A Mobility and Energy-aware Hierarchical Intrusion Detection System for Mobile ad hoc Networks

Eleni Darra, Christoforos Ntantogian, Christos Xenakis, Sokratis Katsikas

Department of Digital Systems, University of Piraeus, Greece

email: {edarra, dadoyan, xenakis, ska}@unipi.gr

Abstract. This paper presents a hierarchical cluster-based IDS architecture for Mobile Ad-hoc NETworks (MANETs) that considers the mobility and energy of nodes in the cluster formation in order to improve detection accuracy and reduce energy consumption. The proposed architecture adopts and enhances the Mobility and Energy Aware Clustering Algorithm (MEACA), which is the most appropriate for IDS in MANETs, since it aims at forming mobility aware and energy efficient 1-hop clusters. The algorithm maximizes the clusters' stability by choosing nodes with relatively low mobility and high energy to be the cluster-heads and keeping the constructed clusters unchanged to the extent of their maximum possible lifetime. The key advantage of the proposed IDS is that its detection accuracy is not affected from nodes mobility, since each cluster includes nodes with similar direction and speed. Thus, mobile nodes of the same cluster appear more static to each other eliminating cluster reformation, which negatively affects the detection accuracy. Moreover, the distribution of the detection load is based on the remaining energy in each node. Thus, nodes with adequate energy undertake more detection responsibilities than nodes with low power. In this way, the proposed IDS balances the energy consumption in a fair and efficient manner.

Keywords. Intrusion Detection System, IDS, mobile ad hoc networks, MANETs, hierarchical architecture, clustering algorithm, mobility, energy.

1 Introduction

A mobile ad hoc network (MANET) is an autonomously formed collection of mobile nodes without the involvement of any established infrastructure or centralized control. In MANETs, the nodes themselves communicate with each other creating dynamic network topologies. Due to their unpredictable topology, wireless shared medium, heterogeneous resources and stringent resource constraints, MANETs are vulnerable to a variety of attacks (i.e., target routing, cooperation, confidentiality, integrity, etc.) and thus, sufficient protection from them is fundamental requirement. The implementation of an Intrusion Detection System (IDS) can identify such attacks and trigger the appropriate protection mechanisms [1].

An IDS for MANETs can be divided into two parts: (i) the architecture which explicates its operational structure and (ii) the detection engine that is the mechanism used to detect malignant behaviors. The existing IDS architectures for MANETs fall under three main categories [2 - 4]: stand-alone, cooperative and hierarchical. As hierarchical IDSs apply multiple levels of detection, they present increased detection accuracy compared to others (i.e, stand alone and cooperative). They mainly bring about low processing and communication overhead by employing voting schemes to elect cluster-heads (CH), which monitor large portions of the network reaching more accurate decisions [13]. Moreover, they attempt to distribute fairly the processing workload among the nodes, considering the remaining energy power. Finally, effort has been put to create more robust hierarchies under high node's mobility, by selecting CHs with the objective of lasting longer [9].

A limitation of the existing IDSs for MANETs is that they do not consider the negative impacts of mobility on the detection accuracy [13]. More specifically, in various mobility scenarios the changes in topology and routing tables are rapid and inconsistent. These changes may occur also from malicious behaviors that attempt to disrupt the network operation and routing process. An IDS should distinguish which changes are legitimate, caused by nodes mobility and which are the results of abnormal behaviors, provoked by malicious nodes. Nevertheless, IDSs may erroneously indentify legitimate changes as attacks and vice versa, increasing in this way the ratio of false positives and negatives (i.e., detection accuracy). Moreover, the creation and maintenance of clustered/hierarchical structures adds extra processing load to the network nodes, which also increases under conditions of relatively high nodes' mobility. This overhead is produced by the continuous execution of the clustering functionality, due to the constant change of clusters. Finally, the majority of the existing IDS do not take into account that the detection process should not increase significantly the energy consumption at the level of nodes. Especially, in cooperative IDS architectures, where each node runs a detection engine, the energy consumption may be significantly high, reducing the lifetime of nodes in the network.

