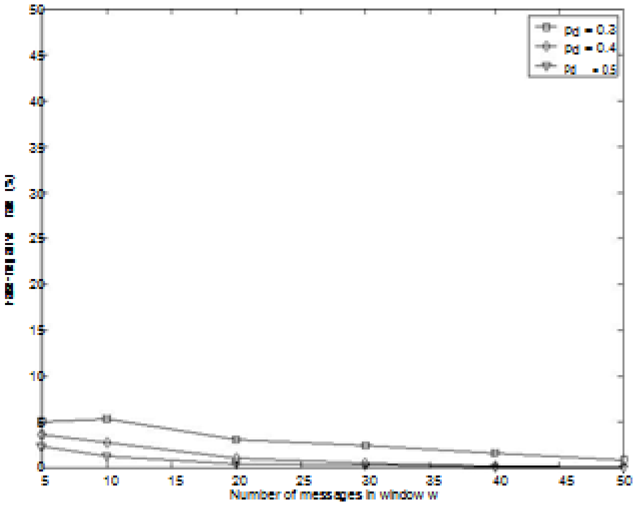

Figure 6: False-negative rate for different window lengths W.

Maximum number of final alerts that could be produced by the collector is 16, since this is the maximum number of windows W that fit in the fixed simulation time. For drop probabilities below the threshold a small number of alerts is produced. This is the number of *false positives* and ideally it should be zero. Since the packets are dropped probabilistically, there are cases where more than 20% (or 10% respectively) of the packets are dropped, even if the drop probability is lower. However, on the average, the cooperative mechanism produces a small number of false positives and this effect is shown clearly on smaller drop probabilities. For example, if we set the threshold t = 20% and assume that packets are dropped at a lower rate $p_D = 0.1$, then the graph indicates that the false positives will be 0.52, which is a rate of 0.52 × 100/16 = 3.25%.

6. Conclusions

In this paper we have introduced a model for distributed intrusion detection in sensor networks which is designed to work with only partial and localized information available at each node of the network. Nodes collaborate and exchange this information with their neighbors in order to make a correct decision on whether an attack has been launched. We focused on routing because it is the foundation of sensor networks. In particular, we demonstrated how our IDS system can be used to detect blackhole and selective forwarding attacks, producing very low false-negative and false-positive rates. We also provided a set of general principles that an IDS system for sensor networks.

References

[1] S. Camtepe and B. Yener, “Key distribution mechanisms fo r wireless sensor networks: a survey,” Rensselaer Polytechnic Instit ute, Troy, New York, Technical Report 05-07, March 2005.

[2] E. Shi and A. Perrig, “Designing secure sensor networks, ” IEEE Wireless Communications, vol. 11, no. 6, pp. 38–43, December 2004.

[3] L. Lazos and R. Poovendran, “Serloc: Robust localizatio n for wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 1, no. 1, pp. 73–100, 2005.

[4] T. Dimitriou and I. Krontiris, Security in Sensor Networks. CRC Press, 2006, ch. Secure In-network Processing in Sensor Networks, pp. 275– 290.

[5] S. Ganeriwal, S. Capkun, C.-C. Han, and M. Srivastava, “S ecure time synchronization service for sensor networks,” in Proceedings of the 4th ACM workshop on Wireless security (WiSe '05), 2005, pp. 97–106.

[6] [6] A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, “Decentralized intrusion detection in wireless sens or networks,” in Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks (Q2SWinet '05). ACM Press, October 2005, pp. 16–23.

[7] Onat and A. Miri, “An intrusion detection system for wi reless sensor networks,” in Proceeding of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3, Montreal, Canada, August 2005, pp. 253–259.

[8] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, “Intrus ion detection for routing attacks in sensor networks,” International Journal of Distributed Sensor Networks, 2005.