Neighbor-based Intrusion Detection for Wireless Sensor Networks

Neighbor-based Intrusion Detection for Wireless Sensor Networks

Andriy Stetsko, Lukáš Folkman, Vashek Matyáš
Faculty of Informatics, Masaryk University, Brno, Czech Republic
xstetsko@fi.muni.cz, xfolkman@mail.muni.cz, matyas@fi.muni.cz

***Abstract*—The neighbor-based detection technique explores the principle that sensor nodes situated spatially close to each other tend to have a similar behavior. A node is considered malicious if its behavior significantly differs from its neighbors. This detection technique is localized, unsupervised and adapts to changing network dynamics. Although the technique is promising, it has not been deeply researched in the context of wireless sensor networks yet. In this paper, we present symptoms which can be used in the neighbor-based technique for detection of selective forwarding, jamming and hello flood attacks. We implemented an intrusion detection system which employs the neighbor-based detection technique. The system was designed for and works on the TinyOS operating system running the Collection Tree Protocol. We evaluated accuracy of the technique in the detection of selective forwarding, jamming and hello flood attacks. The results show that the neighbor-based detection technique is highly accurate, especially in the case when collaboration among neighboring nodes is used.**

***Keywords*-intrusion detection system; neighbor-based detection technique; wireless sensor network**

I. Introduction

A wireless sensor network (WSN) is a highly distributed network of resource-constrained and wireless devices called sensor nodes. Each sensor node monitors some physical phenomenons (e.g., humidity, temperature, pressure, light) inside the area of deployment. The collected measurements are sent to a base station. The communication range of sensor nodes is limited to tens of meters and hence not all of them can directly communicate with the base station. Therefore, data are sent hop-by-hop from one sensor node to another until they reach the base station.

Security becomes an important issue for WSNs and brings new challenges for security engineers. Cryptographic techniques can be used to prevent an external attacker from eavesdropping or altering the ongoing communication. The area of deployment is not usually physically protected and an attacker can easily access the area and capture some nodes. Since they are not tamper-resistant, the attacker can modify the software running on the nodes to launch a variety of internal attacks such as selective forwarding, jamming and hello flood attacks. We assume that the number of captured nodes is significantly smaller than the total number of sensor nodes in the network. However, the damage caused by internal attacks might be significant.

Intrusion detection systems (IDSs) are proper mechanisms to defend against internal attacks and they are widely used in conventional networks. However, these IDSs cannot be directly applied to WSNs mainly due to severe limitations of sensor nodes on energy, memory and computational power. It is enough to have several probes in conventional networks if they are placed in traffic concentration points. In WSNs, some attacks can be observed only by the neighbors of a malicious node. Hence, we assume that each sensor node runs an IDS agent and monitors its neighbors in a promiscuous mode. The collected data, we believe, should be analyzed locally by a sensor node itself or in cooperation with only nodes from the close neighborhood since communication is highly energy consuming – one transmitted bit consumes about as much power as executing 800 instructions [1].

In this work, we consider the neighbor-based anomaly detection technique [2][3]. The basic idea is that neighboring nodes should be dealing with similar network traffic and provide similar values from their sensors. If a node's behavior significantly differs from its neighbors, the node is considered malicious. The technique is unsupervised (it does not require training on labeled data), localized and capable of adapting to changing network dynamics. Although these properties are welcome in WSNs, the technique was not deeply researched yet.

In Section II, we describe basics of the neighbor-based detection technique and summarize the related work. Section III contains a description of selective forwarding, jamming and hello flood attacks together with the symptoms (and corresponding statistics) that can be used in the neighbor-based technique to detect these attacks. In Section IV, we present additional symptoms (and corresponding statistics) which we used in this work. In Section V, we provide design and implementation details of the proposed intrusion detection system. We evaluated accuracy of the neighbor-based technique in detection of selective forwarding, jamming and hello flood attacks. The results are provided in Section VI. We conclude the work in Section VII.

II. Related work

We present the works of Liu et al. [2] and Li et al. [3], which applied the neighbor-based technique to detect insider attacks in WSNs.