Driven by the above observations, this paper presents a hierarchical cluster-based IDS architecture for MANETs that considers the mobility and energy of nodes in the cluster formation in order to improve detection accuracy and reduce energy consumption. The proposed IDS architecture adopts and enhances the mobility and energy aware clustering algorithm (MEACA) [12], which aims at forming mobility aware and energy efficient 1-hop clusters. The algorithm maximizes the clusters' stability by choosing nodes with relatively low mobility and high energy to be the CHs and keeping the constructed clusters unchanged to the extent of their maximum possible lifetime. The key advantage of the proposed IDS is that the detection accuracy is not affected from the nodes mobility, since each cluster includes nodes with similar direction and speed. Thus, mobile nodes of the same cluster appear more static to each other eliminating cluster reformation, which negatively affects the detection accuracy. Moreover, the distribution of the detection load is based on the remaining energy in each node. Thus, nodes with adequate energy undertake more detection responsibilities than nodes with low power. In this way, the proposed IDS balances the energy consumption in a fair and efficient manner. Finally, the proposed IDS minimizes the communication overhead as there is 1-hop distance between a CH and its cluster members (CMs).

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 elaborates on the proposed IDS architecture and the MEACA algorithm. Section 4 evaluates the proposed IDS architecture focusing on its advantages and disadvantages and, finally, section 5 draws the conclusions.

2 Related Work

There is a rather limited literature of IDS for MANETs that copes with the impact of mobility on the detection accuracy, while at the same time takes into account the energy consumption at the level of nodes. The hierarchical IDS architecture, proposed by Ma and Fang [5], follows a modular approach based on clusters and presents a number of strengths including: (i) the nodes with the highest battery power are elected to serve as CHs, (ii) it supports two layers of detection (i.e., local and network) providing increased detection accuracy, and (iii) the CH monitors the network packets exchanged thus, there is no extra communication overhead between the CH and the CMs. The major drawback is that high nodes' mobility may reduce the detection accuracy of the IDS and increase the ratio of false positives, since a number of nodes may move out of the range of a CH. This limits the information that the network detection module may use to perform detection.

Otrok et al. have proposed a hierarchical approach [6] that attempts to balance the consumption of resources (which results from intrusion detection tasks) among the nodes of a cluster. It encourages network nodes to participate in the election of CHs and tries to prevent elected CHs from misbehaving. One of the main operational strengths of this architecture is that the nodes with the highest battery power are elected to serve as CHs. On the other hand, there is no discussion regarding the mobility of nodes and its implications in the detection accuracy.

Marchang and Datta [7] have proposed two IDS architectures that rely on a voting scheme to perform intrusion detection, instead of employing an anomaly or signature based intrusion detection engine. The main disadvantages of these two are: (i) high ratio of false alarms, since they do not take into account mobility, and (ii) they do not consider energy consumption.

H. Deng et al. in [8] have proposed a clustered IDS architecture in which only the CHs carry out intrusion detection. It focuses on detecting attacks that target the routing infrastructure of a network and forms clusters using the "Distributed Efficient Clustering Approach" protocol. Although this architecture distributes fairly the processing workload among the nodes, as the CHs rotate after a certain period of time, there is no analysis of the mobility implications in the detection accuracy.

Manousakis et al. [9] have proposed a hierarchical IDS architecture that uses a dynamic tree-based structure in which detection data are aggregated upwards, from leaf nodes to authoritative nodes at the root of the hierarchy (i.e., upper layer nodes), and the latter dispatch directives down to the former (i.e., lower-level nodes). The tree-based structure is established and maintained using two algorithms: the initial solution generation and the state transition mechanism. The main drawback of this algorithm is that the election process of CHs does not consider the energy of nodes.

Finally, Sun et al. [10], [11] have proposed a cooperative IDS architecture that focuses on routing disruption attacks using an intrusion detection engine based on statistical methods with adjustable threshold values. The technique of adjustable thresholds ensures that periodical changes in routing information, caused by nodes' mobility, remain under the detection threshold, while malicious behaviors that are persistent exceed the thresholds indicating the occurrence of attacks. This addresses the negative impacts of mobility on the detection accuracy. However, the simulation analysis revealed that the decrease in the ratio of false positives was relatively low.

