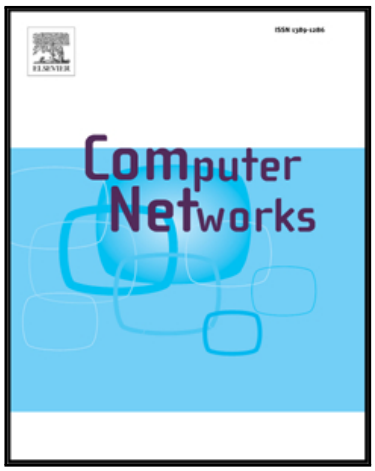


A Model of Data Forwarding in MANETs for Lightweight Detection of Malicious Packet Dropping

Leovigildo Sánchez-Casado, Gabriel Maciá-Fernández,
Pedro García-Teodoro, Roberto Magán-Carrión

PII: S1389-1286(15)00172-3
DOI: 10.1016/j.comnet.2015.05.012
Reference: COMPNW 5575

To appear in: *Computer Networks*

Received date: 10 October 2014
Revised date: 15 May 2015
Accepted date: 22 May 2015

Please cite this article as: Leovigildo Sánchez-Casado, Gabriel Maciá-Fernández, Pedro García-Teodoro, Roberto Magán-Carrión, A Model of Data Forwarding in MANETs for Lightweight Detection of Malicious Packet Dropping, *Computer Networks* (2015), doi: 10.1016/j.comnet.2015.05.012

A Model of Data Forwarding in MANETs for Lightweight Detection of Malicious Packet Dropping

Leovigildo Sánchez-Casado*, Gabriel Maciá-Fernández, Pedro García-Teodoro, Roberto Magán-Carrión

Department of Signal Theory, Telematics and Communications
School of Computer Science and Telecommunications
CITIC-UGR, University of Granada
C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain

Abstract

This work introduces a model of data forwarding in MANETs which is used for recognizing malicious packet dropping behaviors. First, different legitimate packet discard situations are modeled, such as those generated by collisions, channel errors or mobility related droppings. Second, we propose an anomaly-based IDS system based on an enhanced windowing method to carry out the collection and analysis of selected cross-layer features. Third, a real deployment of the IDS is also considered by suggesting a methodology for the collection of the selected features in a distributed manner. We evaluate our proposal in a simulation framework and the experimental results show a considerable enhancement in detection results when compared with other approaches in the literature. For instance, our scheme shows a 22% improvement in terms of true positives rate and a remarkable 83% improvement in terms of false positives rate when compared to previous well-known statistical solutions. Finally, it is notable the simplicity and lightweightness of the proposal.

Keywords: IDS, Malicious node, MANET, Packet dropping attacks

1. Introduction

Wireless networks have considerably evolved in the last years, leading to the appearance of different related technologies, architectures and applications [1]. One of the architectures that have attracted much attention, especially by the research community, are the so called Mobile Ad hoc NETworks (MANETs).

A MANET is a kind of network composed of mobile devices distributed in a geographic area where there is a lack of fixed infrastructure or centralized administration.

*Corresponding author. Tel.:+34-958241717
Email addresses: sancale@ugr.es (Leovigildo Sánchez-Casado), gmacia@ugr.es (Gabriel Maciá-Fernández), pgteodor@ugr.es (Pedro García-Teodoro), rmagan@ugr.es (Roberto Magán-Carrión)

Nodes within a communication range communicate directly, while those out of the range make use of other nodes to forward the messages to a given destination. These features make this kind of networks attractive for their application to areas like environmental monitoring, military applications, disaster management, etc. [2].

As MANETs proliferate, many other engineering and research problems appear. First, security issues associated with this communication paradigm are becoming more and more relevant and, thus, they need to be conveniently addressed. In addition, there is a number of specific factors that must be taken into account in the design or implementation of security systems related to MANETs. These peculiarities usually refer to the constrained nature of the nodes, in terms of their reduced bandwidth, short lifetime of the battery, power-constrained processing, etc. Due to the inherent complexity of MANETs, most of the off-the-shelf techniques and procedures developed for wired networks and even for Wireless Local Area Networks (WLANs) are not suitable for mobile ad hoc networks [3].

Among other threats [4], packet dropping attacks have remarkable consequences in MANETs. Malicious nodes drop received data or control messages instead of relaying them, thus affecting the traffic in the network [5]. There are different types of dropping attacks, depending on the particular strategy adopted by the attacker [6]. *Blackhole* attacks imply malicious nodes dropping all the packets they receive. *Grayhole* attacks are similar but, here, malicious nodes drop packets statistically, following a predetermined probability distribution.

The motivations for striking such attacks are different. First, malicious nodes might seek to save energy resources (*selfish behavior*). Second, an attacker might be trying to affect to the performance of a single node, or even disrupt the whole network operation. Nevertheless, regardless of the particular motivation for carrying out the attack, the final behavior exhibited by the dropper node and thus its effects is the same. Consequently, the developed scheme will be able to detect the attack anyway.

This paper proposes a cross-layer anomaly-based Intrusion Detection System (IDS), where features from Medium Access Layer (MAC) and network layers are considered. First, we suggest a model of normality for the forwarding process of nodes in MANETs. This model considers the existence of events that might cause legitimate drops, such as collisions, channel errors or mobility related situations. Modeling these events, we are able to distinguish between that circumstances and actual malicious dropping actions, *i.e.*, anomalies in the expected behavior.

Based on the proposed model, we develop an IDS system for the detection of dropping attacks. The main benefits are: *(a)* the promising results exhibited prove the detection capabilities of the developed approach; *(b)* the computational overhead is reduced as a consequence of using a simple analytical model, so that the IDS is lightweight and can be used in constrained environments; *(c)* the IDS is efficient from an energy point of view, since it does not waste resources in nodes with scarce activity due to the enhanced windowing method proposed.

The organization of the rest of the paper is as follows. Some related work in the field is reviewed in Section 2. The analytical model of the forwarding process which is used as the basis for our detection proposal is presented in Section 3. The IDS system for the detection of dropping attacks is explained in Section 4. After that, we discuss how to implement the system in a real network in Section 5. Section 6 describes the

experimental results obtained. Finally, we draw our conclusions in Section 7.

2. Related Work

A wide variety of solutions that handle packet dropping attacks in MANETS can be found in recent research [7, 8]. Below, we discuss and classify them in three categories according to their basic operation, ACK-, reputation- and detection-based schemes. We have deliberately omitted some other solutions, such as credit-based schemes, as we consider them as prevention mechanisms.

2.1. ACK-based Schemes

Here, nodes in the network communicate with their neighbors to explicitly request acknowledgments and confirm the reception of sent packets.

A two-hop ACK-based scheme is proposed in [9]. It uses authentication to prevent nodes in a single hop from forging ACK packets on behalf of the two-hop neighbors. For reducing the communication overhead, they propose to ask the two-hop neighbors randomly [10]. These two schemes fail when any two-hop neighbors do not cooperate.

TWOACK [11] detects malicious links by sending two-hop acknowledgment packets in the opposite direction of the routing path, each sender having a list of not acknowledged packets and a counter of the forwarded and missed data packets. Besides, to reduce the routing overhead, authors present in [12] an improvement of their scheme called 2ACK, where only a portion of the packets are acknowledged.

Djahel *et al.* [13] investigate how to mitigate the loss of topological information due to the dropping of Topology Control (TC) messages in the OLSR (Optimized Link State Routing) protocol. They propose a three-hop acknowledgment-based scheme, which adds two extra control packets. Nodes use *3hop_ACK* packets to acknowledge TC messages, while *HELLO_rep* packets advertise two-hop neighbors to a requesting MultiPoint Relay (MPR) node. If the number of missed TC / *3hop_ACK* packets surpasses a given limit, the MPR node is considered malicious.

The authors in [14] complete their previous works in [9] and [10] by employing two-hop cryptographic acknowledgments for unicast packets, and a passive feedback mechanism for monitoring broadcast packets. The collected information is afterwards used as the basis for an accusation-based collaborative mechanism for node isolation.

In [15], the rank of intermediate nodes is updated whenever an acknowledgment is received in the source. If this rank drops to zero, the node is classified as malicious.

The main problem of these approaches is that, if acknowledgments are lost due to any legitimate reason (as it will be seen in Section 3), this fact can increase the number of false positives provided by the detection scheme.

2.2. Reputation-based Schemes

The basic idea behind these techniques is that each node first generates an opinion with respect to others. Afterwards, all of them collectively detect low reputation nodes.

In [16], the CONFIDANT protocol is designed to revoke malicious nodes. It is composed of four components. A *monitor* supervises, through a passive-feedback technique, the behavior of its first-hop neighbors. Whenever a suspicious event is detected,

details are passed to a *reputation system*, which maintains a table with the ratings for all the nodes. A *trust manager* sends and receives alarm messages, informing about detected adversaries. Finally, a *path manager* is responsible for launching an appropriate response.