A. Data collection

In [2], every sensor node a_i ($i = \overline{1,n}$) monitors its direct neighbors $N(a_i) = \{b_{i1}, ..., b_{im_i}\}$ in a promiscuous mode.

Here n denotes the total number of nodes in a network and m_i denotes the number of neighbors of the node a_i. Our proposal follows the same approach.

In [3], a network is partitioned into groups. Sensor nodes are in the same group if they are physically close to each other and their sensed data are dissimilar at most δ. The monitoring node a_i collects audit data about its neighbors but only those from the same group.

B. Detection

In both [2] and [3], the networking behavior of the node b_{ij} observed by the node a_i is modeled by the q-component vector $f(b_{ij}) = \{f_1(b_{ij}), ..., f_q(b_{ij})\}$ with each component describing the node's activity in one aspect. If the node a_i monitors the nodes $\{b_{i1}, ..., b_{im_i}\}$, it stores the set of the corresponding attribute vectors $F(a_i) = \{f(b_{ij}) : j = \overline{1, m_i}\}$. We use the same notations in this work.

Both schemes are based on the assumption that in any local area, all attribute vectors $f(b_{ij}) = \{f_1(b_{ij}), ..., f_q(b_{ij})\}$ follow the same multivariate normal distribution.

The node a_i considers the node b_{ij} as suspicious if the (Mahalanobis) distance from $f(b_{ij})$ to the "center" of the set $F(a_i)$ is greater than a predefined threshold. The authors of both works use orthogonalized Gnanadesikan-Kettenring estimation to find the "center" of the set $F(a_i)$.

The authors of [2] claim that nodes can be evaluated in terms of packet dropping rate, packet sending rate, forwarding delay time and sensor readings. The authors of [3] claim that their scheme can be used to detect insider attacks by monitoring sensor sensed data, packet sending rate, packet dropping rate, packet mismatch rate, packet receiving rate and received signal strength respectively. Unfortunately, the authors of [2][3] do not provide any notion of circumstances under which the assumption (regarding multivariate normal distribution) holds.

In [2], the experiments were done only on synthetic data which follow multivariate normal distribution. In [3], the experiments were done on real data that contain only sensor readings. Such behavioral characteristics as packet sending rate, packet dropping rate, packet receiving rate and received signal strength remain unexplored. Our experiments revealed that the packet sending, packet dropping and packet receiving rates did not follow the normal distribution for networks operating a tree-based routing protocol (without data aggregation) [4].

C. Cooperation

In [3], the node a_i alerts other sensor nodes in the same group. Each sensor node in the group decides whether a node is malicious or not based on the number of alerts received from other nodes and its own observations. In [2], sensor nodes share their observed attribute vectors in the close neighborhood. The authors describe how to filter different attribute vectors regarding the same node that is two hops away – the vector coming from the most trustworthy relay is chosen. However, they do not provide any information on how to combine different attribute vectors regarding the same 1-hop neighbor. In our work, we consider several such options (see Section V-B).

III. Attacks and their symptoms

In this section, we describe selective forwarding, hello flood and jamming attacks together with their symptoms which can be used in the neighbor-based technique to detect these attacks. For information on other attacks (including packet alteration attack), see [4].

A. Jamming

In a jamming attack a malicious node purposefully tries to interfere with physical transmission and reception of wireless communication. In this paper, we consider deceptive, random and reactive jammers. A *deceptive jammer* constantly injects regular packets without any gap between subsequent packet transmissions. A *random jammer* alternates between sleeping and jamming randomly. A *reactive jammer* stays quiet when the channel is idle but starts transmitting as soon as it senses activity on the channel [5].

Received signal strength is a measurement of the power present in a received radio signal. The signal strength of the node b_{ij} received at the node a_i is denoted as $SS(b_{ij})$.

The distribution of $SS(b_{ij})$ is affected by a deceptive jammer [5]. The node a_i considers the node b_{ij} malicious if the distribution of $SS(b_{ij})$ significantly differs from distributions of $\{SS(b_{i1}), ..., SS(b_{im_i})\}$. The approach cannot be applied to reactive and random jammers [5].