3 The Proposed IDS

3.1 IDS Architecture

The proposed IDS architecture is organized into autonomous and distributed multi-leveled hierarchies. Each level of them consists of several clusters in which specific nodes act as CHs gathering local audit data from its CMs, analyzing them and extracting conclusions about the integrity of the nodes in the cluster. The autonomous clusters-based hierarchies are formed using the MEACA algorithm [12] which can be applied in dynamically changed network topologies.

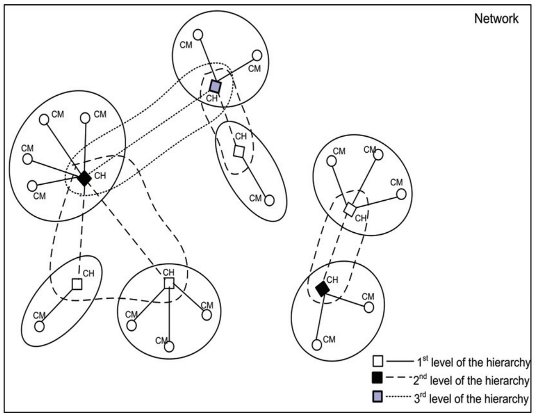
Network
CM
CM
CH
CM
CM
CM
CM
CH
CM
CH
CM
CM
CM
CM
CH
CM
CH
CH
CM
CM
CM
CM
CH
CM
CM
□ —— 1st level of the hierarchy
■ – – 2nd level of the hierarchy
▨ ······ 3rd level of the hierarchy

Fig. 1. Graphical example of the proposed IDS architecture

Initially, the algorithm creates the first level of hierarchies by forming autonomous clusters. Afterwards, the CHs of the previously formed clusters are selected to participate in the next level of hierarchies. Some of them will keep their attributes acting as CHs in the new level, while many others will act as CMs. Generally, the algorithm is repeated until the higher level of each hierarchy consists of a single node (i.e., hierarchy CH). It is important to note that in each layer the nodes that participate in a cluster should have 1-hop distance among them. A graphical example of the proposed

IDS architecture is represented in **Fig. 1**. Intrusion detection occurs at the CH of each cluster by aggregating data from the CMs in order to have increased detection accuracy. If the responsible CH cannot detect an attack accurately, it forwards the detection data to the CH of the upper level, if such exists.

3.2 Algorithm Description

The multi-level hierarchies of the proposed IDS have the following characteristics: (i) every node in the network becomes either a CH or a CM, (ii) every node is associated with only one cluster in each level of the hierarchy, and (iii) every CM is 1-hop distance from its CH. During the algorithm execution, every node sends attribute values to its neighbor nodes (i.e., nodes that have 1-hop distance). Each node keeps a neighborhood table that includes information regarding A_m (i.e., mobility attribute), A_e (i.e., energy attribute) and the related node *ID*. When an attribute's value is received, the corresponding entry of the sending node in the table is updated. If a node no longer receives any value from a neighbor node then, the related entry in the neighborhood table is cleared. A_m and A_e are required to determine a node's priority to become a CH. A_m measures the mobility stability of a nodes and A_e measures the remaining time of a node before its energy is ended.