The authors in [17] propose Friends and Foes, a scheme to punish selfish nodes. Here, each node maintains *credits* for each other and classifies the rest in three categories periodically updated: friends, *i.e.*, those for which the node will relay packets; foes, when no service is given at all; and selfish, corresponding to those that consider the node as a foe. When a node sends a packet, it searches for a friend as a next hop. Similarly, when it is requested to forward a packet, it only does it for friend requesters.

The concept of *inner-circle consistence* was adopted in [18] to identify forged route replies and prevent packet dropping attacks. The idea is to let each node discover its k-hop neighborhood. All its neighbors form its inner-circle, responsible for filtering malicious outgoing data. Specifically, route replies need to get approval from its inner-circle, which verifies the validity. If a reply contains false routing information, an attack is detected and prevented by a voting process performed by each inner-circle node.

The major drawback of this type of techniques is the excessive traffic required for sharing the reputation information.

2.3. Detection-based Schemes

Marti *et al.* [19] proposed a system called Watchdog, where a monitor node compares the packets that it sends with the overheard packets forwarded by the next hop. If a packet is not localized in a given period, a counter is incremented for the next hop. A counter with a value higher than a given threshold indicates a malicious behavior. When a malicious node is identified in the path towards the destination node, a response mechanism called Pathrater is launched to avoid routing through the misbehaving node.

Zhang *et al.* [20] suggest a scheme in which each node in the network runs an IDS agent based on Support Vector Machines (SVM) that monitors local traces, collecting data like user and system activities or communications within the radio range. Also, each agent is responsible for detecting, locally and independently, signs of intrusions. However, if an anomaly is detected in the local data, or if evidence is inconclusive and needs further investigation, neighboring IDS agents will collaboratively investigate in a broader range.

In [21], a data mining analysis is performed to extract correlations among features. Classifiers like Naïve-Bayes, RIPPER or C4.5 are then used for the detection.

In [22], the authors follow a multi-layer approach combining three different subsystems that use an association rule algorithm, Markov chains and a Bayesian classifier. They perform the intrusion detection in application, routing and MAC layers. The results from all the classifiers are combined locally, and the final result is sent to a global decider.

Kurosawa *et al.* [23] introduce an anomaly detection scheme which uses a dynamic training method. In this scheme, the number of control packets and the average of the differences between the destination sequence numbers in the routes are employed to detect deviations from the normal network state. This state is dynamically updated to

improve the detection accuracy. Other authors also propose dynamic adaptations in their works, like those in [24, 25].

CRADS [26] uses a nonlinear SVM-based detector and some data reduction techniques to decrease the amount of features considered and reduce the learning overhead.

Similarly, in [27], the authors make use of Fisher Discriminant Analysis (FDA) to eliminate low-information content data, thus making the SVM classifier feasible for constrained resources nodes.

The authors in [28] incorporate a Bayesian filter into the standard watchdog implementation in order to reduce the number of false positives. These Bayesian watchdogs share the obtained information to perform a collaborative detection.

2.4. Discussion of Related Work

Many of the aforementioned works are focused on the detection of dropping events in the network, identifying them as malicious or abnormal behaviors. However, they do not fully consider that the detection of packet droppings can be fooled by the existence of other causes, like the mobility of the nodes, which can make the RTS/CTS mechanism to fail when a node moves out of the communication range, thus leading to packet drops. Besides, other legitimate causes might also lead to packets drops: collisions in packets, due to contending nodes trying to access the shared medium at the same time, and packet errors, due to the existence of high Bit Error Rates (BER), interferences or signal losses.

Some approaches based on data mining techniques implicitly consider these mobility situations. However, their drawback is the inherent computational overhead, which makes them not feasible to be implemented in resource-constrained networks.

It should be emphasized that recognizing the actual cause (mobility, collisions, errors, malicious behavior) for a packet dropping in MANETs is still an open challenge, which must necessarily be addressed in order to reduce the number of false positives in IDS schemes.

One of the few works dealing with collisions and packet errors is [29]. First, a theoretical framework models the causes of packet losses. Then, it is applied to DSR-based networks. Regretfully, the authors only study a very limited topology, without taking into account mobility aspects.

For this reason, in our previous work [30] we proposed a heuristic to complete the model in [29], in order to properly deal with scenarios with mobility. Here, some features from MAC and routing layers were considered. As a result of such multi-layer approach, we obtained much better detection efficiency than that obtained in [29].

In the present paper, a new analytical model which natively includes any possible situation in data forwarding for mobile ad hoc environments is presented. Besides including mobility as a legitimate cause for packet discards, our new model differs from the one in [29] in how the dropping probability is computed and the features are collected. On the one hand, the latter involves obtaining features from two nodes: the sender, which sends RTS/DATA packets and receives CTS packets; and the receiver, which receives RTS/DATA packets, sends back CTS packets and is supposed to forward the data. On the other hand, as it will be explained in Section 3, our new model obtains the features from a single node. In addition, we have introduced a novel windowing

scheme in Section 4.4. As a main result of these differences, the set of features of our previous work is reduced, which simplifies, speeds up and improves the detection scheme.

3. Model for the Forwarding Process in MANETs

As a previous step to develop our approach for dropping attack detection, we model the forwarding process in a MANET. The model considers different legitimate circumstances in communications (collisions, channel errors or mobility) as well as malicious behaviors, and allows inferring how they all may affect the performance of the overall retransmission procedure.

The forwarding process in a MANET node implies several steps, detailed in the flowchart depicted in Fig. 1. After a data packet is correctly received by a node, several successive events must necessarily occur for the packet to be forwarded:

1. $\overline{dest}$ event: the considered node is not the final destination of the packet.
2. $rout$ event: the node has a valid route for relaying the packet towards the desired destination.
3. $\overline{drop}$ event: the node is not a malicious dropper and, thus, it would not drop the packets instead of forwarding them.

If all of the previous events occur, the node tries to forward the packet. To do this, two subsequent actions are taken. First, the node will try to send a Request-To-Send (RTS) message (the box *"Sends RTS"* in Fig. 1 is reached). Let us term this event as RTS event, and its associated probability P_{RTS}. To estimateP_{RTS}, the above events 1 to 3 are considered, *i.e.*, there exists a route for the destination and the node is neither the final destination nor a dropper. Thus, P_{RTS} is computed as:

$$P_{RTS} = \Pr\left(RTS \,\middle|\, \overline{dest}, rout\right) = (1 - P_{DROP}) \tag{1}$$

where P_{DROP} is the probability that the packet is maliciously discarded by the node. Note that the event $\overline{drop}$ is modeled as a probability, meanwhile the events $\overline{dest}$ and $rout$ are not. Since these two conditions could be easily determined by the inspection of every received packet in a node, in the calculation of the conditional probability given in (1) we only consider those packets that fulfill the conditions $\overline{dest}$ and $rout$.

Second, the node checks if it receives a Clear-To-Send (CTS) message (the question *"CTS received?"* is reached in Fig. 1). This message is received from the next hop in the route when the corresponding RTS packet has reached its destination and the CTS packet is successfully received. Let us term this as CTS event, and P_{CTS} its associated probability.

Note that RTS and CTS packets, after being sent, can be lost due to several legitimate reasons, *e.g.*, RTS and CTS messages might suffer a collision if another node in the range of the receiver node transmits an RTS at the same time that the first RTS or CTS are sent. In addition, both messages may also be affected by channel errors, which prevent them from reaching their destination. Another scenario where a packet is discarded happens when the nodes are out of the communication range because they

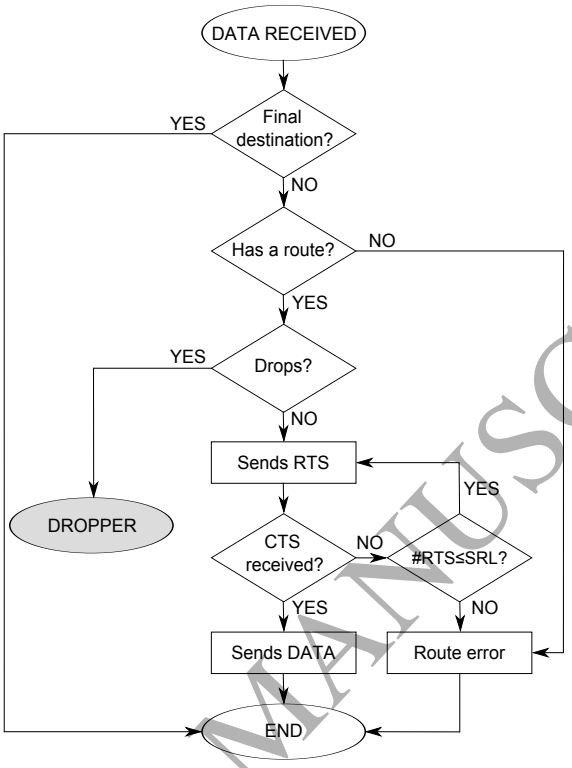


Figure 1: Flowchart for the forwarding process in MANETs.

have moved and they did not have enough time to properly update the routing table. This way, they cannot communicate each other.