Packet sending rate is the number of packets sent over a predefined period of time. The packet sending rate of the node b_{ij} observed by the node a_i is denoted as $PS(b_{ij})$.

PS of a deceptive jammer significantly differs from the neighboring nodes since it sends an abnormal amount of packets in comparison to legitimate nodes.

Packet receiving rate is the total number of packets received over a predefined period of time. We differentiate two types of a packet receiving rate – with and without retransmissions. The first one is denoted as $PR(b_{ij})$ and the second one as $PR'(b_{ij})$.

The packet receiving rate in combination with the packet sending rate can be used to detect a random jammer (see received-to-sent ratio in Section IV).

Packet delivery ratio is the ratio of packets that are successfully delivered to a destination compared to the number of packets that were sent by a sender. It can be calculated as the ratio of the number of received acknowledgements with respect to the number of sent packets. The packet delivery ratio of the node b_{ij} calculated at the node a_i is denoted as $PDR(b_{ij})$.

The packet acking rate, i.e., the number of acknowledgements sent to the node b_{ij} which were observed by the node a_i, is denoted as $PA(b_{ij})$.

Presence of any type of a jammer in a node's neighbourhood can be noticed as the node's packet delivery ratio drops down [5]. However, the attacker itself is not revealed.

B. Hello flood

In a hello flood attack [6], a malicious node broadcasts hello packets using a more powerful transceiver than a general sensor node does. Nodes receiving such hello packets may falsely assume that they are within the radio range of the sender and try to forward their packets through this malicious node. These packets will be lost since they will not even reach the malicious node.

Received signal strength measured by the nearest neighbor of a malicious node is significantly higher than signal received from other neighbors.

C. Selective forwarding

This attack is performed by a malicious node that refuses to forward certain packets and simply drops them [6].

Packet dropping rate is the number of packets that were sent to a certain node but were not forwarded by that node. The packet dropping rate of the node b_{ij} observed by the node a_i is denoted as $PD(b_{ij})$.

A node that has its packet dropping rate significantly higher than other nodes in the neighborhood is considered malicious. Sensor nodes that are close to each other should have similar packet dropping rates. If a part of a network is congested or effected by environmental changes then majority of nodes in that part will have the packet dropping rates increased.

Packet forwarding rate of a certain node is the number of packets that the node received from its neighbors and consequently forwarded to its parent node during a predefined period of time. The packet forwarding rate of the node b_{ij} observed by the node a_i is denoted as $PF(b_{ij})$.

IV. New statistics

Using simulations we analyzed distributions of the packet forwarding, packet dropping, packet sending and packet receiving rates for the Collection Tree Protocol (CTP) operating network [4]. The simulation setup is described in Section VI-B. We concluded that even for sensor nodes in the close neighborhood these statistics do not follow the normal distribution. Of course, any sample can be approximated by the normal distribution but such approach will result into a high number of false negatives and positives. Unfortunately, Liu et al. [2] evaluated their scheme only on synthetic data, while Li et al. [3] did that only on sensor readings.

The simulations also revealed that packet receiving rates (and hence packet sending, packet dropping and packet forwarding rates) of neighbors might differ significantly (see Figure 1). Although packet dropping rate and packet sending rate can be used in the neighbor-based technique for detection of selective forwarding and random jamming attacks respectively (see Section III), they will result into a high number of false positives and false negatives in networks operating tree-based routing protocols without data aggregation [4]. Further in the text, we present the statistics which mitigate the mentioned problem to an acceptable level for such networks.

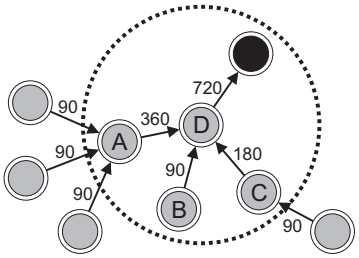


Figure 1: Packet receiving rate. The receiving rates of the nodes A, B and C are 270, 0 and 90 respectively.

In order to detect the selective forwarding attack, we propose to calculate a *packet dropping ratio* – the ratio of the number of dropped packets to the number of received packets without retransmissions. This value should be significantly higher for an attacker when compared to normal nodes from its neighborhood (see Rule 3 in Section V-A).