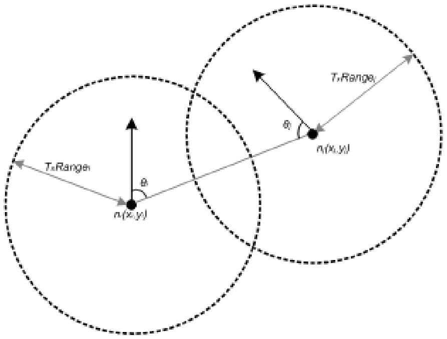

Fig. 2. LET parameters between two nodes

The mobility stability is defined by the Link Expiration Time (LET) [9], [14], [15]. By predicting the LET for any link on a route R, the route's R expiration time is estimated as the least of the LET values of all links on R. Based on this prediction, routes are reconfigured before they disconnect [16]. In [15] a mobility prediction method is presented for estimating the expiration time of the wireless link between two ad hoc nodes. The estimation of the LET, or in other words the time period T that two ad hoc nodes being within mutual transmission range (i.e., remain connected) is done as follows: If the motion parameters of two neighbors (such as speed, direction, and radio propagation range) are known, we can determine the duration of time these two nodes will remain connected. Assume two nodes i and j . Let (x_i , y_i) be the coordinates of mobile host i and (x_j , y_j) be that of mobile host j . Also let v_i and

v_j be the speeds, and θ_i and θ_j (*where* $0 \le \theta_i, \theta_j < 2\pi$) be the moving directions of the nodes i and j, respectively. Also, $TxRange_i$, $TxRange_j$ are the transmission ranges of nodes i and j. In this situation, the $TxRange_i$, $TxRange_j$ is the same. So, the LET_{ij} for the direct link between nodes n_i and n_j is defined in **Fig. 2**:

$$LET_{ij} = \begin{cases} \dfrac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2}, & nodes\ i,j\ are\ in\ range \\ 0 & , nodes\ i,j\ are\ not\ in\ range \\ \infty & , nodes\ i,j\ are\ relatively\ static \end{cases}$$

where:

$a = v_i\cos\theta_i - v_j\cos\theta_j$, $b = x_i - x_j$, $c = v_i\sin\theta_i - v_j\sin\theta_j$, $d = y_i - y_j$, $r = TxRange_{i,j}$

The host with a large value of average LET is able to maintain relatively long connection with their neighbor hosts [15] [16]. Therefore, the average LET of each host, which can be calculated in a distributive manner, can be used as mobility metric for the CH selection.

```
N: Node, T: Attributes' table, $A_e$: Energy attribute, $A_m$: Mobility attribute, ID: identity of a node
Step I: N receives $A_e$, $A_m$, ID from its neighboring nodes
Step II: N creates table T with $A_e$, $A_m$, ID of its own and its neighboring nodes
Step III: N selects the highest value of $A_m$ in T, defined as max ($A_m$)
Step IV: N determines threshold $A_m$* = a × max ($A_m$), where a ∈ (0, 1)
Step V: N removes from T the nodes with $A_m$ < $A_m$*. Let T' be the remaining nodes in T.
Step VI: N chooses from T' the node with the highest $A_e$. Let C be this node.
            {
              if node C is the node N
                then N becomes CH
              else
                N sends a registration request message to C. If the latter accepts,
                 then C becomes CH and N becomes CM of C
            }
Step VII: Steps I to VI are repeated from the elected CHs to create the next level of the hierarchy
```

Table 1. Pseudo code of the proposed algorithm for the formation of the hiearchies

At the initialization of the algorithm, all nodes are in an undecided role state, where they do not know yet which is the CH or CM. When a node *N* executes the algorithm to determine whether it will become a CH or CM, first it requests from its neighboring nodes (i.e., nodes with 1-hop distance) their attributes values (i.e., *Step I*). Next, the node *N* creates an attributes' table *T* with the received *ID*, A_m and A_e including its own attribute values (i.e., *Step II*). Then, *N* selects the node with the highest value of A_m defined as *max* (A_m) (i.e., *Step III*). In the next step (i.e., *Step IV*), *N* determines a mobility threshold defined as A_m* = *a* × *max* (A_m), where *a* ∈ (0, 1). The parameter *a* is selected randomly from node *N*. Based on the mobility threshold value, the node *N* excludes the nodes from the attributes' table that have A_m lower than A_m*. In this way, it achieves to eliminate the unstable nodes. Let *T'* be the remaining nodes after the elimination of unstable nodes. After this, node *N* selects, from the remaining nodes in table *T'*, the node with the highest A_e (i.e., *Step V*). Let *C* be the node with the highest A_e from the remaining nodes. If *C* is the same node as node *N*, then it becomes a CH. Otherwise, *N* sends a registration request message to *C*. If the latter accepts it, then *C* becomes a CH and node *N* becomes a CM of node *C* (i.e., *Step VI*). The process continues until all the nodes define their CH. Note that each node that becomes a CM sends to its neighboring nodes advertisements with *ID*, A_m, A_e values equal to NULL to acknowledge that it cannot be a CH. For the creation of the next levels of the hierarchy, steps I to VI are repeated between the CHs (i.e., *Step VII*). After the creation of all the hierarchies, each node periodically broadcasts their *ID*, A_m and A_e values to its neighbor nodes to acknowledge any changes in the registration tables (see section 3.3). **Table 1** gives the pseudo code of the algorithm for the formation of the hierarchies.