All these circumstances cause messages to be lost and CTS packets not to be received, thus leading to an RTS retransmission. The IEEE 802.11 RTS/CTS procedure allows a limit number of attempts to retransmit RTS packets, *i.e.*, if a sender does not receive any CTS reply in response to multiple retransmissions of an RTS packet up to a predefined limit, the sending process fails. This upper value is called Short Retry Limit (SRL), and its default value is 7. Once the SRL is exceeded, the corresponding packet is discarded, and the sender node assumes the link to be broken and the next hop to be no longer accessible.

Therefore, the probability that the CTS message is correctly received at the sender node (CTS event) can be approximated as follows. Our model divides this probability, P_{CTS}, in two terms. The first one is related to collisions or channel errors, taking into account those situations in which RTS retransmissions occur without exceeding the SRL limit. The second term is associated with mobility situations in which the num-

ber of RTS retransmissions is higher than SRL, thus considering the link as broken[1]. Therefore, the CTS packet will be received if none of the two aforementioned situations happens. Thus, the probability that CTS event happens given that RTS event has occurred is:

$$P_{CTS} = \Pr\big(CTS \,\big|\, RTS\big) = 1 - (P_{COL} + P_{MOB}) \tag{2}$$

where P_{COL} is the probability for the RTS or CTS packets to be lost due to collisions or channel errors, and P_{MOB} the probability of packets losses due to broken links caused by mobility circumstances.

Finally, if the sender node captures the medium, it transmits the desired data, *i.e.*, the data packet is forwarded by the node (FWD event). To forward the message, both the events RTS and CTS need to have occurred successfully (see Fig. 1), so the probability for the whole forwarding process, P_{FWD}, is computed as:

$$\begin{aligned} P_{FWD} &= \Pr\big(CTS, RTS \,\big|\, \overline{dest}, rout\big) \\ &= \Pr\big(CTS \,\big|\, RTS\big) \cdot \Pr\big(RTS \,\big|\, \overline{dest}, rout\big) \\ &= (1 - P_{DROP}) \cdot [1 - (P_{COL} + P_{MOB})] \end{aligned} \tag{3}$$

As in the calculation of P_{RTS}, this probability is calculated over all the packets that fulfill the conditions $rout$ and $\overline{dest}$.

Note that, although we have applied this model to the forwarding of data packets, it is also applicable to other kinds of packets in MANETs, like control packets, and to different protocols, either reactive or proactive. The only assumption here is that the associated forwarding process uses the RTS/CTS mechanism, as it will be discussed in Section 4.1. In consequence, the detection approach of packet dropping behaviors developed and presented below is applicable with minor modifications to several other cases than that of data packets.

4. Malicious Packet Dropping Detection

In this section, a new detection methodology for packet dropping in MANETs is explained. First, we describe the attack model and the underlying scenario. Second, we detail the proposed detection approach. Next, we provide details about parameters estimation and suggest a windowing methodology. Finally, a summary of the whole process is presented.

4.1. Attack Model and Scenario Description

We consider the existence of L legitimate nodes geographically distributed in a given area that are moving at a certain speed following a random trajectory. We also

[1] Although there are other reasons to consider a link as broken, like node failures, congestion or others, in this work, for simplicity, we will use indistinctly the terms *mobility* or *broken link* to encompass all these situations. Of course, this aspect does not affect to the fundamentals of our proposal.

assume that IEEE 802.11 is employed as the MAC layer protocol and that the RTS/CTS mechanism is used to send packets, since it is required as part of the 802.11 Distributed Coordination Function (DCF) and used by default. This use is coherent with the mobility of the nodes, as the lack of virtual carrier detection (RTS/CTS mechanism) in such mobility scenarios would imply a lot of collisions due to the well-known hidden station problem. Finally, nodes communicate using an ad hoc routing protocol, and different kinds of traffic flows can be randomly generated by them.

In this general scenario, we additionally consider the existence of M malicious nodes, with the same behavior as the legitimate ones, except that they will also drop received packets instead of forwarding them. We assume an attack model in which the malicious nodes act individually and do not collude with others, *i.e.*, several attackers do not cooperate or coordinately misbehave in order to evade detection systems. A further extension of our work would imply the combination and evaluation of our technique with others which specifically deal with collusion attacks. For instance, our scheme might be complemented by performing some end-to-end checking, like the one proposed in [31]. This would determine if data packets truly reach the destination and, therefore, chains of colluding attackers could be detected. Other collusion related proposals that might be adopted are [32, 33, 34].

4.2. Overview of the Detection Approach

As in other detection proposals, our approach follows a window basis procedure to consider or not a node as malicious discretely over time. This way, a set of network related features is first obtained for each node in a given temporal window of analysis. From these features, the probability values given in Section 3 are afterwards estimated. Finally, a decision about the behavior of a target node is taken.

The probability of occurrence of packet dropping can be calculated from (3) as:

$$P_{DROP} = 1 - \frac{P_{FWD}}{[1 - (P_{COL} + P_{MOB})]} \tag{4}$$

This dropping probability is subsequently compared to a predefined detection threshold θ. If P_{DROP} is greater than this threshold and according to an anomaly-based approach, we conclude that the analyzed node is malicious, and legitimate otherwise:

$$node = \begin{cases} \text{malicious}, & \text{if } P_{DROP} \geq \theta \\ \text{legitimate}, & \text{otherwise} \end{cases} \tag{5}$$

Obviously, the operating point of the detector depends on the value used for the detection threshold. If θ is set to a low value, more malicious nodes in the network will be detected, but also more legitimate nodes will be misclassified as malicious (*i.e.*, false positive rate increases). On the contrary, the use of high values for θ will result in fewer malicious nodes being detected, but it will also produce low false positives. As it will be seen in Section 6, a tradeoff value for these two situations is typically the best choice.

4.3. Parameters Estimation

Here we discuss how to calculate the probabilities involved in our analytical model taking into account different features obtained from the network. The parameters to be estimated are P_{FWD}, P_{COL} and P_{MOB}. An empirical approximation is going to be used to estimate both P_{FWD} and P_{COL}.

First, P_{FWD} can be calculated as the percentage of data packets forwarded by the node with regard to the number of packets received by it. With this purpose, the IDS monitors the traffic of the analyzed node in search of received data packets whose destination is not the analyzed node itself. The estimator for this probability, $\hat{P}_{FWD}$, is:

$$\hat{P}_{FWD} = \frac{\#DATA_{FWD}}{\#DATA_{RECV}} \tag{6}$$

It must be reminded that, only if a node is not the final destination of the packet and there exists a valid route, the packet will be counted as a received data packet in $\#DATA_{RECV}$.

About the legitimate packet discards, our model distinguishes two possible situations: *(i)* the one happening due to collisions or channel errors, which takes into consideration those RTS retransmissions not exceeding the SRL value and contributes to P_{COL}; and *(ii)* the situation contributing to P_{MOB}, which is caused by broken links and considers those RTS retransmission exceeding the SRL value.

Regarding P_{COL}, since the associated effect is related to the traffic load, the number of RTS packets sent by the node without a proper CTS reply ($\#RTS_{SENT} - \#CTS_{RECV}$) is computed, as well as the total number of attempts to seize the channel. As said, only those packets which are not directly related to broken links situations are taken into account, *i.e.*, those RTS retransmissions which do not exceed the SRL limit. In summary, an estimator for the collision and channel error probability, $\hat{P}_{COL}$, can be computed as:

$$\hat{P}_{COL} = \frac{\#RTS_{SENT} - \#CTS_{RECV}}{\#RTS_{SENT}} \tag{7}$$

Finally, we estimate P_{MOB}. The proposed estimator for the probability of a broken link situation can be easily computed, since it will take one of just two values. $\hat{P}_{MOB}$ is set to 1 when the number of RTS retransmissions exceeds the SRL limit in a measuring window, since here the node considers that it does not have a connection with the next hop. The estimator is set to 0 otherwise, because the link is not considered to be down. That is,

$$\hat{P}_{MOB} = \begin{cases} 1, & \text{if } \#RTS_{SENT} > SRL \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Since mobility related situations may have some peculiarities, it is convenient to dedicate a detailed discussion to study how mobility is estimated and which circumstances may occur. Thus, for a better understanding of what happens when a broken

link is detected, it is necessary to provide some information about the performance of the routing protocol used and its interactions with the MAC IEEE 802.11 protocol.

AODV (Ad hoc On-demand Distance Vector) routing protocol [35] is considered as a case study in this work. From the results obtained for it, the proposed methodology may be easily extended to other similar protocols, like DSR or others [36].

AODV is a reactive protocol, so that the routes to a specific destination are established on-demand only when they are needed. This way, if a node requires a communication, it first broadcasts a Route REQuest message (RREQ) which is forwarded by other nodes in the environment. Then, if a node receives the RREQ message and it knows a valid route to the destination, it sends back a Route REPly message (RREP). This process is known as a *route discovery* procedure.

