2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications

An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs

Leovigildo Sánchez-Casado, Gabriel Maciá-Fernández and Pedro García-Teodoro
Department of Signal Theory, Telematics and Communications
Faculty of Computer Science – University of Granada
Granada 18071, Spain
{sancale, gmacia, pgteodor}@ugr.es

***Abstract*—This paper introduces a novel IDS approach for detecting malicious packet dropping behaviors in MANETs. Although some similar proposals can be found in the specialized literature, two main differences exist with ours. First, mobility aspects are explicitly considered into the approach by means of a heuristic which considers the operation of the forwarding process at the nodes. Second, this fact results in a significant improvement in the detection performance of the system, especially in terms of low false positive rate. The different experimental results obtained show the promising nature of our approach, both in terms of the detection capabilities exhibited and from the point of view of the simplicity of the scheme.**

***Index Terms*— mobile ad hoc networks; packet dropping; malicious behavior; false positive; mobility**

I. INTRODUCTION

A MANET is a particular type of network composed of a set of self-configurable mobile devices, geographically distributed in a given area and without a fixed infrastructure or centralized administration. Nodes that are within the communication range communicate directly, while those which are out of the range make use of other nodes to relay their messages to reach the destination (multi-hop strategy). These principal characteristics make this kind of networks an optimal and particularly useful candidate in certain areas, such as environmental or military applications, disaster management, etc. However, as MANETs proliferate, many security issues associated with this communication paradigm become more relevant and thus need to be conveniently addressed.

Among others, packet dropping attacks are one of the most disruptive threats in MANETs. Nodes exhibiting this behavior maliciously drop received data or routing messages instead of forwarding them. This way, the normal operation of the network is disrupted [1]. Different categories can be considered to classify this kind of attacks depending on the particular strategy adopted. The most popular are *black hole* and *gray hole* attacks. When the node completely drops all the received packets, this is considered as a black hole attack. On the other hand, the gray hole attack is caused by a node dropping packets in a selective way, e.g. one out of N packets received, one packet every certain time, only packets corresponding to specific flows, etc.

There are several motivations for a node to evade its responsibility on forwarding packets in the network. For example, a node may refuse to relay packets in order to preserve or economize its energetic resources. These nodes are usually known as *selfish nodes*. On the other hand, malicious nodes try to introduce themselves in the routing/forwarding path in order to seize communications. To do so, they modify routing messages either by publishing that they have the shortest path to the destination or by spoofing the destination address to guarantee that the sender chooses them as the next hop.

Focused on detecting packet dropping in MANETs, this paper proposes an intrusion detection system (IDS) which is based on a cross-layer approach. For that, statistics from the network and medium access control (MAC) layers are collected and analyzed. An analytical model including collisions and channel errors is used, whereas a heuristic is applied in order to distinguish between mobility-related situations and real attacks. The results obtained in experimentation show two main conclusions. First, the detection rate is maximized while the false positive rate is minimized, both in static and in mobile scenarios. Second, our approach overcomes the computational overhead problem usually present in most of current detection schemes.

The rest of the paper is organized as follows. Section II provides some related work regarding packet dropping in MANETs. A necessary (short) background to understand our proposal is explained in Section III, after which the particular cross-layer IDS approach for detecting this kind of malicious behavior is introduced in Section IV. Section V describes the experimental environment to test the proposal, as well as the detection results obtained. Finally, main conclusions and future work are presented in Section VI.

II. RELATED WORK

A big number of intrusion detection systems have been proposed in the literature for dealing with packet dropping in mobile ad hoc networks [2].

In their pioneering work in MANETs, Marti et al. [3] presented Watchdog. Here, a monitor node *mn* compares the recently sent packets by it with the overheard packets forwarded by the next hop *nh*. If a sent packet does not match longer than a timeout, a failure tally is incremented for the node

978-0-7695-4745-9/12 $26.00 © 2012 IEEE
DOI 10.1109/TrustCom.2012.75

nh. If the tally exceeds a threshold, *nh* is determined to be a malicious node.

In [4], Zhang et al. introduce a local and cooperative scheme in which each mobile node runs a SVM-based IDS, collects data locally and performs its own detection. If an evidence needs further investigation, a cooperative and global detection procedure is carried out.

A cross-feature method is described in [5], where a data mining analysis is performed to extract correlations between features. Then, a classifier like C4.5, RIPPER or Naïve-Bayes is used to carry out the detection procedure.

The authors in [6] introduce a multi-layer approach composed of three different subsystems that uses a Bayesian classifier, Markov chains and an association rule algorithm for intrusion detection in MAC, routing and application layer respectively. The results from the three layers are integrated in a local module and the final result is sent to a global module.