3.3 Case Study

For a better understanding of the presented notions, in this section we apply the aforementioned algorithm in a MANET comprised of 15 nodes (see **Fig. 3**).

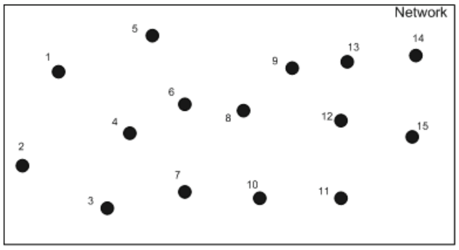

Fig. 3. The structure of the network

For the presented example we set various values of the mobility A_m and energy A_e attributes measured in seconds. Moreover, node 1 receives attributes values from nodes 2, 3, 4 and 5 which has 1-hop distance with them.

Node 1	ID	A_m	A_e
	1	742	4097
	2	826	2539
	3	560	6088
	4	663	4772
	5	368	5982

Node 2	ID	A_m	A_e
	2	826	2539
	3	560	6088
	4	663	4772

Node 4	ID	A_m	A_e
	3	560	6088
	4	663	4772
	6	632	4700
	7	392	3909

Table 2. Example of attributes' tables of nodes 1, 2 and 4

As shown in **Table 2**, in the attributes' table of the node 1, node 2 has the greatest value of A_m and thus, *max* $(A_m) = 826$. Assuming now that node 1 selects α=*0,7*, the mobility threshold is calculated as $A_m^* = 0,7 \times 826 = 578,2$. Thus, node 5 is excluded from the process as its mobility attribute is lower than A_m^*. From the remaining nodes (1, 2, 3 and 4), node 1 selects the node with the highest A_e, that is node 3 (see **Table 2**). Since node 3 is not the node executing the algorithm (i.e., node 1), a registration message is sent from node 1 to node 3. If the latter accepts, then it becomes CH while node 1 becomes its CM.

Next, node 2 receives attribute values from nodes 3 and 4. Observing **Table 2**, it is evident that node 2 has the greatest A_m and, therefore $max\ (A_m) = 826$. Next, node 1 selects $\alpha=0,57$ and the mobility threshold is derived as $A_m^* = 0,57 \times 826 = 470,82$. Since all nodes have mobility attribute A_m higher than the mobility threshold, none of these nodes are excluded. Next, node 2 selects node 3 with the highest A_e. Node 3 is not the one executing the algorithm, and therefore, node 2 sends a registration message to node 3. The latter has previously accepted this message and node 2 becomes its CM. Since nodes 1 and 2 have the same CH (i.e., node 3) they belong to the same cluster.

Then, node 4 receives attributes values from nodes 3, 6 and 7. Observing **Table 2**, node 4 has the greatest A_m value with $max\ (A_m) = 663$. Node 4 selects $\alpha=0,9$ and thus, $A_m^* = 0,9 \times 663 = 596,7$. Nodes 3 and 7 are excluded, since their mobility attributes have smaller values than A_m^*. From the remaining nodes, node 4 has the highest A_e. Since node 4 is the same as the one executing the algorithm, it becomes a CH.