In order to detect deceptive and random jamming attacks, we propose to calculate a *received-to-sent ratio* – the ratio of the number of received packets to the number of sent packets. The attacker sends some "extra" packets which it did not receive to forward and hence it has a lower received-to-sent ratio than it is common in its neighborhood then (see Rule 4 in Section V-A).

In order to detect the reactive jamming attack, we propose to use the *packet delivery ratio* (PDR) as it was defined in Section III-A. The definition of the ratio comforts deployment for detections in CTP operating networks. PDR should be extremely low for the node that is being jammed (see Rule 5 in Section V-A).

V. IDS design and implementation

The design of our IDS is based on the conceptual architecture proposed in [7]. Every node runs an agent which monitors the information flowing in its neighborhood. The agent consists of data acquisition, statistics, detection, alert database and collaboration components. Further in the text, we present only the detection and collaboration components. For information on other components, see [4].

A. Detection

The detection component is started periodically, after a predefined period of time has elapsed. For each attack the information from the statistics component is sorted by a specific parameter. Then, only the first 20 nodes of the resulting sorted set are passed for the detection itself. The choice of the sorting parameter is done so that the possible attackers

are among these 20 nodes with the highest probability. The records about the nodes $\{b_{i1}, ..., b_{im_i}\}$ stored at the node a_i are sorted according to the SS for the detection of the hello flood attack (obviously, the attacker is among the nodes with the highest SS). Nodes having high PR are attractive for the attacker to capture when discussing selective forwarding and so we sort them according to their PR for the detection of this attack. A deceptive jammer stands out among its neighbors because of its extremely high PS. Even for the case of a random jammer, its PS most probably exceeds the average PS in its neighborhood. Hence, the statistics are sorted according to the PS in both cases. Sorting according to the PDR is used for the detection of reactive jamming as the presence of this attack can be noticed based on identifying nodes with extremely low PDR.

We assume that non-malicious nodes must prevail in a neighborhood over malicious nodes. Otherwise, the technique will not be able to detect the malicious nodes. The node b_{ij} is considered malicious by the node a_i if the Euclidean distance from $f_k(b_{ij})$ to the "center" of the set $\{f_k(b_{i1}), ..., f_k(b_{im_i})\}$ is greater than δ_k. We define the "center" of the set as the arithmetic average of its elements. Furthermore, from the set of suspicious nodes we exclude the nodes having the value of the attribute $f_m(b_{ij})$ lower than γ_m. The attribute f_m is chosen per each attack and γ_m has to be set so that only nodes which cause minimal threat are excluded. Formal definition of the detection rule follows:

$$f_k(b_{ij}) - AVG(f_k(b_{i1}), ..., f_k(b_{im_i})) > \delta_k \quad and\ f_m(b_{ij}) > \gamma_m \tag{1}$$

For the detection of the hello flood attack:

$$SS(b_{ij}) - AVG(SS(b_{i1}), ..., SS(b_{im_i})) > \delta_{SS} \tag{2}$$

Selective forwarding attack:

$$\frac{PD(b_{ij})}{PR'(b_{ij})} - AVG(\frac{PD(b_{i1})}{PR'(b_{i1})}, ..., \frac{PD(b_{im_i})}{PR'(b_{im_i})}) > \delta_{\frac{PD}{PR'}} \quad and\ PD(b_{ij}) > \gamma_{PD} \tag{3}$$

Deceptive and random jamming:

$$AVG(\frac{PR(b_{i1})}{PS(b_{i1})}, ..., \frac{PR(b_{im_i})}{PS(b_{im_i})}) - \frac{PR(b_{ij})}{PS(b_{ij})} > \delta_{\frac{PR}{PS}} \quad and\ PS(b_{ij}) > \gamma_{PS} \tag{4}$$

Reactive jamming:

$$AVG(\frac{PA(b_{i1})}{PS(b_{i1})}, ..., \frac{PA(b_{im_i})}{PS(b_{im_i})}) - \frac{PA(b_{ij})}{PS(b_{ij})} > \delta_{\frac{PA}{PS}} \quad and\ PS(b_{ij}) > \gamma_{PS-RETRY} \tag{5}$$

B. Collaboration

Knowing precise values of packet dropping, sending, receiving, acking and forwarding rates as well as received signal strength is important for the detection of attacks (see Section V-A). However, the knowledge is always limited, e.g., due to different types of collisions [8] and the fact that the node a_i does not hear all of the neighbors of b_{ij}. In order to refine the audit data, a node broadcasts messages to its neighbors, each containing the node's attribute vectors. On receiving such a message, the IDS agent stores the records from the message into the collaboration database.