Kurosawa et al. [7] deal with black hole attacks in MANETs by using the destination sequence number and the number of control packets sent and received to detect deviations from the normal network state. This state is dynamically updated to improve the detection accuracy.

CRADS [8] combines the use of a nonlinear SVM-based detector and some data reduction techniques to decrease the size of the feature set, thus minimizing the learning overhead. In a similar line, the authors in [9] use a linear classification algorithm, namely Fisher Discriminant Analysis (FDA), to remove data with low-information content, making the SVM classifier feasible in ad hoc nodes.

The previous works take into account that, in mobile ad hoc environments, the detection of packet dropping is hindered by the mobility of the nodes out of the communication range, which can cause the IEEE 802.11 RTS/CTS (Request to Send/Clear to Send) mechanism to fail, thus leading to packet drops. Other legitimate reasons which may generate packets drops are:

- Collisions, produced by several contending nodes trying to access the shared medium at the same time.
- Corruption of the packet, due to signal losses, interferences or a high bit error rate (BER).

Actually, these reasons constitute a major concern, mainly because they can cause a large number of false positives if not properly treated by the detection system. This way, recognizing the real cause for a packet dropping is still an open challenge to be addressed when referring to MANET networks.

One of the few works dealing with these circumstances is proposed in [10]. Based on a theoretical model for the different causes of packet loss, the authors detect dropping attacks in DSR-based networks and distinguish these attacks from other legitimate circumstances. However, a very limited topology is studied there, and no mobility aspects are considered. This needs more investigation indeed.

This is the main objective of the present work, where a more complete model is considered to achieve much better detection efficiency in mobility scenarios.

III. BACKGROUND IN PACKET DROPPING

As mentioned above, the forwarding process of a node is analytically modeled in [10], including how collisions and channel errors may affect the behavior of the system. This approach is taken as a starting point for the work presented in this paper, as explained in the following.

Under normal conditions, packets received by an intermediate node will be relayed to the next hop. This operation implies several steps, which are shown in the flowchart depicted in Fig. 1.

Node A wants to transmit a packet to node B. To do this, A waits until the medium is free, requesting it by means of an RTS message (according to a transmission probability P_{Tx}). The message might, with probability P_{COL}, suffer from a collision if another node within the range of A sends an RTS at the same time. If there is no collision, node B replies with a CTS message, which can also collide with a probability P_{COL} if a hidden node, located within the range of B but out of range of node A, transmits some message at the same time. However, a CTS collision only happens if there is no previous RTS collision and, therefore, being the actual CTS collision probability $(1 - P_{COL})\cdot P_{COL}$.

Once node A has accessed the medium, i.e. neither RTS nor CTS collision has occurred, it transmits the desired data to B, which will receive the packet unless a channel error happens. This occurs with probability P_{ERR}. Thus, B will receive the packet correctly only if there was no RTS collision, no CTS collision nor channel error, i.e.

$$P_{RECV} = (1 - P_{COL})\cdot[1 - (1 - P_{COL})\cdot P_{COL}]\cdot(1 - P_{ERR}) \quad (1)$$

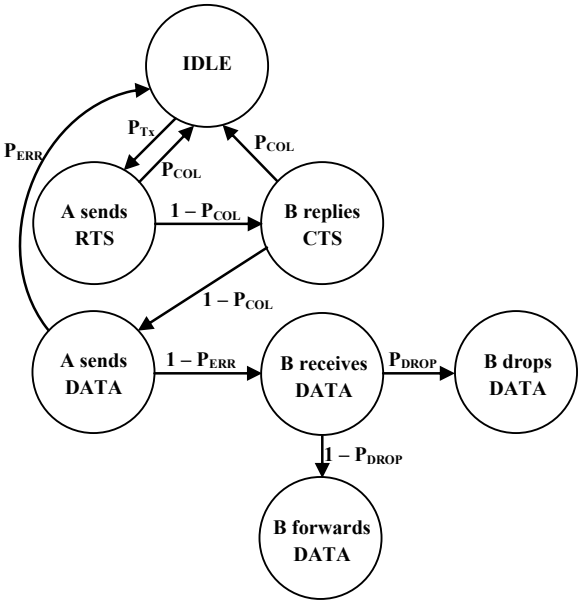

Figure 1. Flowchart for the forwarding process in MANETs.

