A Feedback Reputation Mechanism to Secure the Optimized Link State Routing Protocol

João P. Vilela João Barros

Instituto de Telecomunicações
Department of Computer Science
Universidade do Porto, Porto, Portugal
{joaovilela, barros}@dcc.fc.up.pt

***Abstract*— We consider the problem of securing routing information in Mobile Ad-hoc Networks (MANETs). Focusing on the Optimized Link State Routing protocol, we devise a *feedback* reputation mechanism which assesses the integrity of routing control traffic by correlating local routing data with feedback messages sent by the receivers of control traffic. Based on this assessment, misbehaving nodes are shown to be reliably detected and can be adequately punished in terms of their ability to communicate through the network. To the best of our knowledge, this is the first *practical* implementation of a reputation mechanism in a standardized proactive routing protocol for MANETs.**

I. Introduction

Among the numerous proposal for routing protocols in MANETs, the Optimized Link State Routing (OLSR) protocol [1], [2] is arguably one that offers promising performance in terms of bandwidth, required overhead and delivered traffic albeit at the cost of a wide range of security challenges, mostly with respect to the design assumption that all nodes comply with the protocol in the exchange of crucial topology information. Dropping this somewhat optimistic assumption, we evaluate the security risks inherent to OLSR and propose a feedback reputation mechanism for detection and punishment of misbehaving nodes; i.e. those that disrupt the network by generating false routing control traffic. Our main contributions are as follows:

- A taxonomy of security vulnerabilities that are specific to the OLSR protocol for MANETs;
- A feedback reputation mechanism to detect and punish the generation of false routing control information;
- Results of a simulation study to illustrate the induced overhead and the effectiveness of our feedback reputation mechanism;
- A detailed description of the technical issues that arose at all stages of the specification, implementation and validation process, as well as a set of solutions to put reputation-based schemes into practice.

The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS. DAIDALOS receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

This work was partly supported by the Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) under the grant SFRH/BD/28056/2006.

The rest of the paper is organized as follows. A review of related work is delayed until Section III to enable a prior description of the OLSR protocol and its inherent security aspects in Section II. Subsequently, Section IV describes our feedback reputation mechanism, and its performance is discussed in Section V. Section VI concludes the paper.

II. The Optimized Link State Routing Protocol

We start by briefly describing the OLSR protocol. As a proactive protocol, it exchanges routing information periodically and has routes immediately available when needed. As a link-state protocol, it maintains network topology information obtained from flooded routing control traffic which is used to determine the best path to every destination in the network.

OLSR offers, in fact, more than a pure link state protocol, because it provides the following features:

- *minimization of flooding* by using only a set of selected nodes, called *multipoint relays* (MPRs), to diffuse its messages to the network;
- *reduction of the size of control packets* by declaring only a subset of links with its neighbors who are its *multipoint relay selectors* (MPR selectors).

The protocol employs an efficient link state packet forwarding mechanism called *multipoint relaying*. This mechanism is based on having each node select a subset of its neighbors such that this subset ensures connectivity to every two-hop neighbors. The nodes on this subset are called *multipoint relays* (MPRs) and the subset is the *multipoint relay set* (MPR set). Moreover, those neighbors that select a given node as their MPR are called the *MPR selector set* of the given node. The use of MPRs for control traffic transmission results in a scoped flooding instead of a full node-to-node flooding, thus inducing a reduction on the amount and volume of exchanged control traffic.

There are two main types of control messages in OLSR: HELLO and TC (Topology Control) messages.

1) HELLO messages are periodically broadcast by each node, containing the sender's identity and three lists: a list of neighbors from which control traffic has been heard (during a protocol defined time interval) but no bi-directionality has been confirmed, a list of neighbors with which bi-directionality has already been confirmed, and the MPR set of the originator node. These messages are only exchanged between neighboring nodes but they allow each node to have information about

one and two-hop neighbors; that information is later used in the selection of the MPR set.

2) TC messages are also emitted periodically by some nodes in the network. These messages are used for diffusing topological information to the entire network. A TC message contains the MPR selector set and a sequence number associated to the MPR selector set. Typically, not all nodes in the network are selected as MPRs, but all nodes must have a non-empty MPR set in order to communicate. Thus, the choice of sending the MPR selector set instead of the MPR set results in a reduction on the number of TC messages sent to the network. These TC messages provide each node with a global view of the network topology, to be later used in computing routes.