No reputation scheme is implemented in this work and the received information is assumed to be correct.

It is common that several IDS agents running on neighboring nodes monitor the same node. Several records describing the same node can be found in the collaboration database then. We present several rules to filter records having the one of the most precise values of some attack-relevant attribute. Furthermore, these rules take into account another attribute needed for the detection of the attack in order to eliminate the number of false positives. The rules are formally stated below.

The record candidate at the node a_i is chosen from the set of the records $\{b_1, ..., b_{m_l}\}$ and is denoted as b in this text. b_m ($m = \overline{1, m_l}$) is the record about the neighbor b_{ij} received from the neighbor b_{im}. It is presumed that the record claiming the highest SS, PR, PS or PA is the one closest to the reality.

For the case of the hello flood attack it is:

$$SS(b) = MAX(SS(b_1), ..., SS(b_{m_l})) \tag{6}$$

Selective forwarding attack:

$$PR(c_k) > \frac{1}{q} \times MAX(PR(b_1), ..., PR(b_{m_l})) \quad and\ PF(b) = MAX(PF(c_1), ..., PF(c_{m_k})) \tag{7}$$

Deceptive and random jamming:

$$PS(c_k) > \frac{1}{q} \times MAX(PS(b_1), ..., PS(b_{m_l})) \quad and\ PR(b) = MAX(PR(c_1), ..., PR(c_{m_k})) \tag{8}$$

Reactive jamming:

$$PA(d) = MAX(PA(b_1), ..., PA(b_{m_l})) \quad and\ PS(c_k) > \frac{1}{q} \times PS(d)\ and \quad PDR(b) = MAX(PDR(c_1), ..., PDR(c_{m_k})) \tag{9}$$

q was set equal to $1.\overline{33}$ in our implementation.

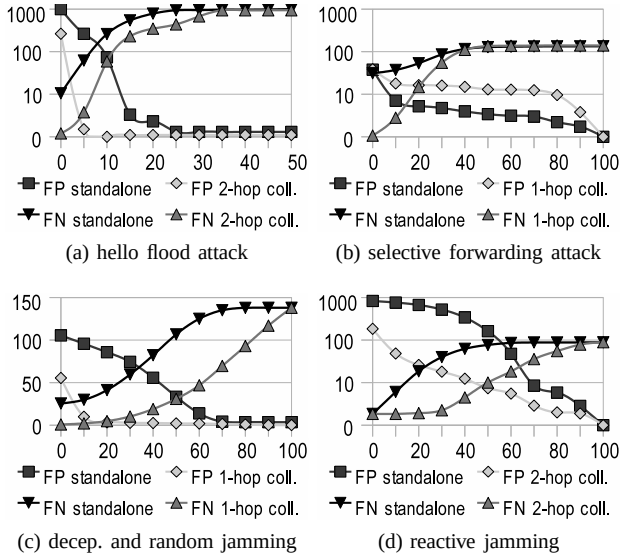
VI. Experiments

A. Evaluation metrics

In order to evaluate the proposed IDS, the number of false positives and the number of false negatives are counted. We defined and discussed these metrics in [9].

If a node considers its neighbor malicious, while it is not, this is considered as a false positive. The number of false positives is the total number of such events in the network.

If a node that runs the IDS agent does not detect its neighboring malicious node, this is considered as a false



(a) hello flood attack (b) selective forwarding attack

(c) decep. and random jamming (d) reactive jamming

Figure 2: Relationship between the number of false positives (FP), false negatives (FN), the Y axis, and the value of the detection threshold, the X axis.

negative. The number of false negatives is the total number of such events in the network. Several false negatives may refer to the same node since the malicious node might have more than one neighbor which did not detect it. Besides, the number of false negatives per each attacker was counted too and compared with the number of IDS agents that successfully revealed the attacker (we talk of the number of correct detections of an attacker in this respect).