Finally, when the packet is received at B, it may be either dropped or forwarded by the node. As the forwarding process implies the packet not to be dropped, the event can be computed as:

$$P_{FWD} = P_{RECV} \cdot (1 - P_{DROP}) \quad (2)$$

In summary, the probability of malicious packet dropping behavior can be approximated through (2) as:

$$P_{DROP} = 1 - P_{FWD}/P_{RECV} \quad (3)$$

This theoretical approach takes into account different legitimate causes by which a packet may be not relayed: either because it is not received (due to collisions or errors), or because it is dropped. The manner in which the different aforementioned probabilities are estimated is presented below.

An empirical approximation is used to calculate collisions in RTS and CTS for P_{RECV} in (1). Since this effect is related to the traffic load, we will take into account the number of unanswered RTS packets sent to a node in a certain time window and the total number of attempts to reserve the channel. This way, if a given RTS packet is not replied, the most probable cause is the occurrence of a collision. In summary, the related collision probability part in (1) is computed as:

$$P'_{COL} = (1 - P_{COL}) \cdot [1 - (1 - P_{COL}) \cdot P_{COL}] = \#RTS_U / \#RTS_T \quad (4)$$

where $\#RTS_U$ and $\#RTS_T$ are the number of unanswered RTS messages and total sent RTS messages, respectively.

The channel error probability has been selected based on the experimental results obtained in [11]. In this work, the authors perform a thorough investigation to model the probability of error in wireless links under several conditions.

Finally, P_{FWD} is obtained as the percentage of data packets forwarded by a given node with regard to those received by it. For that, we monitor the received data packets whose destination is not the overheard node, as well as the packets sent by the node when it is not the source of the communication. The estimated value for P_{FWD} is then:

$$P_{FWD} = \#DATA_{FORWARDED} / \#DATA_{RECEIVED} \quad (5)$$

Once P_{DROP} has been obtained from (3), it is compared with a predefined detection threshold value *Thr*. If the probability value is greater than this threshold the monitored node is concluded to be malicious, and legitimate otherwise:

$$\text{class(node)} = \begin{cases} \text{malicious}, & \text{if } P_{DROP} \geq \text{Thr} \\ \text{legitimate}, & \text{otherwise} \end{cases} \quad (6)$$

The main drawback of this theoretical model is that it does not consider that the nodes can be under mobility conditions which, as indicated in Section II, are a relevant cause for the RTS/CTS mechanism to fail. This way, the detection process may result in high false positive rates. This major limitation is going to be addressed in our proposal in the following section.

IV. A Novel Scheme for Efficient Packet Dropping Detection in Mobility Scenarios

The packet dropping IDS proposed here deals with environments with mobility. That is, malicious dropping actions must be differentiated from others which are really legitimate due to the movement of the nodes. For this purpose, a heuristic which employs basic features from network and MAC layers is introduced.

For a better understanding of the heuristic it is necessary to give some brief tips about how the routing and MAC protocols work and interact. In this work, the AODV (Ad hoc On-Demand Distance Vector) routing protocol [12] is considered, although the proposed detection methodology can be easily extended to other similar protocols, like DSR.

AODV is a reactive protocol, i.e. routes to a given destination are established on demand. If a node needs a connection, it broadcasts a route request message (RREQ) that would be forwarded by other nodes. When a node receiving such a message has a route to the destination, it sends a route replay message (RREP) backwards. This whole process is known as *route discovery*.

In order to work properly, each node keeps track of the nodes it can communicate directly, considered as its neighbors, by listening for HELLO messages periodically broadcasted by each node. To avoid unnecessary bandwidth and energy consumption due to these messages, it is common in MANETs to use a link layer-based procedure to update the list of neighbors. When a node starts sensing the medium and sending RTS messages for relaying a packet, the procedure checks if the 802.11 RTS/CTS mechanism reaches the maximum number of retransmissions, i.e. the maximum number of RTS messages without a CTS reply. This value for RTS_{max} is set to 7 by default in the protocol. In such a case, AODV considers that the link is broken and initiates a mechanism called *route maintenance*. Once the procedure starts, two possibilities may occur (Fig. 2):

- *Scenario 1*: If the broken link is closer to the source node than to the destination, the intermediate node brings down the route and sends immediately a RERR message backwards to alert its precursors about the link fail. Then, the precursors stop sending packets to the intermediate node and recursively retransmit the RERR messages.
- *Scenario 2*: If the link is closer to the destination, the intermediate node tries to perform a local repair of the route, by sending a RREQ message like the source would do. After a certain time, if the route cannot be repaired, the node will send a route error message RERR to its precursors.

Note that, during a certain time, the node with a broken link (intermediate node) will continue receiving messages which are unable to be forwarded. That is, the node behaves in a similar way that a malicious node does. This period of time will be considerably longer in Scenario 2, since the route maintenance can take up to dozens of seconds before the RERR message can be sent.