To work properly, each node keeps track of the nodes it can communicate directly (*i.e.*, its neighbors) by listening to HELLO messages which are periodically broadcasted by each node. This implies however a high bandwidth and energy consumption, so it is more common to use the well known IEEE 802.11 RTS/CTS procedure in MANETs. As explained, when the system exceeds the maximum number of retransmissions allowed, SRL, AODV concludes that the link is broken and initiates a *route maintenance* process. At this point, two possibilities appear (see Fig. 2):

- *Scenario 1*: In the case that the broken link is closer to the source node than to the destination one, the intermediate node throws the route away and sends back a Route ERRor message (RERR) to alert its precursors about the link fail. In such a case, the precursor nodes stop sending packets to the intermediate node and retransmit the RERR messages.
- *Scenario 2*: In the case that the link is closer to the destination node than to the source, the intermediate node tries to repair locally the route. For that, it sends a RREQ message in a similar way that the source node would do. If the route cannot be repaired after a period of time, the intermediate node will send a route error message, RERR, to its precursors.

Note that the node with a broken link behaves like a malicious node during a certain time, as it continues receiving messages but it cannot forward them. The route maintenance process in Scenario 2 can take up to dozens of second, so that the mentioned period of time is even longer here than in the Scenario 1 case.

Therefore, it is important to distinguish which scenario has taken place, since the decision about how long the probability $\hat{P}_{MOB}$ will be considered 1 (and therefore, P_{DROP} set to 0 and the node considered as legitimate) is not trivial at all. To solve this, the IDS will also monitor if a RREQ message is sent by the node when a broken link is detected. In such a case, $\hat{P}_{MOB}$ will be set to 1 during a certain time T. The choice of the proper value for T will be discussed in Section 6.

4.4. Enhanced Windowing for Collecting Features

As established above, the estimation of the malicious dropper nature of a node involves (5), (6), (7) and (8). The network features that appear in these equations are RTS_{SENT}, CTS_{RECV}, $DATA_{RECV}$ and $DATA_{FWD}$. A normal methodology is

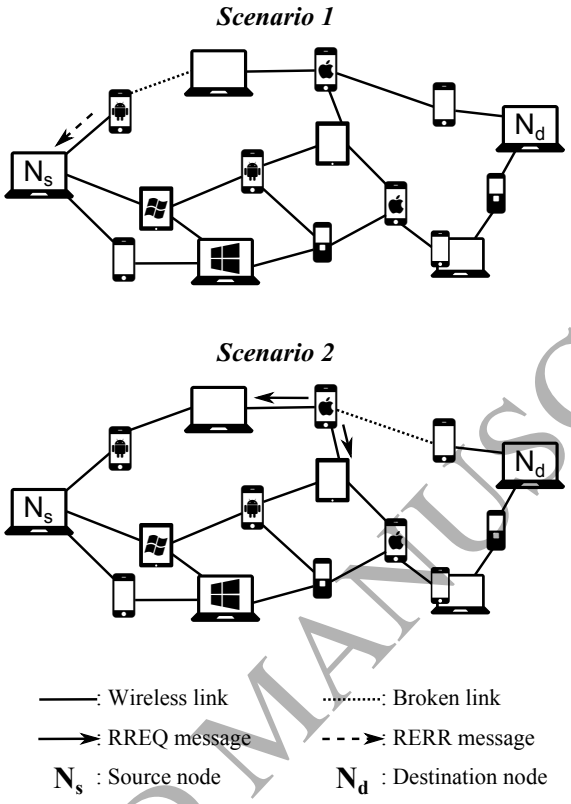


Figure 2: Scenarios which may occur when the RTS/CTS mechanism fails. The first one corresponds to the case when the broken link is closer to the source node than to the destination, while in the second scenario the broken link is closer to the destination node.

to monitor these features by considering temporal observations over successive non-overlapped analysis windows of fixed duration. However, this methodology presents two main drawbacks:

i. The first one is related to situations where the temporal window ends just after the transmission of an RTS packet. Here, it is not possible to guess if the packet will be properly replied, if a collision will occur or if a mobility situation will happen. This fact can lead to undesirable effects due to the *discontinuities* caused by the windowing. Fig. 3 shows a specific example of that situation. In the figure, dotted lines represent the end of the time windows. As it can be seen, the temporal window could end during the retransmission of an RTS, *e.g.*, just after $RTS\ \#5$ is sent. In this case, the whole circumstance which characterizes a mobility related situation will not be caught in any of the temporal windows, and therefore, the legitimate drops due to mobility will not be considered as legitimate, because mobility will not be detected.

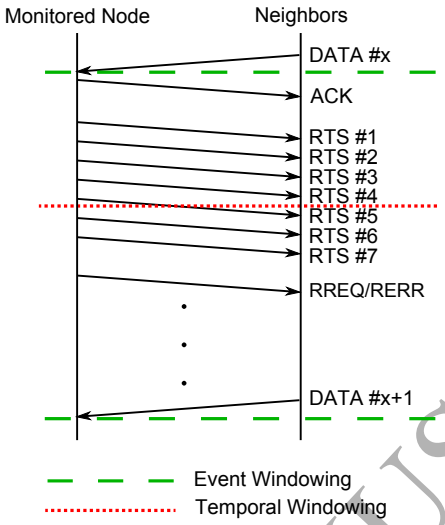


Figure 3: *Discontinuities* caused by the time-based windowing.

ii. The second drawback is related to the fact that, even if during a certain interval there are no features to collect or there are few, they will be analyzed anyway, thus obtaining *biased information* that could lead to wrong detection results. Fig. 4 shows how this problem can arise. Suppose that in the temporal window only few data packets are received or, as shown, just one is. Suppose that the

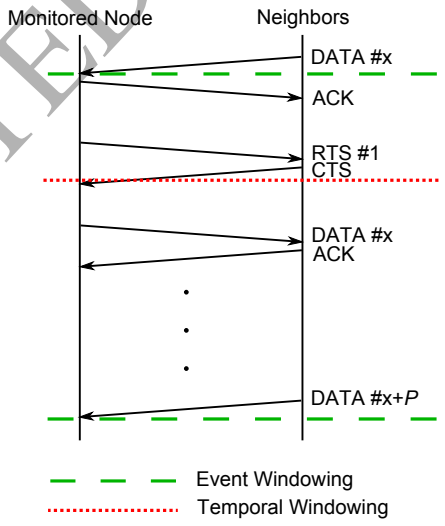


Figure 4: *Biased information* due to the time-based windowing.

temporal window ends even before the CTS in response to $RTS\ \#1$ is received too. In such a case, the analytical model will consider a very high percentage of dropped packets (100% in our example), thus leading to the misclassification of the node as malicious.

To overcome these inconveniences, we propose to carry out an *event-based windowing* procedure instead of a time-based one. That is, the features are obtained for *non-overlapping windows of P received data packets* for each node in the network. Examples of the differences between both types of windowing are shown in Fig. 3 and Fig. 4, where dashed lines correspond to the end of the event windows.

With event-based windows, the first problem *(i)* is avoided, since the end of each window will always coincide with a data packet reception event. Fig. 3 evidences that, by employing the event-based windowing, we ensure that mobility situations can be fully collected. Either if the collection is performed after $DATA\ \#x$ is received or if it is performed when the node receives $DATA\ \#x + 1$, the whole event is collected.

Regarding the problem *(ii)*, now the collection of statistics will always consider the same number of events, P, thus attenuating the effect of *biased information*. Fig. 4 illustrates how our event-based scheme guarantees that a representative amount of data are used, thus minimizing potential wrong classifications.

Besides the solution of these reported problems, an additional significant advantage should be mentioned for the proposed event-based windowing scheme. It refers to the fact that, if a given node is not receiving traffic at all, it makes no sense to perform a detection process every certain time, as this only involves a waste of the resources of the node. Thus, the use of the proposed windowing implies resources saving in nodes with scarce activity, since the detection procedure is expected to be launched fewer times, involving lower computation and, consequently, lower energy consumption.

4.5. Complexity

Here we briefly discuss the complexity of the proposed scheme, taking into consideration both storage and computational requirements for each IDS instance.

Regarding memory needs, each IDS procedure running in a given node just requires to handle 5 features (4 of them integers and 1 boolean) for each node monitored. Thus, considering a typical 32-bit architecture, only 17 bytes of storage per monitored node are enough to allocate these features.

In terms of computational overhead, for each node to be monitored our scheme executes a maximum of 13 basic operations (arithmetic, comparisons and assignments) per analysis window. Since the number of computations performed is always the same, we can conclude that the complexity of our scheme is constant for each monitored node. Expressed in *Big* $\mathcal{O}$ notation, the complexity of the proposed detector is $\mathcal{O}(1)$ per analysis window and monitored node, which is lower than that of most data mining techniques, usually of order $\mathcal{O}(n)$, $\mathcal{O}(n^2)$, or even greater. For instance, the detection performed by the SVM classifier used in [27] requires between 2,700 and 9,000 computations per analysis window and monitored node.

4.6. Summary of the Detection Approach

In order to determine if a given node N_i in the network is behaving maliciously as a packet dropper, some main features are relevant:

- $\#RTS_{SENT,i}$: total number of RTS messages sent by the node N_i.
- $\#CTS_{RECV,i}$: total number of CTS messages received by node N_i.
- $\#DATA_{RECV,i}$:total number of data packets received by node N_i.
- $\#DATA_{FWD,i}$: total number of data messages forwarded by node N_i.
- $RREQ_i$: a boolean feature whose value is $TRUE$ if a RREQ message has been broadcasted by node N_i, and $FALSE$ otherwise.