B. Network and IDS configuration

A network of 224 sensor nodes was simulated in the TOSSIM – a simulator for TinyOS applications. A node sent a measurement to the gateway using the CTP every 16 seconds (the minimum interval tested by the authors of the CTP [10]). A network was set to low gain mode monitoring with a threshold of -88 dB [11]. In such configuration, a node has 28 neighbors on average [4].

The detection period was set to 304 seconds. Also the setting with both 34 and 608 seconds were tested. While the first one did not provide good detection results, the second one did not improve results significantly in comparison to the 304 seconds long period detections. Five detection rounds were evaluated in 10 simulations for each attack and results were averaged (they were stable among each detection round and simulation). Both standalone and collaboration (1-hop and 2-hop) detections were simulated using the same data so that results could be compared. Five attackers were involved in each simulation when detections of hello flood, selective forwarding, deceptive and random jamming attacks were tested. Two nodes were unable to send out packets due to reactive jamming in simulations concerned to this attack.

C. Hello flood

The hello flood attacker was implemented as a node whose signal strength is 40 dB stronger than it would be if a normal node were situated at the same place in the network. Relationship between the number of false positives (false negatives) and the signal strength detection threshold (δ_{SS}) can be seen in Figure 2a for standalone and 2-hop collaborating IDS agents.

When collaboration was used and δ_{SS} was set to 5 dB, 3.76 false negatives and 0.28 false positives were announced on average (see Figure 2a). Each of the five attackers was reported in 223.23 alerts per detection round [4].

In conclusion, hello flood attackers can be efficiently revealed using the collaborating neighbor-based IDS.

D. Selective forwarding

In our simulations, the selective forwarders were chosen randomly from internal nodes of the CTP tree that forward at least 300 messages per detection period (there were 21 such nodes on average). These nodes could be considered steady forwarders which would have not tended to become leaf nodes later during the simulation. The selective forwarder dropped every other packet in our implementation.

For the case of the detection of the selective forwarders γ_{PD} was set to value 19 (the IDS announced up to this number of dropped packets on average about non-malicious nodes during the simulations). When $\delta_{\frac{PD}{PR'}}$ was set to 21, the standalone IDS reported 55.47 false negatives and 5.13 false positives (about 1.6 different nodes) on average. Having the same value of $\delta_{\frac{PD}{PR'}}$ set (approx. the intersection of the collaboration curves in Figure 2b) and the collaboration module enabled, the number of false negatives decreased to 16.67. At the same time, the number of false positives increased to 16.33 (about 1.27 different nodes) on average. Each of the attacker was revealed by 15.56 and 24.65 IDS agents for the case of the standalone and 1-hop collaboration enabled system respectively [4].

Although the number of false positives was high in the case of the collaborating system, the number of different non-malicious nodes reported by these alerts was even lower than in the case of the standalone system. It can be concluded that our 1-hop collaborating IDS is capable of revealing selective forwarding attacks.

E. Deceptive and random jamming

The deceptive jammer was implemented as a node continuously injecting packets regardless of the medium access protocol properties. In [11], we successfully detected the deceptive jammer based on its high PS.

The random jammer was implemented as a node which sends packets randomly in interval between 0.28 and 0.48 seconds. Hence, it sends from 633 to 1085 packets per detection round (in comparison, 16.76 out of 224 nodes send more than 700 packets and 6.56 of them send more

than 1100 packets in each detection round). γ_{PS} was set to 699. We believe that node which sends less then 699 packets per detection period is not able to cause jamming (as non-malicious nodes tend to send this amount of packets too).

The results for standalone IDS configuration are not satisfactory as it can be seen in Figure 2c. The 1-hop collaborating IDS provided optimum results when the received-to-sent ratio threshold ($\delta_{\frac{PR}{PS}}$) was set to 15 (approx. the intersection of the false positives and negatives curves). The number of false negatives was 3.36 and there were 4.24 false positives (about 2.8 different nodes) on average in every detection round (see Figure 2c). Each of the attackers was correctly detected by 26.93 IDS agents on average [4].