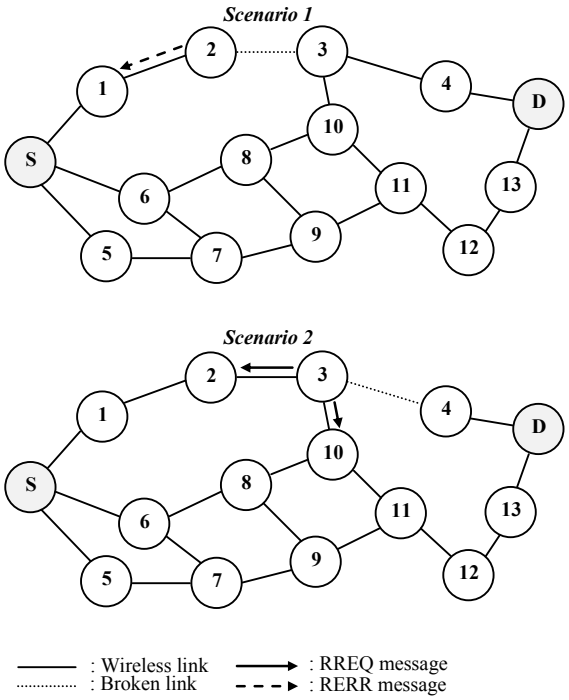

Figure 2. Scenarios which may occur when RTS/CTS mechanism fails.

Therefore, the following main features are involved in determining if a given node *i* in the network is dropping packets due to mobility reasons or not:

- $\#RTS_i$: the total number of RTS messages sent by the node *i* to any other node in the neighborhood.
- $\#CTS_i$: the total number of CTS messages replied by the neighbor nodes towards the node *i*.
- $RREQ_i$: this is a boolean feature that takes a true value if any RREQ message has been broadcasted by the node *i*, and false otherwise.

Note that RERR messages are not used as a feature, although they might seem to be useful for distinguishing the previously described Scenario 1 and Scenario 2. The reason for this is that the format of the RERR packets does not include the originator node. Therefore, it is not easily possible to identify the node that detects the broken link, which will cause false negatives if a malicious node forwards a RERR message and is wrongly classified as legitimate.

Taken into account the above, we finally propose to reduce the number of alarms generated in detection by deriving the probability of a real attack P_{ATTACK} from P_{DROP} through the heuristic presented below. It is based on the three previously indicated features (#RTS, #CTS and RREQ) and follows a time basis procedure. That is, each feature is obtained for non-overlapping time windows *w* of T seconds of duration for each node *i* in the network, so that the decision of a given node being malicious or not is windowed over time.

***Heuristic*:** It can be expressed by words as follows:

- If $(\#RTS_i(w)–\#CTS_i(w))$ exceeds a given value, named RTS_{LIMIT}, we assume that Scenario 1 has occurred. Therefore, the node will be reliable during this single window *w*, because the packet dropping is due to the broken link rather than malicious reasons.
- Besides, if $(\#RTS_i(w)–\#CTS_i(w)) > RTS_{LIMIT}$ and $RREQ_i(w) = TRUE$, we assume that the route is being locally repaired (Scenario 2) and the node will be treated as reliable during the following *N* windows. The election of the value for *N* will be justified in Section V.

Both conditions are mathematically expressed in the following equations:

$$cond_{1,i}(w)=\begin{cases}1, & \text{if } [\#RTS_i(w) - \#CTS_i(w)] > RTS_{LIMIT}\\ 0, & \text{otherwise}\end{cases} \quad (7)$$

$$cond_{2,i}(w)=\begin{cases}1, & \text{if } \sum_{j=0}^{N}[RREQ_i(w-j)cond_{1,i}(w-j)] > 0\\ 0, & \text{otherwise}\end{cases} \quad (8)$$

If none of the previous conditions is satisfied, P_{ATTACK} takes the value of P_{DROP}, and if it exceeds *Thr*, the IDS considers the monitored node as malicious and some response mechanism/s should be triggered.