Taking into account all the above facts, and considering the five previous features ($\#RTS_{SENT}$, $\#CTS_{RECV}$, $\#DATA_{RECV}$, $\#DATA_{FWD}$ and $RREQ$), we finally derive the probability of occurring a dropping attack, P_{DROP}, by reducing the criterion in (4), which decides if a node is malicious or not, to the following expression:

$$P_{DROP} = \begin{cases} 0, & \text{if } \hat{P}_{MOB} = 1 \\ 1 - \frac{\hat{P}_{FWD}}{1-\hat{P}_{COL}}, & \text{otherwise} \end{cases} \tag{9}$$

The detection process is described through Algorithm 1.

Algorithm 1 Pseudo-code for the dropping detection

```
for each window ω in the monitoring time do
    for each node N_i in the network do
        Obtain P̂_FWD using (6)
        Estimate P̂_COL with (7)
        Estimate P̂_MOB using (8)
        Compute P_DROP with (9)
        if P_DROP < θ then
            if RREQ == FALSE then
                N_i is legitimate during the window ω.
            else
                N_i is legitimate for every ω during T.
            end if
        else
            N_i is malicious during the window ω.
        end if
    end for
end for
```

It must be noted that the detection proposal is based on an analytical model which employs simple features to carry out the detection process. The use of this methodology incurs lower computational overhead in comparison with more sophisticated techniques based on data mining or machine learning algorithms, which require higher

computational complexity. Also, the proposed model overcomes the need for an extensive training phase, thus minimizing the large training data sets (labeled or not) which must be used, as well as the associated overhead. However, the operating point of our system must still be empirically obtained for specific scenarios or network conditions.

5. Implementing the Packet Dropping Detection Scheme

Beyond the theoretical development of our cross-layer malicious packet dropping detection method, in the following we discuss how to deploy our proposal.

First, we discuss a *stand-alone* approach, where the features used to determine the malicious behavior of a given node are collected exclusively by the own node. As the features collection is directly and locally made by each node, the IDS can access all the information, being able to perform a more accurate detection. For example, the IDS has access not only to the statistics of sent packets by a given node, but also to those corresponding to the received packets. Thus, the five features presented in Section 4 ($\#RTS_{SENT}$, $\#CTS_{RECV}$, $\#DATA_{RECV}$, $\#DATA_{FWD}$ and $RREQ$) can be employed in a straightforward way.

Note that this *stand-alone* approach is not really a feasible implementation, as it assumes that every node of the network is trustworthy. Yet, this is an interesting case study, as it will give us the theoretical bounds of performance for our system and, for this reason, it will be included in our experimental evaluation (see Section 6).

As a feasible alternative, we suggest a *distributed* gathering architecture for the IDS deployment. In this case, the features for estimating the potential malicious behavior of a given node are indirectly collected by other nodes, which cooperate in order to provide a collaborative data collection process. These nodes, called *monitors*, must promiscuously collect and analyze all the important features within their communication area. In promiscuous mode, a node is able to gather all the frames sent throughout its vicinity, regardless of their destination addresses. Nevertheless, the monitor node cannot learn with certainty whether the packet was correctly received by the wireless interface of the neighbor monitored node. Therefore, two of the needed features for the IDS operation, $\#CTS_{RECV,i}$ and $\#DATA_{RECV,i}$, are approximated as:

- $\#CTS_{SENT,i}$: total number of CTS messages sent towards node N_i by its neighbor nodes.
- $\#DATA_{SENT,i}$: total number of data packets sent to node N_i by its neighbor nodes.

It must be noted that the use of these two features is just an approximation, because a sent packet can be lost due to some reasons. However, in the experimentation presented in Section 6 we show the effect of this estimation and demonstrate that it does not degrade significantly the performance of the detection system. Once the monitors have collected all the needed information about a given node, the behavior of this node is estimated by using the proposed heuristic.

We must take into account that the detection process is independent of the features collection process. This way, although the data collection process proposed here is collaborative, since several monitors collect and share the needed information, the specific

implementation of the detection approach can be carried out in different ways once the information provided by the monitor nodes is gathered: centralized, if a central node gathers all the information and computes the heuristic; clustered, if the monitors are deployed in different regions and share the information with those belonging to the same region; isolated, if each monitor, after gathering the information shared by others, computes individually the heuristic; or even collaborative, if several monitor nodes cooperate to compute the heuristic after sharing the needed features.

It should also be noted that the use of monitor nodes implicitly assumes that they are trustworthy, *i.e.*, we presuppose the feasibility of a trust management system, assumption generally adopted in similar works in the bibliography, like [31] or [37]. This, however, is not strictly necessary, as some kind of voting process may be alternatively implemented to decide about potential differences in the values of the features due to the existence of one or more malicious nodes trying to evade the detection. This way, the set of trustworthy monitor nodes can be substituted by the own neighbor nodes of a given one in the network.

Finally, another implementation issue is that of dealing with redundant information coming from different monitors for the same event. Although this is a minor problem, it should be carefully addressed by synchronizing monitor nodes and efficiently filter the information in order to not affect the IDS performance.

6. Performance Evaluation

This section describes the experimental framework used to validate the packet dropping IDS approach proposed here, and the results obtained from that evaluation. We have carried out extensive experiments to verify the proper performance of our proposal.

6.1. Experimental Environment

Network Simulator 2 (NS-2) [38] has been adopted as evaluation platform [39] to simulate several deployments for a MANET environment. The simulation area covers a 1,000 m x 1,000 m region, where each node has a communication range of 250 m. According to the previous discussion in Section 4, AODV is chosen as the routing protocol, and IEEE 802.11b as the MAC protocol. Other parameters chosen for simulation are those shown in Table 1 and Table 2. It should be noted that default values are selected for them.

The total number of user nodes in our network is 25, the number of them launching dropping attacks varying from 1 to 20. Moreover, the number of application related flows is set to 20, where each flow consists of a Constant Bit Rate (CBR) connection, with a bit rate of 4 packets/s and a payload size per packet of 512 bytes.

The propagation model considered is Two Ray Ground [40], and the nodes have a communication range of 250 m.

The mobility model for the nodes refers to the Random Waypoint Model (RWP) [41], with a fixed minimum speed equal to 1m/s and a maximum speed varying from 5 to 30 m/s. The pause time is 15 s, that is, after reaching the desired destination the node waits for 15 s before choosing a new random destination and repeating the procedure.

Table 1: Configuration Parameters in NS-2.

Parameter	**Value**	**Parameter**	**Value**
Radio Model	*TwoRayGround*	MAC Type	*802_11*
Antenna	*OmniAntenna*	$CW_{min/max}$	31/1023
Tx/Rx Gain	1	Slot Time	20 μs
High	1.5 m	SIFS	10 μs
NIC	*WirelessPhy*	Data Rate	11 Mb
Capture Thr	10 dB	Basic Rate	2 Mb
Carrier Thr	$1.5e^{-11}$ W	PLCP Rate	1 Mb
Rx Thresh	$3.6e^{-10}$ W	*SSRC*	7
Tx Power	0.2818 W	RTS Thr	0 *bytes*
Frequency	914 MHz	Queue Type	*PriQueue*
Loss Factor	1	Size	50

Table 2: AODV Parameters in NS-2.

Parameter	**Value**	**Parameter**	**Value**
Active Route T/O	10 s	RREP Wait Time	1 s
Rev. Route Life	6 s	#RREQ Retries	3
Max. RREQ T/O	10 s	Link Layer Det.	yes

According to the thorough investigation performed by the authors in [42] to model the error probability in wireless links under several conditions, we have initially fixed the channel error probability to 0.37%. Moreover, in order to test our scheme under other different circumstances, we have varied the channel error probability from 0.37% to 7%, as shown in Section 6.2.3.

Malicious nodes in the environment are configured to drop 20% of the data packets received to be forwarded towards a final destination.

The upper bound for the time involved in the local link repairing process depends on some parameters of AODV. Among them, we should remark some randomness due to the binary exponential backoff mechanism used to avoid congestion. Taking into account all of the above, the mentioned bound is around 60 s and, as a consequence, this is the final duration selected for the time window T during which $\hat{P}_{MOB}$ is set to 1 (see end of Section 4.3).