The results suggest that our IDS with 1-hop collaboration can be used to detect deceptive and random jammers.

F. Reactive jamming

The reactive jammer was not implemented in our work so far. However, the simulations were run in such manner so that some nodes' communication was jammed by some number of other nodes. These simulations can be viewed as that there was ongoing reactive jamming in the network.

$\gamma_{PS-RETRY}$ was set to 49 (as the number of retransmissions is 30 and a node sends 19 packets per detection period). The standalone IDS was not able to cope with the detection of reactive jamming as it reported too many false positives and false negatives at any value of the packet delivery ratio threshold $\delta_{\frac{PA}{PS}}$ (see Figure 2d). On the other hand, for the case of the 2-hop collaborating IDS agents, the number of false positives was 8.24 (reporting 3.48 nodes) and the number of false negatives was 8.12 on average in every detection round. $\delta_{\frac{PA}{PS}}$ was set to 47 (approx. the intersection of the curves in Figure 2d). At the same time, there were 31.96 and 27.4 correct alerts reporting the two jammed nodes [4]. These results suggest that our IDS with the 2-hop collaboration enabled can be used to identify reactive jamming.

VII. Conclusion and Future Work

We explored hello flood, selective forwarding and jamming attacks in order to find out if they can be detected using the neighbor-based technique. Several symptoms of attacker's and victim's behavior which can be used in neighbor-based IDS were presented in Section III.

The main problem we encountered while trying to apply the neighbor-based technique for detection of selective forwarding and random jamming attacks in CTP operating networks (without data aggregation) was that packet dropping rate and packet sending rate respectively resulted into a high number of false positives and false negatives. In Section IV, we proposed the statistics which mitigated the mentioned problem to an acceptable level for such networks. We implemented an IDS for the TinyOS operating system which uses the received signal strength, packet delivery ratio as well as proposed packet dropping ratio and received-to-sent ratio to detect selective forwarding, jamming and hello flood attacks. We evaluated the implemented IDS in the TOSSIM simulator. We found out that our IDS is capable of detecting all of the attacks mentioned above with reasonably low occurrence of false positives and negatives when collaboration among nodes is employed.

Further, we plan to identify how much energy is consumed by the proposed IDS in different settings as well as extend it by a reputation system in order to cope with malicious nodes propagating false information.

Acknowledgment

This work was supported by the project 102/09/H042 of the Czech Science Foundation and by the project 1M0545 of the Czech Ministry of Education, Youth and Sports.

References

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proceedings of ACM ASPLOS*, pp. 93–104, 2000.

[2] F. Liu, X. Cheng, and D. Chen, "Insider attacker detection in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, pp. 1937–1945, 2007.

[3] G. Li, J. He, and Y. Fu, "A group-based intrusion detection scheme in wireless sensor networks," in *Proceedings of GPS – Workshops*, pp. 286–291, IEEE, 2008.

[4] A. Stetsko, L. Folkman, and V. Matyas, "Neighbor-based intrusion detection for wireless sensor networks," Tech. Rep. FIMU-RS-2010-04, Faculty of Informatics, Masaryk University, May 2010.

[5] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *Network, IEEE*, vol. 20, no. 3, pp. 41–47, 2006.

[6] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[7] R. Roman, J. Lopez, and S. Gritzalis, "Situation awareness mechanisms for wireless sensor networks," *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 102–107, 2008.

[8] I. Krontiris, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks." 13th European Wireless Conference, 2007.

[9] A. Stetsko and V. Matyas, "Effectiveness metrics for intrusion detection in wireless sensor networks," in *EC2ND. Milan, Italy. To appear in IEEE Computer Society*, 2009.

[10] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "The collection tree protocol," in *Proceedings of ACM SenSys*, pp. 1–14, 2009.

[11] L. Folkman, "Neighbour-based intrusion detection in wireless sensor networks," Master's thesis, Masaryk University, 2010.