The detailed description of the final detection algorithm is shown in Fig. 3, which can be reduced to the following expression:

$$class(i,w) = \begin{cases}\text{malicious}, & \text{if } P_{ATTACK,i}(w) \geq Thr\\ \text{legitimate}, & \text{otherwise}\end{cases} \quad (9)$$

where

$$P_{ATTACK,i}(w) = \begin{cases}0, & \text{if } cond_{1,i}(w) \text{ OR } cond_{2,i}(w)\\ P_{DROP,i}(w), & \text{otherwise}\end{cases} \quad (10)$$

Detection Algorithm

```
1  For w=1 to the number of windows in the
   monitoring time:
2    For i=1 to the number of nodes in the
     network:
3      Obtain #RTSi(w), #CTSi(w) and RREQi(w)
4      Estimate P'COL with the approximation (4).
5      Calculate PFWD using (5).
6      Get PDROP with (1) and (3).
7      Apply heuristic to compute PATTACK.
8      Compare PATTACK with Thr to determine if the
       monitored node i is malicious in the
       window w.
9    End for
10 End for
```

Figure 3. Pseudo code for the packet dropping detection algorithm.

V. PERFORMANCE ANALYSIS

This section presents first a description of the experimental environment used to evaluate the packet dropping IDS proposed here. As several tests have been made to verify the assumed hypothesis and the proper performance of the approach, after that, the experimental results obtained are discussed.

A. Experimental environment

In this research, the popular tool Network Simulator 2 (NS-2) [13] is used to simulate several deployments of a MANET. Its choice is justified because, nowadays, it is one of the most used simulators by the academic and research community. The simulation area is restricted to a 1000m x 1000m square, with each node having a communication range of 250m. AODV is chosen as the routing protocol, and 802.11b is used as the MAC layer protocol. Other in-depth simulation parameters are shown in Table I and Table II, where default values have been selected.

TABLE I. CONFIGURATION PARAMETERS IN NS-2

Parameter	Value	Parameter	Value
Radio Model	*TwoRayGround*	MAC Type	*802_11*
Channel	*WirelessChannel*	├CW$_{min/max}$	31/1023 *slots*
Antenna	*OmniAntenna*	├Slot Time	20 µs
├Tx/Rx Gain	1	├SIFS	10 µs
└High	1.5 m	├Data Rate	11 Mb
Network Interface	*WirelessPhy*	├Basic Rate	2 Mb
├Capture Thresh	10 dB	├PLCP Rate	1 Mb
├Carrier Thresh	$1.5e^{-11}$ W ≈ 550 m	├SSRC	7
├Rx Thresh	$3.6e^{-10}$ W ≈ 250 m	├SSLC	4
├Tx Power	0.2818 W ≈ 250 m	└ RTS Thresh	0 bytes
├Frequency	914 MHz	Queue Type	*PriQueue*
└Loss Factor	1	└Size	50

TABLE II. AODV PARAMETERS IN NS-2

Parameter	Value	Parameter	Value
Active Route Timeout	10 s	#RREQ Retries	3
Reverse Route Life	6 s	RREP Wait Time	1 s
Max. RREQ Timeout	10 s	Link Layer Detection	yes

The total number of nodes is 25. On the other hand, the number of application traffic flows is fixed to 20, each flow consisting of a Constant Bit Rate (CBR) connection, with 4 packets/second data and payload size equal to 512 bytes.

To model the movement of the nodes the Random Waypoint Model is used, with a fixed minimum speed of 1 meter/second and a maximum speed varying from 5 to 20 meters/second. The pause time is set to 15 seconds, i.e. once the node reaches the desired destination, it waits for the pause time before choosing a new random destination and repeating the process.

The malicious nodes are configured to drop 20% of the data packets going through them and supposed to be relayed. However, they participate normally in the routing process, without modifying or discarding any control packet. Thus, they can be considered as gray holes that do not try to maliciously include themselves in the path. We have proposed a model of attack in which the malicious node acts individually. This paper does not deal with a model of attack where several nodes collude to evade the detection process.

The duration of the selected time window for collection of features is 5 seconds. The upper bound for the time that can take the local repairing process depends on some parameters of AODV, including certain randomness caused by a binary exponential backoff mechanism used to avoid congestion. From all of this, the mentioned bound is close to 60 seconds, and therefore, N is selected to be equal to 12 –see (8)–.

The effectiveness of the proposed IDS is evaluated by computing two parameters, namely the true positive rate (TPR) and the false positive rate (FPR). As known, a true positive is the correct classification of a malicious node, whereas a false positive is the incorrect classification of a legitimate node. In this line, we obtain various operation points in the ROC (Relative Operation Characteristic) space by varying the decision threshold. The ROC curve shown in Fig. 4 has been obtained by repeating 75 times (with different seeds) every simulation. The maximum speed is fixed to 10 meters/second.

As can be seen in the curve, if the detection threshold is increased, the system is expected to improve the false positive rate, but to make worse the detection accuracy. On the other hand, a lower threshold will result in a better detection rate, but in an increase in the false positive rate.