Each OLSR control message can be uniquely identified through a tuple consisting of its *originator identifier* and its *message sequence number*. A node may receive the same message several times; therefore, to avoid duplicate transmission and processing of control traffic, each node maintains a *duplicate set* where the unique identifier of each received message and a boolean value indicating whether the message has already been re-transmitted are stored during a protocol defined *holding time*. This mechanism is called the duplicate transmissions avoidance mechanism.

Through the exchange of OLSR control messages, each node stores the following information about the network. The available links to neighbor nodes are kept in the *link set*, and the neighbor nodes themselves are kept in four sets according to their nature: the one-hop neighbors in the *neighbor set*, the two-hop neighbors and the nodes which provide access to them in the *neighbor 2-hop set*, the chosen MPRs in the *MPR set* and the nodes which selected the current node as MPR of theirs in the *MPR selector set*. Nodes also keep information about the network topology gathered from TC messages; it is stored in the *topology set* in the form of tuples consisting mainly of a destination node identifier and an identifier of a last-hop to that destination.

In summary, the OLSR protocol can be specified as shown in Table I.

Notice that in a proactive routing protocol, each node has two tasks to accomplish [3]: (1) correctly generate the routing protocol control traffic (this way giving correct information to the other nodes on the network) and (2) correctly relay the routing protocol control traffic on behalf of other nodes (this way allowing for the control traffic to reach every node in the network). In its original specification, the OLSR protocol has the underlying assumption that all nodes comply in the exchange of crucial topology information through control traffic, which makes it vulnerable to several attacks.

Table II gives a taxonomy of OLSR security vulnerabilities[1] and provides examples of attack actions based on the network illustrated in Fig. 1.

[1]Notice that we do not consider, for example, the *jamming attack* in which an attacker saturates the medium by sending a large amount of messages, because those attacks result from the inherent characteristics of the communication medium and is independent of the employed routing protocol.

TABLE I
OPTIMIZED LINK STATE ROUTING OPERATION

1) Each node periodically broadcasts its HELLO messages;
2) These are received by all one-hop neighbors but are not relayed;
3) HELLO messages provide each node with knowledge about one and two-hop neighbors;
4) Using the information from HELLOs each node performs the selection of their MPR set;
5) The selected MPRs are declared in subsequent HELLO messages;
6) Using this information, each node can construct its MPR selector table with the nodes that selected it as a multipoint relay;
7) A TC message is sent periodically by each node and flooded in the network, declaring its MPR selector set;
8) Using the information of the various TC messages received, each node maintains a topology table which consists of entries with an identifier of a possible destination (a MPR selector in the TC message), an identifier of a last-hop node to that destination (the originator of the TC message) and a MPR selector set sequence number;
9) The topology table is then used by the routing table calculation algorithm to calculate the routing table at each node. Details about this procedure may be found in [1] and [2].

III. PREVIOUS WORK

A. Security Solutions for OLSR

Recently a number of contributions have provided partial solutions to OLSR security [3], [4], [5], [6]. In the following, we provide an overview of their main features, outlining the underlying assumptions and identifying a number of open issues.

Adjih *et al* present techniques [3] to counter a set of attacks on OLSR based on a mechanism for key distribution. Their proposal establishes a line of defense in which (i) nodes are either trusted or untrusted and (ii) trusted nodes are not compromised. Each routing control message is signed and time-stamped: the signature identifies messages from trusted nodes, and time-stamps prevent the replay of old messages. The approach does not address the following issues: (a) trusted nodes may behave incorrectly because of malfunctioning, unintentionally corrupting the routing protocol; (b) nodes in MANETs typically get in and out very often, thus it is hard to separate nodes into trusted and untrusted; (c) the signing mechanism is not detailed (a possibility is [6]).

Raffo *et al* consider the compromise of trusted nodes [6]. The authors assume that a public key infrastructure (PKI) and a time-stamp algorithm (e.g. the one in [3]) are in place. An additional message (ADVSIG) is sent in conjunction with routing control traffic. This message contains time-stamp and signature information. Each node has a so called *Certiproof* table where information received in ADVSIGs is kept. This information is then reused as a proof of correctness of the link state information in subsequent messages. The procedure ensures that a lone attacker node is not able to send wrong link state information to the network. Its drawbacks are as follows: (a) it does not protect against denial of service or wormhole

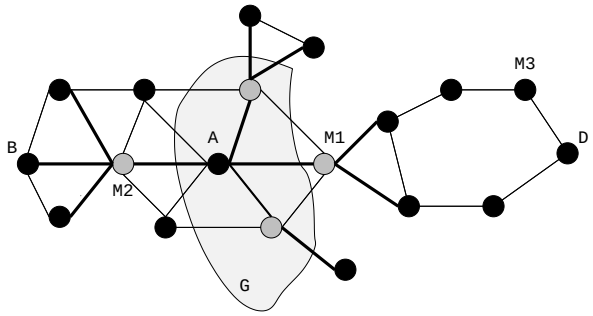

Fig. 1. Exemplary network topology for the OLSR protocol.