The algorithm is executed from all nodes to become either CHs or CMs. At the end of the formation of the first level of the hierarchy, nodes 3, 4, 5, 6, 12 and 15 are CHs. More specifically, nodes 1 and 2 have node 3 as CH; node 4 and 5 CHs, node 7 and 8 have node 6 as CH; node 9, 10 and 13 have node 12 as CH; and nodes 11 and 14 have node 5 as CH. This is also depicted in **Fig. 4**.

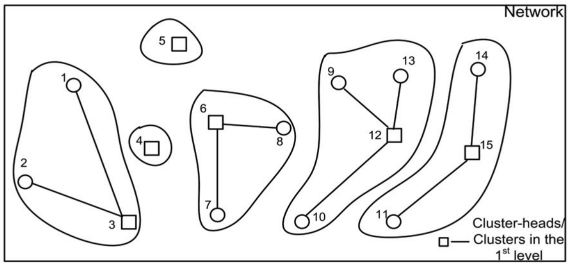

Fig. 4. 1st level of the hierarchy

The second level of hierarchies is formed by the elected CHs of the first level of hierarchy by executing the proposed algorithm again. The nodes that participate in the next level are: 3, 4, 5, 6, 12 and 15. After the algorithm execution, the formation of new clusters is depicted in **Fig. 5**.

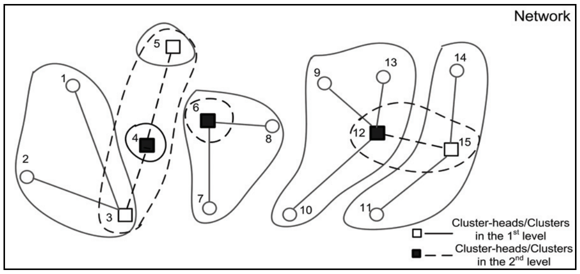
Network
5
1
4
6
8
2
7
3
9
13
14
12
15
10
11
Cluster-heads/Clusters in the 1st level
Cluster-heads/Clusters in the 2nd level

Fig. 5. 2^{nd} level of the hierarchy

For the next level, the CHs of the previous level are selected, that is node 4, 6 and 12. From these, node 6 does not participate, since in the previous level it created a cluster with only itself. In this case, the specific hierarchy has already been completed. Each of the participating nodes receives attribute values from others. However, this is not possible in the considered case because the distance between nodes is greater than 1-hop. Therefore, node 4 and 12 are root CHs of the related hierarchies, as depicted in **Fig. 6**.

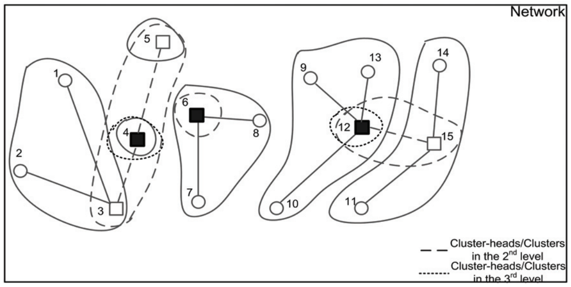
Network
5
1
4
6
8
2
7
3
9
13
14
12
15
10
11
Cluster-heads/Clusters in the 2nd level
Cluster-heads/Clusters in the 3rd level

Fig. 6. 3^{rd} level of the hierarchy

After determining roles, each node maintains registration tables, one for every level of hierarchy that participates. If a node is a CH, the registration table keeps the *IDs* of its CMs. If a node is a CM, the registration table keeps the *ID* of its CH. If a node creates a cluster by itself then, the table keeps its *ID*. **Table 3**, **Table 4** and **Table 5** present the registration table of the 1^{st}, 2^{nd} and 3^{rd} level of the hierarchy.