Thus, the optimal operation point of our system can be achieved empirically. For best performance, the parameter *Thr* is fixed to 0.15, which seems to provide a good tradeoff between true positives and false positives, taken into account the proposed model for malicious nodes.

Although the value of RTS_{max} is set to 7 by default in the protocol, a lower value has been chosen for RTS_{LIMIT} to reduce the number of RTS messages not taken into account due to temporal windowing, this is, the number of RTS messages that can be found at the beginning of a given window, but should belong to the previous one. As shown in Table III, the false positive rate varies according to the value of RTS_{LIMIT}. The parameter is set to 4 since, given this value that produces the lower FPR, it is very likely that one of the abovementioned scenarios may be taking place.

TABLE III. DETECTION RESULTS FOR DIFFERENT RTS_{LIMIT} VALUES

RTS_{LIMIT}	*TPR (%)*	*FPR (%)*
4	100	2.29
5	100	2.35
6	100	2.50
7	100	12.00

B. Verifying the heuristic

Some first tests are intended to validate the heuristic presented in Section IV. As it was previously cited, a node under mobility conditions can discard packets, acting as a malicious node.

Fig. 5 shows the value of P_{DROP} (in percentage) together with the values of the features ($\#RTS_i(w)$–$\#CTS_i(w)$) and $RREQ_i(w)$ over the time, both for a malicious node and for a legitimate one.

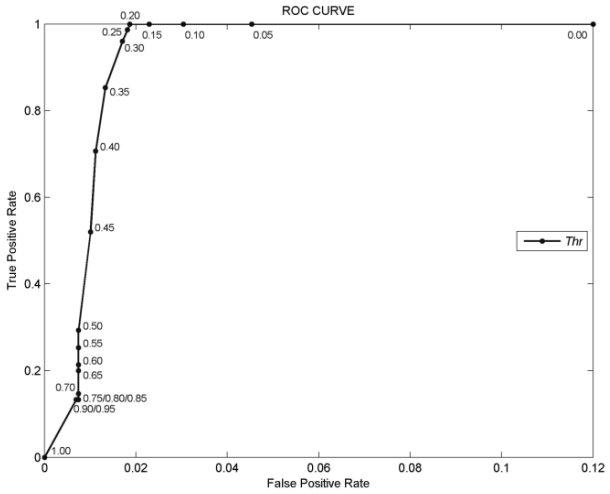

Figure 4. ROC curve by varying the *Thr* parameter.

We can see that both nodes would result in positive detection, even when one of them is not malicious at all. It is possible to check that, in the non-malicious case (Fig. 5b), there are two different peaks, related to the two situations that can appear.

The first one, located at 270 seconds, is caused by the Scenario 1, i.e. when the value of ($\#RTS_i(w)$–$\#CTS_i(w)$) exceeds RTS_{LIMIT} and none RREQ message is sent. Thus, the node transmits an RERR message but drops packets until its precursor receives the RERR and stops sending it data.

The second peak, starting at 435 seconds, is due to a local route repair (Scenario 2). The number of unanswered RTS messages is higher than the fixed limit, while the node sends RREQ messages in order to get a new available route. This process can take a long time if no other route is found, the P_{DROP} value of the node being high during this entire period.

In some cases, the features of the malicious node can be similar to those of the legitimate one. As shown in Fig. 5a, ($\#RTS_i(w)$–$\#CTS_i(w)$) exceeds RTS_{LIMIT} at 330 seconds. This is due to the own movement of the malicious node, which can also be found under the studied scenarios. Therefore, it will also be temporarily considered as legitimate.

As demonstrated in this subsection, it can be concluded that the initial hypothesis about the possible existence of errors (both false positives and false negatives) in detecting malicious packet dropping in mobile environments is verified. In fact, this constitutes the necessary support for our proposed approach.

C. Detection Results

Two different sets of tests have been finally performed in order to evaluate the correct performance of our IDS approach in different environments. It must be said that every simulation was repeated 75 times, by varying the seed and thus obtaining a different scenario in each run. Ninety five percent confidence intervals are used. Besides, the experimental results have been compared with those obtained by the "basic" scheme proposed in [10], in order to show how our system overcomes the performance of the previous.

The first set of tests deals with the study of detection efficiency for different mobility conditions. Three mobility scenarios are simulated, which include 5 m/s, 10 m/s and 20 m/s. Besides, a zero mobility scenario is evaluated in order to check if the proposed approach does not considerably degrade the performance of the basic model. Table IV shows both TPR and FPR for the different conditions.