Nodes in gray are MPRs of node A; light edges represent the connections between nodes; dark edges identify the used links between A and all of its two-hop neighbors through the selected multipoint relay set. $M_{1,2,3}$ denotes the misbehaving nodes, D is the destination node and G defines a group of nodes.

TABLE II
TAXONOMY OF OLSR SECURITY VULNERABILITIES

ATTACK	METHOD	EXAMPLE	TARGET	RESULT
Identity spoofing	False HELLO	M_3 generates HELLOs pretending to be A	All nodes	MPR nodes of M_3 will present themselves as last-hop for node A, resulting in false route advertisements to node A
Link spoofing	False HELLO	M_1 generates HELLOs advertising bi-directional links to most of A's two-hop neighbors	Neighbor nodes	A chooses M_1 as its main MPR[4] which allows M_1 to intercept and modify most of A's traffic
	False TC	M_1 generates TCs advertising D as his MPR selector, directly to G[5]	Group of nodes	Distance between M_1 and D will be deemed to be one hop, thus M_1 will become the main bridge between G and D
Traffic relay/ generation refusal	Drop packets/ Blackhole	After becoming a preferential relay choice for A or G[6], M_1 drops packets received from them	Specific node Group of nodes	Loss of connectivity / Degradation of communications
	Refuse to generate control traffic	M_1 is selected as MPR for A and does not advertise that information to the network	Specific node	Node A unreachable, degradation of communications
Replay attacks	Traffic replay	M_1 sends to other nodes "old" previously transmitted[7] TC or HELLO messages	All kinds	Outdated, conflicting and/or wrong information enters the network which may cause defective routing
Wormhole	Protocol disobedience	M_2 and M_3 collude and exchange packets encapsulated, without the modifications presumed by the routing protocol	All kinds	The extraneous inexistent link M_2 - M_3 becomes a preferential choice for traffic and is fully controlled by M_2 and M_3

Examples presented are based on Fig. 1. ($M_{1,2,3}$ - misbehaving nodes, A - attacked node, D - destination node, G - group of nodes). [4] Because the smaller the MPR set is, the more efficient the OLSR results are; [5] M_1 is one hop away from G nodes; [6] It may use e.g. the described link spoofing techniques; [7] The messages can also be correctly authenticated.

attacks and (b) it imposes a large overhead to the network in terms of additional traffic and computation of signatures.

Based on the previous scheme, Adjih *et al* proposed a mechanism [5] to counter relay attacks based on the geographical position of nodes and a scheme that deals with compromised nodes based on *network flow conservation*, where misbehavior in traffic relaying is detected based upon the number of packets sent and received by each node. The drawbacks of this proposal are the following: (a) the weak assumption that forwarding the correct number of packets by a node proves that the packets were sent properly; and (b) a centralized security authority that manages misbehavior detection and remedy is difficult, if not impossible, to implement in a MANET.

Adjih *et al* continue their work focusing on key management techniques [4] providing a brief overview of methods to prevent wormhole and message replay attacks. The technique to prevent wormhole attacks is based on a variant of the counting technique in [5] in which nodes advertise a set of hashes of the packets received over each of the last k intervals. This way it is possible to check if packet losses cross a certain threshold, in which case a node is assumed to be compromised. Replay attacks are prevented as usual with time-stamps.

Dhillon *et al* propose a fully distributed certificate authority (DCA) based on threshold cryptography [7]. A node requests a certificate from any coalition of k nodes (shareholders) of the network. Each of the shareholders determines if it wants

to serve the request based on whether the requesting node is assumed to be behaving correctly. Upon receiving k "partial certificates" they are combined to generate a valid certificate as if it was signed by a regular CA. A monitoring system used to determine behavior of network nodes is not incorporated in the proposal.