N 1	**ID: 3**	**N 2**	**ID: 3**	**N 3**	**ID: 1** **ID: 2**	**N 4**	**ID: 4**	**N 5**	**ID: 5**
N 6	**ID: 8** **ID: 7**	**N 7**	**ID: 6**	**N 8**	**ID: 6**	**N 9**	**ID:12**	**N10**	**ID:12**

N11	ID:15
N 12	ID: 9 ID:10 ID:13
N13	ID:12
N14	ID:15
N15	ID:11 ID:14

Table 3. 1st level of hierarchy

A node uses its registration table to decide when it needs to re-cluster. Re-clustering takes place in case that the registration table of a node becomes empty. This means that if a node is CH, it re-clusters only when it loses contact with all its CMs. On the other hand, if it is a CM, it re-clusters when it loses contact to the CH. In any case, re-clustering is performed, locally, in the nodes neighborhood.

N 3	ID: 4
N 4	ID: 3 ID: 5
N 5	ID: 4
N 6	ID: 6
N12	ID:15
N15	ID:12

Table 4. 2nd level of the hierarchy

N 4	ID: 4
N12	ID:12

Table 5. 3rd level of the hierarchy

4 Evaluation

The proposed IDS architecture considers nodes' mobility in the formation of clusters in order to improve detection accuracy. In general, nodes' mobility decreases the detection accuracy by increasing false positives and negatives. This happens because an IDS cannot distinguish which changes in the network topology and routing tables are legitimate, caused by nodes mobility and which are the results of abnormal behaviors, provoked by malicious nodes. The proposed IDS tries to minimize these by creating clusters that includes nodes with similar direction and speed. Thus, mobile nodes of the same cluster appear more static to each other, eliminating in this way the negative effects of mobility in the detection accuracy.

Moreover, the distribution of detection load is based on the remaining energy in each node. Thus, nodes with adequate energy undertake more detection responsibilities than nodes with low power. In this way, the proposed IDS balances the energy consumption in a fair and efficient manner.

The proposed IDS also attempts to minimize the imposed communication and processing overhead, which disrupts the network operation. More specifically, it minimizes the processing overhead by employing detection engines only at some key nodes (i.e., CHs), while the remaining nodes do not use any detection engine. Although, in general, the creation and maintenance of clusters add extra processing workload to the network nodes, the clusters created by the proposed algorithm are as stable as possible and do not change frequently. In our approach the communication overhead is limited to the minimum since every CM is 1-hop away from its CH. The 1-hop distance is also valid in each level of the hierarchy among the CMs.

Another advantage of the proposed IDS is that it attempts to reduce the bandwidth consumption in each level of the hierarchy. As data travel up the levels of hierarchy, significant data reduction/aggregation may be possible at intermediate levels, reducing the bandwidth consumed in transit. Finally, the proposed IDS architecture tries to perform detection at the lower possible level of the hierarchy if sufficient audit data exist. In this way, it minimizes the bandwidth consumption and communication overhead by avoiding sending audit data to the higher levels.

On the other hand, a critical issue that should be investigated further lies in the fact that the proposed hierarchical IDS architecture may impose unfair workload distribution among the network nodes, since the nodes elected as CHs are overloaded with detection responsibilities. Another issue is that a malicious node or set of nodes may be elected as CHs hindering or misleading intrusion detection.

5 Conclusion

This paper presents a hierarchical cluster-based IDS architecture for MANETs that considers the mobility and energy of nodes in the cluster formation, in order to improve detection accuracy and reduce energy consumption. The proposed IDS architecture adopts and enhances the MEACA [12] algorithm which maximizes the clusters stability by choosing nodes with relatively low mobility and high energy to become CHs. The key advantage of the proposed IDS is that the mobile nodes of the same cluster appear more static to each other, eliminating in this way the negative effects of mobility in the detection accuracy. Moreover, the distribution of the detection load is based on the remaining energy of each node. Thus, nodes with adequate energy undertake more detection responsibilities than nodes with low power. In this way, the proposed IDS balances the energy consumption in a fair and efficient manner. Finally, it minimizes the communication overhead due to the 1-hop distance between a CH and its CMs. As a future work, we will conduct simulations to estimate the detection accuracy (also false positive, false negative) of the proposed IDS under various mobility and attacks scenarios.