6.2. Detection Results

The detection performance of the introduced IDS is evaluated by means of two well known parameters, namely the True Positives Rate (TPR), or detection accuracy/rate, and the False Positives Rate (FPR). As known, we obtain a number of operating points to estimate the Relative Operation Characteristic (ROC) curve by varying the decision threshold θ in (5). It is important to note that the ROC curve has been obtained by

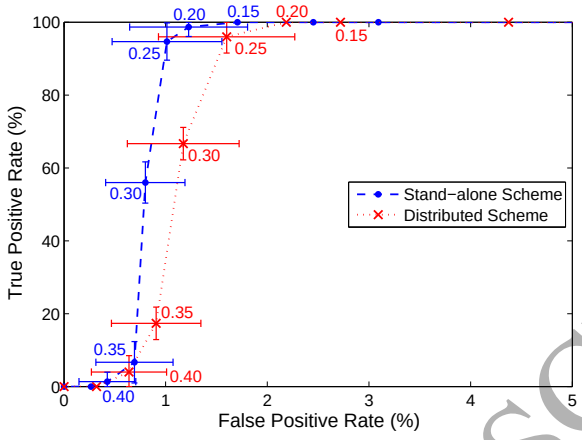


Figure 5: ROC curve of the stand-alone and distributed implementations by varying the θ parameter.

repeating 75 times (with different seed values) each of the simulations[2]. The maximum speed of the nodes is set to 10 m/s in all the cases. This way, our results are comprised of the mean value of these 75 simulations and the 95% confidence intervals of these averages.

Fig. 5 shows the ROC curves obtained for both stand-alone and distributed implementations. As expected, the results obtained for the distributed-collection IDS approach are a little bit worse than the ones got in the stand-alone case. This is due to the fact that, in the distributed case, an approximation for two features is used, which considers that every sent CTS and data packet will be received. As a very little portion of these packets can be lost due to channel errors or collisions, the performance of this scheme is slightly deteriorated.

As shown in the curve, FPR improves and TPR decreases as the detection threshold θ increases. On the contrary, lower detection thresholds result in better TPR values, but in increasing FPR figures. This is coherent with the fact that upper (lower) values for the detection threshold imply lower (upper) sensitivity of the system against "malicious" behaviors. Of course, in such a case the FPR (TPR) values are improved.

The optimal operating point of the system is achieved empirically from the above results. In particular, θ (which must be in the range $[0-1]$, as it is compared to a probability value) is set to 0.15, as it seems to represent a good tradeoff between FPR and TPR.

[2]Although the Central Limit Theorem proves that 30 runs (N=30) would be sufficient for our simulations to approximate a Normal distribution, the Law of Large Numbers states that the larger the number of repetitions, the closer the average of the results to the expected value. Most of the related works choose N in the range $[5-20]$, but we set N to 75 to increase the accuracy of the results.

6.2.1. *Influence of Window Size*

The size of the selected event-based window for collecting the features has also been chosen through experimental results. Tests using 50, 75, 100 and 125 received data packets have been performed and the results are shown in Table 3. As before, 75 repetitions with different seeds and a maximum speed of 10 m/s are used.

Table 3: Operating Point for Different Window Sizes.

Window	**Stand-alone Scheme**		**Distributed Scheme**	
size	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**
50	100.0±0.00	6.51±1.30	100.0±0.00	8.75±1.48
75	100.0±0.00	2.99±0.77	100.0±0.00	4.85±1.06
100	100.0±0.00	1.92±0.64	100.0±0.00	2.88±0.84
125	98.67±2.61	1.17±0.49	98.67±2.61	1.82±0.67

As expected, the bigger the window the better detection capabilities in terms of FPR, although the size of the window cannot grow indefinitely, since this fact leads to increasing delays in the detection process. Accordingly to these results, in the experiments described in what follows, the window size has been fixed to 100 data received packets, value which provides a good tradeoff between the delay in detection and the values for TPR and FPR.

6.2.2. *Influence of Mobility*

We now study the detection efficiency for different mobility conditions. Six scenarios are thus simulated, with speed values from 5 m/s to 30 m/s to consider a wide range of possibilities. Fig. 6 shows graphically both TPR and FPR for such different scenarios.

As shown, both the stand-alone and the distributed implementations achieve excellent results regarding the two metrics considered. TPR exceeds in all scenarios 97%,

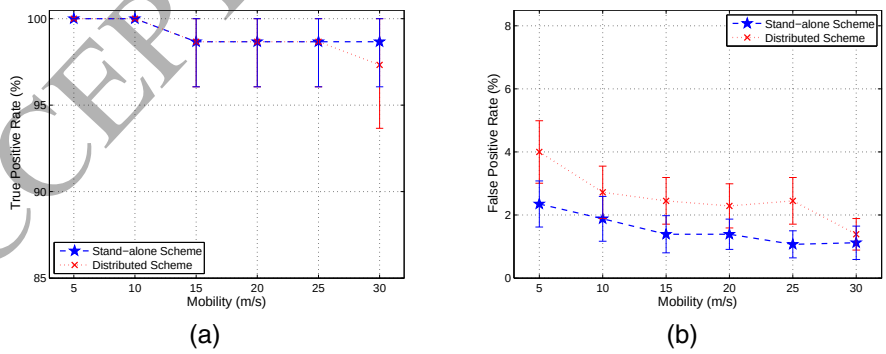


Figure 6: TPR (a) and FPR (b) in different mobility scenarios, for stand-alone and distributed schemes.

meanwhile FPR always remains below 4%. These results confirm the capabilities of our model which, as expected, are slightly degraded as mobility increases.

On the other hand, as previously discussed, the detection results obtained for the distributed-collection IDS approach are a bit worse than the ones obtained in the stand-alone case.

6.2.3. *Influence of Channel Error Probability*

The detection efficiency under different channel error probabilities has also been studied. Although the results in [42] show that an error probability of 0.37% is a good estimation for IEEE 802.11 channels, we have also considered higher error probabilities to test the performance of our scheme when different channel characteristics, like shadowing or multipath fading, cause large packet losses. Fig. 7 depicts graphically both TPR and FPR in these situations.

As shown, TPR degrades for the stand-alone implementation as channel error probability increases. This is mainly due to the fact that, for every received packet, it is more likely that the node has to retransmit it several times. Remind that a MAC level retransmission is performed in IEEE 802.11 links if no positive acknowledgment is received. This way, these multiple retransmissions can hide the dropping behavior of the malicious nodes, especially when the dropping rate is not very high, as in our case. As a result, FPR also decreases.

For the distributed implementation, although a node may receive packets with errors which do not have to be forwarded until their correct reception, it is likely that some of the monitors consider the packets to be properly received, treating them like dropped packets. Therefore, TPR (but also FPR) will be higher.

6.2.4. *Influence of Number of Malicious Nodes*

Another set of experiments are aimed at analyzing the performance of both detection approaches for an increasing number of malicious nodes. This is intended to demonstrate that the performance of the proposals is not severely degraded despite

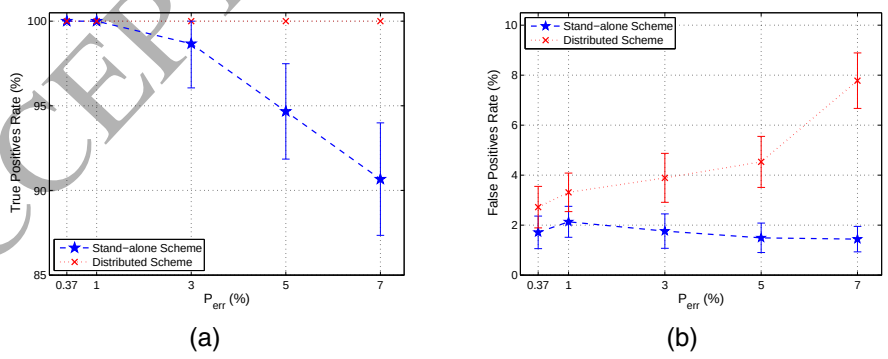


Figure 7: TPR (a) and FPR (b) for different channel error probabilities, for stand-alone and distributed schemes.

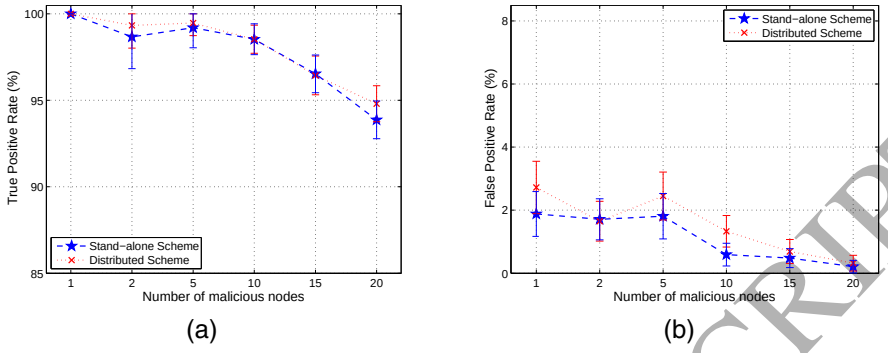


Figure 8: TPR (a) and FPR (b) for different number of malicious nodes, for stand-alone and distributed schemes.

several nodes in the network are compromised. The results obtained are presented in Fig. 8.

They show that, even when a high number of malicious nodes exist in the network, the detection accuracy of the proposed scheme remains keeping FPR below 3%. That is, the proposal provides again very good detection results. It can be noted that, even in the latter case, in which 80% of nodes behave as droppers, the detection rate of the proposed schemes is above 93%.