B. Cooperation Aspects

Beyond the cryptographic schemes discussed in the previous paragraphs, current proposals for secure routing include cooperation enforcement mechanisms, which can be divided in two categories: currency-based mechanisms and reputation-based mechanisms. Currency-based mechanisms are based either on the exchange of virtual currency between nodes [8] or on the availability of a service which trades credits by receipts retrieved from messages in transit in the network [9]. In terms of reputation-based solutions, they are typically composed by three distinct mechanisms: (1) a local monitoring mechanism to observe the behavior of network nodes and determine their trustworthiness, (2) a reputation dissemination mechanism to convey other nodes with the results from the observations performed by the previous mechanism, and (3) a punishment/isolation mechanism to protect the network from misbehavior.

Nuglets are a virtual currency used to pay for packet forwarding services [8]. In the Packet Purse Model, the source node loads nuglets in the packet before sending it and each forwarding node acquires some of these nuglets as payment. In the Packet Trade Model each forwarding node buys the packet from the previous node by some nuglets and sells it to the following node for more nuglets. Both approaches rely on a tamper proof security module. The authors recognize that it is difficult to estimate the number of nuglets to send in the packet in order for it to get to the destination in the Packet Purse Model, and the Packet Trade Model allows overloading of the network because the sources are not bound to pay for sending packets. In a followup paper [10] the authors overcome the issue of the estimation of the amount of nuglets to send by using a counting technique where each node holds a nuglet counter that is decreased when a node sends an own packet and increased when he forwards packets on behalf of other nodes.

The *watchdog and pathrater* [11] are two extensions to the Dynamic Source Routing (DSR) protocol that attempt to detect and mitigate the effects of routing misbehavior. The watchdog is a mechanism for detecting misbehavior based on promiscuous monitoring of the next node in the path to detect if he correctly forwards packets sent to it. If a node bound to forward a packet fails to do so after a certain period of time, the watchdog increments a failure rating for that specific node and a node is considered as misbehaving when this failure rating exceeds a certain threshold. The pathrater then uses the gathered information to determine the best possible routes by avoiding misbehaving nodes. This mechanism, which does not punish these nodes (it actually relieves them from forwarding operations), provides increase in throughput of networks with misbehaving nodes.

CONFIDANT stands for Cooperation Of Nodes, Fairness in Dynamic Ad-hoc NeTworks [12]. It is an extension to DSR composed of four distinct mechanisms. The monitor mechanism detects deviations by listening to the transmissions of the next node in the path to detect relay refusal attacks. The trust manager is responsible for sending and receiving alarm messages, and for managing the trust given to received alarms according to the trust levels of the source nodes. The reputation system manages the ratings for the nodes in the network; they are modified accordingly to a rate function which assigns different weights to different types of misbehavior. The path manager participates in the route selection mechanism by deleting routes that contain nodes which have been classified with an intolerable rating, and takes measures to isolate misbehaving nodes. This protocol is subject to spreading of wrong accusations, which was addressed by the authors through the use of Bayesian statistics for classification and exclusion of liars.

CORE is a COllaborative REputation mechanism [13] to enforce node cooperation in MANETs. It is composed by a validation mechanism and a sophisticated reputation mechanism that considers three reputation types which are combined in a global reputation value. The validation mechanism monitors the execution of some operation by neighbor nodes. The subjective reputation is based on performed observations and avoids sporadic misbehavior by giving relevance to past observations in its calculation. The indirect reputation is based on the exchange of solely positive information provided by other nodes in the network. The functional reputation is based on the observation of different operational functions (e.g. routing and packet forwarding) combined in a global reputation value. This value determines each nodes' willingness to perform network operations on behalf of them.

In summary, the current security extensions to OLSR cover a sizeable number of distinct problems. Consensus seems to have been reached in the use of signature and key management systems to ensure the integrity and authenticate the sender of routing control traffic. Similarly, time-stamps have found full acceptance in the referred proposals dealing with the replay of old messages. Apart from the cryptographic security solutions required to guarantee authentication and integrity, it is essential to have mechanisms to enforce user cooperation by providing incentives to cooperate and/or punishing cooperation refusal. The solutions developed so far are basically of two kinds: currency-based solutions which depend on tamper proof components which may reduce their widespread applicability, and reputation-based solutions which rely on the ability to identify the nodes in the network.

IV. A Feedback Reputation Mechanism to Secure the Optimized Link State Routing Protocol

From the previous discussion, we conclude that mechanisms for securing and enforcing cooperation with routing protocols are of utmost relevance for the operation of MANETs. Based on a taxonomy of vulnerabilities and previous work

on securing the OLSR protocol, we identified two types of attacks for which there are commonly accepted solutions: (i) identity spoofing attacks can be tackled with signature and key management systems, and (ii) replay attacks can be addressed with a time-stamp mechanism.