6 References

1. A. Mishra, K. Nadkarni, A. Patcha, “Intrusion Detection in Wireless Ad Hoc Networks,” IEEE Wireless Communications, Vol. 11, Issue 1, pp. 48-60, 2004.
2. M. Rafsanjani, A. Movaghar, F. Koroupi, “Investigating Intrusion Detection Systems in MANET and Comparing IDS for Detecting Misbehaving Nodes”, in Proceeding of the World Academy of Science, Engineering and Technology, 34, pp 351-5, 2008

3. Anantvalee, Tiranuch and Wu, Jie, “A survey on intrusion detection in mobile ad hoc networks”, In: Xiao, Y., Shen, X., Du, D.-Z. (Eds.), Wireless/Mobile Network Security, Springer. pp. 170-196.
4. C. Panos, C. Xenakis, I. Stavrakakis, “A Novel Intrusion Detection System for MANETs”, In Proc. of International Conference on Security and Cryptography (SECRYPT 2010), Athens, Greece, 2010.
5. C. Ma, Z. Fang, “A Novel Intrusion Detection Architecture Based on Adaptive Selection Event Triggering for Mobile Ad-hoc Networks”, In Proc. IEEE Second International Symposium on Intelligent Information Technology and Security Informatics, p. 198-201, 2009
6. H. Otrok, N. Mohammed, L. Wang, M. Debbabi, P. Bhattacharya, “A game-theoretic intrusion detection model for mobile ad hoc networks”, Elsevier Computer Communications vol. 31, Issue 4, pp. 708-721, 2008
7. N. Marchang, R. Datta, “Collaborative techniques for intrusion detection in mobile ad hoc networks”, Elsevier Ad Hoc Networks, vol. 6, Issue 4, pp. 508–523, 2008
8. H. Deng, R. Xu, J. Li, F. Zhang, R. Levy, W. Lee, “Agent-based cooperative anomaly detection for wireless ad hoc networks”, In: Proc. of the 12th conference on parallel and distributed systems, pp. 613-620, 2006
9. K. Manousakis, D. Sterne, N. Ivanic, G. Lawler, A. McAuley, “A stochastic approximation approach for improving intrusion detection data fusion structures”, In Proc. of IEEE Military Communications Conference (MILCOM 2008), San Diego, CA, pp. 1-7, 2008
10. B. Sun, K. Wu, Y. Xiao, R. Wang, “Integration of mobility and intrusion detection for wireless ad hoc networks”, Wiley International Journal of Communication Systems, vol. 20, Issue 6, pp. 695 – 721, 2007
11. B. Sun, K. Wu, UW. Pooch, “Routing anomaly detection in mobile ad hoc networks”, In Proc. of IEEE international conference on computer communications and networks (ICCCN’03), pp.25-31, 2003
12. Y. Xu, W. Wang, “MEACA: Mobility and Energy Aware Clustering Algorithm for Constructing Stable MANETs”, In Proc. of IEEE Military Communications Conference (MILCOM 2006), Washington, D.C., pp.1-7, 2006
13. C. Xenakis, C. Panos, I. Stavrakakis “A Comparative Evaluation of Intrusion Detection Architectures for Mobile Ad Hoc Networks,” Computers & Security, Elsevier Science, Vol. 30, Issue 1, pp: 63-80, 2011
14. S. Leng, Y. Zhang, H. Chen, L. Zhang, K. Liu, “A Novel k-Hop Compound Metric Based Clustering Scheme for Ad Hoc Wireless Networks”, IEEE Transactions On Wireless Communications, Vol. 8, Issue 1, pp. 367 – 375, 2009
15. S.-J. Lee, W. Su, M. Gerla, “Ad hoc Wireless Multicast with. Mobility Prediction”, In Proc. of IEEE ICCCN'99, Boston, pp. 4-9, 1999
16. D. Gavalas, C. Konstantopoulos and G. Pantziou, "Mobility Prediction in Mobile Ad Hoc Networks", in "Next Generation Mobile Networks and Ubiquitous Computing", S. Pierre (Ed.), ISBN10: 160566250X, IGI Global, USA, Chapter 21, pp. 226-240, 2010.