6.2.5. Discussion and Comparison of Detection Results

From the above figures, it is clear that our IDS related proposal can efficiently detect the malicious nodes in the environment with an overall accuracy upper to 93%. It should also be remarked that the system gets very low false positives rates, which are under 4% in any case.

Firstly, we have compared our scheme with that introduced in [29]. Although not directly comparable, as such system was not designed for networks with mobility, it is interesting to note the difference between the detection capabilities of both approaches. This way, the optimal operating point of our system in the baseline scenario (10 m/s, error probability of 0.37% and window size of 100) achieves 100% TPR and less than 3% FPR, while the previous scheme is still able to keep TPR at 100%, but soaring FPR up to an unacceptable 48%.

We have also carried out a more realistic comparison among our results and those exhibited by other similar schemes and comparable scenarios) in [22, 23, 27]. As indicated in Section 2.3, [22] introduces a multi-layer approach composed of three local subsystems (corresponding to MAC, routing and application layers) that make use of a Bayesian classifier, Markov chains and an association rule algorithm for global intrusion detection. Authors in [23] deal with black hole attacks in MANETs through an anomaly detection scheme which uses a dynamic training method. In this scheme, the number of control packets sent and received, as well as the mean differences between the destination sequence numbers sent and the ones received, are used to detect deviations from the expected normal network behavior. The state is dynamically updated

over the time to adapt the detector to the evolution of the system and thus to improve the detection accuracy. Finally, authors in [27] implement a Fisher Discriminant Analysis (FDA) procedure to remove data with low-information content, which makes the developed SVM classifier feasible for ad hoc nodes.

In order to show the suitability of this comparison among detection results, Table 4 presents for each scheme some parameters defining the scenarios considered. The similarities between all of them allow us to deem the validity of the comparative. It must be noted that the work in [22] does not include information regarding the speed of the nodes and, therefore, we have assumed that the detection results are independent of the mobility.

Table 4: Comparison of some Principal Scenario Related Characteristics for Different Detection Schemes.

Characteristics	**Our scheme**	**Ref. [22]**	**Ref. [23]**	**Ref. [27]**
# Nodes	25	30	30	30-50
# Attackers	1-20	1	1	3
Traffic density	$\approx 80\%$	-	$\approx 100\%$	$\approx 60\%$
Mobility	RWP (5-30 m/s)	RWP (-)	RWP (1-20 m/s)	RWP (0-30 m/s)

Fig. 9 shows how our results overcome those obtained by the other schemes. For example, results for TPR in [23] show that it never exceeds 80%, while 12% is the minimum value for FPR. Authors in [22] achieve detection capabilities similar to ours, obtaining a lower TPR but also a lower FPR. However, this scheme integrates three different subsystems (a Bayes-based classifier, Markov chains and an association rule algorithm), which results in a more complex approach. In a similar way, results in [27] are comparable to ours, but the system incurs in a huge overhead due to the use of non-linear SVM and FDA.

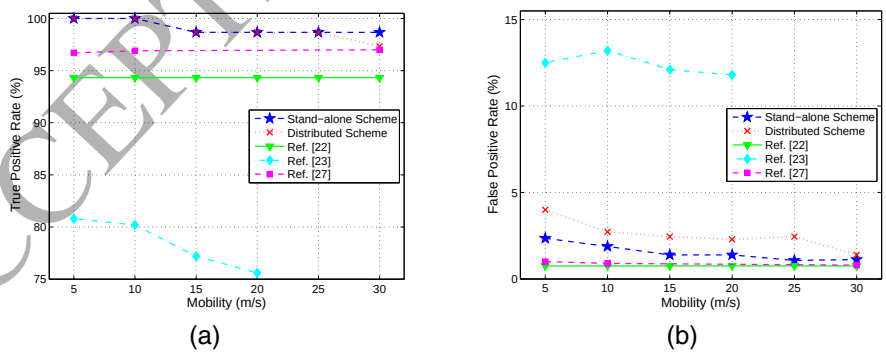


Figure 9: TPR (a) and FPR (b) for different mobility scenarios, for several schemes.

7. Conclusions and Future Work

We have proposed in this paper a novel methodology intended to detect malicious packet dropping behaviors in mobile ad hoc networks. For that, features from MAC and network layers are considered. Moreover, the cross-layer approach is based on an analytical model that represents the forwarding process in an ad hoc network. It should be noted that the use of this simple scheme overcomes the computational overhead in more sophisticated approaches in the literature, which are usually based on data mining algorithms. Also, it demonstrates to improve the detection performance under several circumstances not usually considered by previous related works, and which result in unsuitable values for FPR.

An event-based windowing procedure for features collection and subsequent analysis process is proposed too. It eliminates some limitations of normal time-based windowing, and is able to improve the performance in nodes which exhibit low or null activity, resulting in a lower consumption of resources.

Moreover, we have also discussed a distributed scheme for implementing the IDS approach. This makes use of a set of monitor nodes to collect the features in a collaborative way.

We have verified the good performance of our system by means of simulation, where an extensive set of different scenarios have been analyzed. The results provided clearly highlight the goodness of the IDS approach, which experiences 93% overall TPR with less than 4% FPR. This far overcomes the performance obtained by other similar schemes in the literature.

As shown, experimental results are very encouraging. This way, we are going for such direction through the improvement of some aspects of our approach in the near future:

- In distributed IDS for mobile ad hoc network is highly recommended to reduce the information exchanged and shared. For our collection-distributed design, we are working in the development of a communication protocol that takes into account the restrictions resulting from the MANET context. Therefore, referred mechanism must involve low overhead in the network, the data exchanged being restricted to concise information (events) resulted from locally pre-processed features.

- Additionally to the data collection, it is desirable to completely distribute the IDS tasks, thus enabling intrusion detection and alert management in a distributed manner. Nodes should cooperate to provide alert correlation and attack response, as well.

- This way, the inclusion of trust-based schemes as response mechanism to face malicious packet dropping situations is also of interest to provide survivable measures in the network.

- To implement effective methods for the adaptive determination of the detection threshold, since this value is dependent on the specific network conditions. Different schemes can be analyzed for that, *e.g.*, obtaining an average of previous

threshold values, computing the threshold as a function of the mobility speed, etc.

- Finally, we are planning to extend our approach to include an attack model where several nodes work in collusion to evade the detection process. For that, some mechanisms existing in the literature are going to be first considered.

Acknowledgment

This work has been partially supported by Spanish *Ministerio de Ciencia e Innovación* (MICINN) through project TEC2011-22579, by Spanish *Ministerio de Economía y Competitividad* (MINECO) through project TIN2014-60346-R, by Spanish *Ministerio de Educación, Cultura y Deporte* (MECD) through the grant "University Professor Training Program" (FPU, Ref.: AP2009-2926) and by the "FPU P6A Grants Program" of the University of Granada.

References

[1] T. S. Rappaport, A. Annamalai, R. M. Buehrer, W. H. Tranter, Wireless Communications: Past Events and a Future Perspective, IEEE Communications Magazine 40 (5) (2002) 148–161.

[2] J. He, S. Ji, Y. Pan, Y. Li (Eds.), Wireless ad-hoc and Sensor Networks: Management, Performance, and Applications, Boca Raton, FL: CRC Press, 2014.

[3] P. Brutch, C. Ko, Challenges in Intrusion Detection for Wireless Ad-Hoc Networks, in: Proc. of the Symposium on Applications and the Internet Workshops (SAINT), 2003, pp. 368–373.

[4] P. García-Teodoro, L. Sánchez-Casado, G. Maciá-Fernández, Taxonomy and Holistic Detection of Security Attacks in MANETs, in: S. Khan, J. Lloret Mauri (Eds.), Security for Multihop Wireless Networks, CRC Press, 2014, pp. 1–12.

[5] Y.-A. Huang, W. Lee, Attack analysis and detection for ad hoc routing protocols, in: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection, Vol. 3224 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 125–145.

[6] A. Nadeem, M. Howarth, Protection of MANETs from a range of attacks using an intrusion detection and prevention system, Telecommunication Systems 52 (4) (2013) 2047–2058.

[7] S. Djahel, F. Nait-abdesselam, Z. Zhang, Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges, IEEE Communications Surveys & Tutorials 13 (4) (2011) 658–672.

[8] L. Sánchez-Casado, R. Magán-Carrión, P. García-Teodoro, J. E. Díaz-Verdejo, Defenses Against Packet Dropping Attacks in Wireless Multihop Ad Hoc Networks, in: S. Khan, J. Lloret Mauri (Eds.), Security for Multihop Wireless Networks, CRC Press, 2014, pp. 377–400.

[9] D. Djenouri, N. Badache, New approach for selfish nodes detection in mobile ad hoc networks, in: Proc. of the Workshop 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecurComm), 2005, pp. 288–294.

[10] D. Djenouri, N. Ouali, A. Mahmoudi, N. Badache, Random Feedbacks for Selfish Nodes Detection in Mobile Ad Hoc Networks, in: T. Magedanz, E. Madeira, P. Dini (Eds.), Operations and Management in IP-Based Networks, Vol. 3751 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 68–75.