TABLE IV. DETECTION RESULTS FOR DIFFERENT MOBILITY SCENARIOS

Mobility	Our Proposal		Basic Model [10]	
	TPR (%)	*FPR (%)*	*TPR (%)*	*FPR (%)*
0 m/s	93.75 ± 5.57	0.0 ± 0.0	100 ± 0.0	0.63 ± 0.41
5 m/s	100 ± 0.0	2.30 ± 0.70	100 ± 0.0	36.59 ± 2.24
10 m/s	100 ± 0.0	2.30 ± 0.72	100 ± 0.0	48.22 ± 2.85
20 m/s	100 ± 0.0	2.19 ± 0.70	100 ± 0.0	62.56 ± 2.39

As expected, our IDS outperforms in every mobility

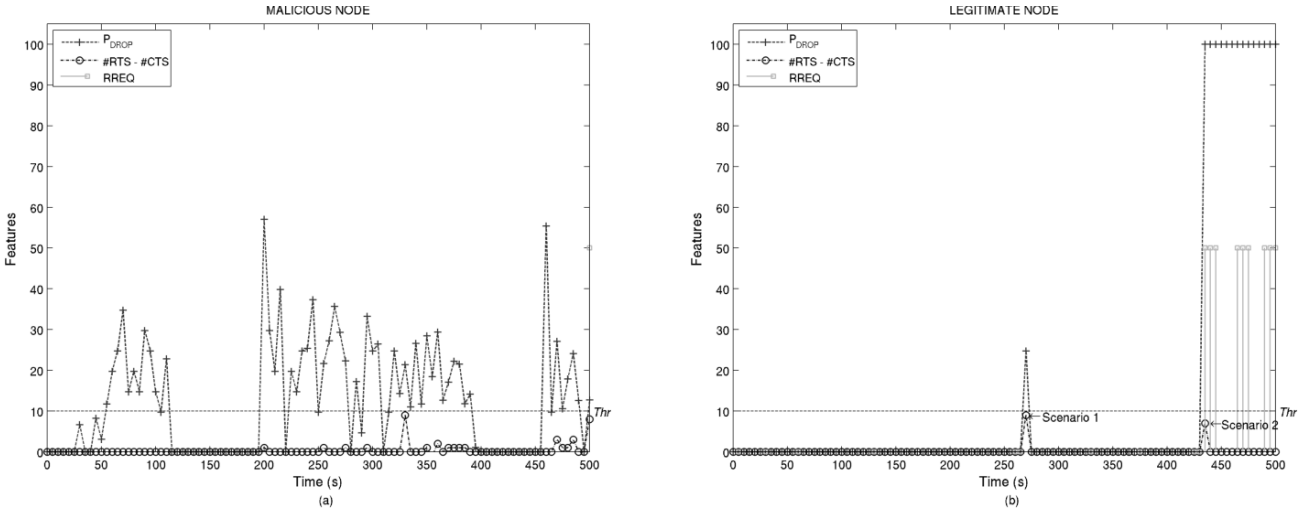

Figure 5. Probability of dropping (P_{DROP}) and features of interest –($\#RTS_i(w)$ – $\#CTS_i(w)$) and $RREQ_i(w)$– both for i = malicious node (a) and for i = legitimate node (b).

scenarios the basic model, especially with regard to the false positive rate, which results improved in a factor upper to 90%. It should be noted however that the TPR value is slightly reduced in the static scenario case. This is mainly due to static scenarios where many collisions appear, which leads the heuristic to consider that mobility is being detected. However, the FPR value is still enhanced by almost 40%.

A second set of experiments tries to examine the performance of both detection approaches (ours and basic) for an increasing number of malicious nodes, in order to prove that the performance of our proposal is not degraded although several nodes in the network are compromised. The results can be seen in Table V.

TABLE V. RESULTS FOR DIFFERENT NUMBER OF MALICIOUS NODES

#Malicious Nodes	Our Proposal		Basic Model [10]	
	TPR (%)	*FPR (%)*	*TPR (%)*	*FPR (%)*
1	100 ± 0	2.30 ± 0.70	100 ± 0	36.59 ± 2.24
2	100 ± 0	1.92 ± 0.67	100 ± 0	35.42 ± 2.28
5	100 ± 0	2.67 ± 0.82	100 ± 0	29.44 ± 1.97
10	98.94 ± 0.70	0.96 ± 0.42	100 ± 0	21.39 ± 1.80

They reveal that, even if a bigger number of malicious nodes exist in the network, the proposed scheme remains accurate in detection, keeping the false positive rate below 3%. That is, like in the previous experimentation, our proposal far overcomes the one in [10] in terms of false positives.