[11] K. Balakrishnan, J. Deng, P. K. Varshney, TWOACK: preventing selfishness in mobile ad hoc networks, in: Proc. of the IEEE Wireless Communications and Networking Conference (WCNC), Vol. 4, 2005, pp. 2137–2142.

[12] K. Liu, J. Deng, P. K. Varshney, K. Balakrishnan, An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs, IEEE Transactions on Mobile Computing 6 (5) (2007) 536–550.

[13] S. Djahel, F. Nait-Abdesselam, A. A. Khokhar, An Acknowledgment-Based Scheme to Defend Against Cooperative Black Hole Attacks in Optimized Link State Routing Protocol, in: Proc. of the IEEE International Conference on Communications (ICC), 2008, pp. 2780–2785.

[14] D. Djenouri, N. Badache, On eliminating packet droppers in MANET: A modular solution, Ad Hoc Networks 7 (6) (2009) 1243–1258.

[15] S. Biswas, T. Nag, S. Neogy, Trust based energy efficient detection and avoidance of black hole attack to ensure secure routing in MANET, in: Applications and Innovations in Mobile Computing (AIMoC), 2014, 2014, pp. 157–164.

[16] S. Buchegger, J.-Y. Le Boudec, Performance Analysis of the CONFIDANT Protocol, in: Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc), 2002, pp. 226–236.

[17] H. Miranda, L. Rodrigues, Friends and Foes: preventing selfishness in open mobile ad hoc networks, in: Proc. of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW), 2003, pp. 440–445.

[18] C. Basile, Z. T. Kalbarczyk, R. K. Iyer, Inner-Circle Consistency for Wireless Ad Hoc Networks, IEEE Transactions on Mobile Computing 6 (1) (2007) 39–55.

[19] S. Marti, T. J. Giuli, K. Lai, M. Baker, Mitigating Routing Misbehavior in Mobile Ad Hoc Networks, in: Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom), 2000, pp. 255–265.

[20] Y. Zhang, W. Lee, Y. A. Huang, Intrusion Detection Techniques for Mobile Wireless Networks, Wireless Networks 9 (5) (2003) 545–556.

[21] Y. Huang, W. Fan, W. Lee, P. S. Yu, Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies, in: Proc. of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), 2003, pp. 478–487.

[22] S. Bose, S. Bharathimurugan, A. Kannan, Multi-Layer Integrated Anomaly Intrusion Detection System for Mobile Adhoc Networks, in: Proc. of the International Conference on Signal Processing, Communications and Networking (ICSCN), 2007, pp. 360–365.

[23] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, Y. Nemoto, Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method, International Journal of Network Security 5 (3) (2007) 338–346.

[24] A. Nadeem, M. Howarth, Adaptive Intrusion Detection & Prevention of Denial of Service Attacks in MANETs, in: Proc. of the 2009 International Conference on Wireless Communications and Mobile Computing (IWCMC), 2009, pp. 926–930.

[25] A. Nadeem, M. Howarth, An Intrusion Detection & Adaptive Response Mechanism for MANETs, Ad Hoc Networks 13, Part B (0) (2014) 368–380.

[26] J. F. C. Joseph, A. Das, B.-C. Seet, B.-S. Lee, CRADS: Integrated Cross Layer Approach for Detecting Routing Attacks in MANETs, in: Proc. of the IEEE Wireless Communications and Networking Conference (WCNC), 2008, pp. 1525–1530.

[27] J. F. C. Joseph, B.-S. Lee, A. Das, 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 8 (2) (2011) 233–245.

[28] M. D. Serrat-Olmos, E. Hernández-Orallo, J.-C. Cano, C. T. Calafate, P. Manzoni, A Collaborative Bayesian Watchdog for Detecting Black Holes in MANETs, in: G. Fortino, C. Badica, M. Malgeri, R. Unland (Eds.), Intelligent Distributed Computing VI, Vol. 446 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2013, pp. 221–230.

[29] T. Hayajneh, P. Krishnamurthy, D. Tipper, T. Kim, Detecting Malicious Packet Dropping in the Presence of Collisions and Channel Errors in Wireless Ad Hoc Networks, in: Proc. of the IEEE International Conference on Communications (ICC), 2009, pp. 1–6.

[30] L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro, An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs, in: Proc. of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 231–238.

[31] P. Agrawal, R. K. Ghosh, S. K. Das, Cooperative Black and Gray Hole Attacks in Mobile Ad Hoc Networks, in: Proc. of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC), 2008, pp. 310–314.

[32] L. Tamilselvan, V. Sankaranarayanan, Prevention of Co-operative Black Hole Attack in MANET, Journal of Networks 3 (5) (2008) 13–20.

[33] A. Baadache, A. Belmehdi, Fighting against packet dropping misbehavior in multi-hop wireless ad hoc networks, Journal of Network and Computer Applications 35 (3) (2012) 1130–1139.

[34] A. Baadache, A. Belmehdi, Struggling against simple and cooperative black hole attacks in multi-hop wireless ad hoc networks, Computer Networks 73 (0) (2014) 173–184.

[35] C. E. Perkins, E. M. Belding-Royer, S. R. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF, RFC 3561 [Online; Accesed Oct 10, 2014] `http://www.rfc-editor.org/rfc/rfc3561.txt`.

[36] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Blni, D. Turgut, Routing protocols in Ad Hoc networks: A survey, Computer Networks 55 (13) (2011) 3032–3080.

[37] D. Zhang, C.-K. Yeo, A Novel Architecture of Intrusion Detection System, in: Proc. of the 7th IEEE Consumer Communications and Networking Conference (CCNC), 2010, pp. 1–5.

[38] S. McCanne, S. Floyd, NS Network Simulator, [Online; Accessed Oct 10, 2014] `http://www.isi.edu/nsnam/ns/`.

[39] J. Friginal, D. de Andrés, J.-C. Ruiz, M. Martínez, A Survey of Evaluation Platforms for Ad Hoc Routing Protocols: a Resilience Perspective, Computer Networks 75, Part A (0) (2014) 395–413.

[40] T. S. Rappaport, Wireless Communications: Principles and Practice, IEEE Press, Piscataway, NJ, USA, 1996.

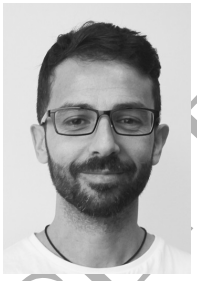
[41] D. B. Johnson, D. A. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, in: T. Imielinski, H. F. Korth (Eds.), Mobile Computing, Vol. 353 of The Kluwer International Series in Engineering and Computer Science, Springer US, 1996, pp. 153–181.

[42] J. Arauz, P. Krishnamurthy, Markov modeling of 802.11 channels, in: Proc. of the 58th IEEE Vehicular Technology Conference (VTC-Fall), Vol. 2, 2003, pp. 771–775.

Leovigildo Sánchez-Casado received his M.Sc. degree in Telecommunications Engineering in 2008, his M.Sc. degree in Electronics Engineering in 2009 and his Ph.D. in the field of Networking/Telematics in 2014, each from the University of Granada (Spain). In 2010, he joined the Department of Signal Theory, Telematics and Communications of the University of Granada as a researcher. He is a member of the "Network Engineering and Security Group (NESG)". His research interests are mainly focused on the area of network security and more specifically on intrusion detection and response, principally in ad hoc networks.

Gabriel Maciá-Fernández received his M.Sc. in Telecommunications Engineering from the University of Seville (Spain) in 1998 and the Ph.D. in Telecommunications Engineering from the University of Granada (Spain) in 2007. In the period 1999-2005, he worked as a specialist consultant in some technological companies (Enditel - Endesa, Vodafone S.A). He joined the University of Granada in 2005, and he is an Associate Professor at the Department of Signal Theory, Telematics and Communications. He is a member of the "Network Engineering and Security Group (NESG)". Currently, his research interests are focused on computer and network security, with special focus on intrusion detection, reliable protocol design, network information leakage and denial of service.

Pedro García-Teodoro received the B.Sc. degree in physics (electronics speciality) from the University of Granada (Spain) in 1989. In 1989, he received a grant from "Fujitsu Spain', and during 1990, he received a grant from "IBM Spain". From 1989 to 2011, he was Associate Professor and, since 2011, Full Professor at the Department of Signal Theory, Telematics and Communications of the University of Granada, and head of the research group "Network Engineering and Security Group (NESG)" of this University. His initial research interest was concerned with speech technologies, in which he developed his Ph.D. thesis in 1996. Since then, his professional interests have been in the field of computer and network security, especially focused on intrusion detection and denial of service attacks.



Roberto Magán-Carrión is a Ph.D. student at the Department of Signal Theory, Telematics and Communications of the University of Granada (Spain) and member of the research group "Network Engineering and Security Group (NESG)". He received his M.Sc. degree in Telecommunications from the University of Málaga (Spain) in 2008. His research interests are focused on Mobile Ad hoc NETworks (MANETs) security and more specifically on response (IR, Intrusion Response) and tolerant (IT, Intrusion Tolerant) solutions as part of global cross-layer security challenge with the aim of achieving survivable systems.