In summary, it is evident from the results obtained that the proposed packet dropping IDS approach can efficiently detect all the malicious nodes with an overall accuracy upper to 99%. Moreover, and not less important, the system gets a very low false positive rate, less than 3% in any case.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an intrusion detection system for detecting malicious packet dropping in mobile ad hoc networks, by collecting features from the MAC and network layers. The cross-layer approach uses a heuristic to detect packet dropping attacks under several circumstances which are not usually taken into account in previous works and which can cause a high number of false positives in detection. It should be noted that the use of a simple heuristic overcomes the computational overhead present in more sophisticated approaches based on data mining algorithms found in the literature.

We have verified by means of simulation the initial hypothesis, several scenarios having been analyzed. The results obtained clearly highlight the excellent performance of our IDS approach, which experienced 99% overall detection rate with less than 2% of false positives rate. This far overcomes the results exhibited by other similar schemes in the literature.

It must be said that the operation point of our system depends on the desired needs of the network administrator. If detecting all possible malicious nodes in the network is the main objective, a lower threshold can be selected, at the expense of increasing the rate of false positives. On the other hand, to use a higher threshold value can be useful to detect higher malicious packet drop rates but produce low, if any, false positives.

As shown, experimental results obtained are very encouraging. However, this first study has some limitations which are projected to be taken into consideration in order to improve the system. Such issues planned to be modified in the future are:

- The incorporation of other relevant features to our IDS (e.g. route changes) which provide a more accurate and detailed information. Their inclusion will presumably enhance the system capabilities.
- Moreover, accepting the fact that an isolated approach is not probably the best solution for MANETs, we are moving to other alternative architectures (e.g. distributed), which seems to be more suitable for these networks.
- Finally, the design from scratch of a whole theoretical model that properly includes any possible situation is part of our ongoing work.

ACKNOWLEDGMENT

This work has been partially supported by Spanish MICINN (*Ministerio de Ciencia e Innovación*) through project TEC2011-22579.

REFERENCES

[1] Y. Huang and W. Lee, "Attack analysis and detection for ad hoc routing protocols", in Proc. Recent Advances in Intrusion Detection 2004 (RAID '04), pp. 125–145, France, September 2004.

[2] S. Djahel, F. Naït-Abdesselam and Z. Zhang, "Mitigating packet dropping problem in mobile ad hoc networks: proposals and challenges", IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 658–672, Fourth Quarter 2011.

[3] S. Marti, T. J. Giuli, K. Lai and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", in Proc. 6th Annual Intl. Conf. on Mobile Computing and Networking (MOBICOM '00), pp. 255–265, USA, August 2000.

[4] Y. Zhang, W. Lee and Y.-A. Huang, "Intrusion detection techniques for mobile wireless networks", Wireless Networks, vol.9, No. 5, p.p. 545–556, September 2003.

[5] Y. Huang, W. Fan, W. Lee and P.S. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies", in Proc. 23rd IEEE Intl. Conf. on Distributed Computing Systems (ICDCS '03), pp. 478–487, USA, May 2003.

[6] S. Bose, S. Bharathimurugan and A. Kannan, "Multi-layer integrated anomaly intrusion detection system for mobile adhoc networks", IEEE Intl. Conf. on Signal Processing and Networking 2007 (ICSCN '07), pp. 360–365, India, February 2007.

[7] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour and Y. Nemoto, "Detecting blackhole attack on AODV-based mobile ad hoc networks by dynamic learning method", Intl. Journal of Network Security, vol. 5, no. 3, pp. 338–346, November 2007.

[8] J.F.C. Joseph, A. Das, B.-C. Seet and B.-S. Lee, "CRADS: integrated cross layer approach for detecting routing attacks in MANETs", IEEE Wireless Communicactions and Networking Conf. 2008 (WCNC '08), pp. 1525–1530, April 2008.

[9] J.F.C. Joseph, B.-S. Lee, A. Das and B.-C. Seet, "Cross-layer detection of sinking behavior in wireless ad hoc networks using SVM and FDA", IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 2, pp. 233–245, April 2011.

[10] T. Hayajneh, P. Krishnamurthy, D. Tipper and K. Taehoon, "Detecting malicious packet dropping in the presence of collisions and channel errors in wireles ad hoc networks", IEEE Intl. Conf. on Communicactions 2009 (ICC '09), pp. 1–6, Germany, June 2009.

[11] J. Arauz and P. Krishnamurthy, "Markov modeling of 802.11 channels", IEEE 58th Vehicular Technology Conf. 2003 (VTC Fall '03), vol. 2, pp. 771–775, USA, October 2003.

[12] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) routing", IETF RFC 3561, July 2003.

[13] Ns Network Simulator. http://www.isi.edu/nsnam/ns/