In this paper we address the link spoofing attack, where a node announces fake links to nodes he cannot reach. This attack has the potential to cause increase in path lengths and the appearance of bottleneck nodes which can then be used to perform blackhole attacks or to partition the network.

To address this issue, we propose a feedback reputation mechanism that enforces the generation of proper routing control traffic by detecting and punishing misbehaving nodes. Although reputation mechanisms have already been proposed (see Section III-B), in practically all cases they have been applied to reactive routing protocols and rely solely on the watchdog as a monitoring mechanism. As acknowledged by different authors, a watchdog type of monitoring might not detect misbehaving nodes in case of (1) collisions, (2) limited transmission power, (3) collusion, and (4) partial packet dropping. Moreover, it only allows local detection of misbehavior and is therefore dependent on dissemination of alarms to declare misbehaving nodes. These alarms can be used to accuse legitimate nodes in a false way.

In contrast, our feedback reputation mechanism has the following features:

- It provides a new and reliable monitoring mechanism based on feedback messages which, at the cost of some bandwidth overhead, eliminates the drawbacks of the watchdog concept;
- It is able to detect and punish the generation of false routing control traffic (link spoofing attack);
- It includes a mechanism for widespread detection of misbehaving nodes without the need for dissemination of alarms that can be used for blacklist attacks, in which legitimate nodes are accused of misbehavior;
- It prevents blacklisting attacks as result of generation of fake feedback messages through the employment of the same type of mechanisms used to assure the integrity of the paths of on-demand routing protocols.

A. Attacker Model

We consider an active attacker. This attacker is a regular network node, and thus has access to the same routing information as all nodes in the network. It is able to inject routing information into the network that reaches neighbor nodes (through broadcast mechanisms) as well as every other node (through the supplied flooding mechanism). The intent of the attacker is to disrupt or adjust the routing protocol at will.

We assume that nodes are authenticated during communications (e.g. through a distribution of keys prior to communication, as suggested in [4]), thus being unable to impersonate other nodes or to use several pseudonyms for communications (Sybil attack). Moreover, replay attacks are prevented through the use of time-stamp mechanisms, such as those in [3] and [5].

B. Feedback Reputation Mechanism Description

The fundamental concern behind the feedback reputation mechanism is that of assuring that nodes correctly generate OLSR control traffic. To achieve this goal, the guiding principle is to reward nodes that comply with the routing protocol and penalize damaging behavior in terms of network availability [12], [13]; i.e. by reducing the ability for misbehaving nodes to communicate through the network.

For this purpose, we add two new elements to the regular OLSR operation:

- *Feedback Message:* A feedback message is used to convey the path traversed by a control traffic message through the network. Upon receipt of a TC message, according to the rules specified below, each MPR node sends a feedback message back to the originator of the TC, containing the path traversed by the TC message which, therefore, records the path traversed by itself as it traverses the network;
- *Rating Table:* Each node of the network keeps a rating table which holds information about the behavior of nodes in the network. Each entry in the rating table has a node ID, a primary and secondary ratings. The node ID uniquely identifies a node in the network, the secondary rating is a classification of the node based on the direct observation of packet retransmissions, and the primary rating is a more mature classification of the node based the correlation of its secondary rating, the analysis of information provided by feedback messages and local routing information kept by the nodes. In order to motivate nodes to behave well, these ratings are used to determine nodes' willingness to relay traffic on behalf of others, i.e. nodes relay most of the traffic for nodes with high ratings and refuse to do so for nodes with low ratings.

A security extension to the OLSR protocol that employs the proposed feedback reputation mechanism can be defined as shown in Table III. Note that steps 4–6, 9, and 11 belong to the regular OLSR operation while the remaining ones are introduced as parts of our security scheme.

The primary and secondary ratings vary from 0 to 100, 100 being the best possible value a node can attain. The initial primary rating ρ and secondary rating α at step 3 basically state the level of trust for each node. If we consider a network with complying nodes, we can set them to high values, otherwise, by setting them to lower values we are forcing the nodes to recover from a misbehavior state at the formation of the network. The feedback message rate λ at step 8 specifies the number of feedback messages that are generated in response to the reception of TC messages by nodes in the network.

As the TC messages traverse the network, they must keep the path they traverse in a way similar to that of on-demand routing protocols. This is performed by having each node add its identity to the path being accumulated in the TC before forwarding the message as usual.

Two mechanisms to build the reputation of nodes are used. The already mentioned watchdog mechanism which produces

TABLE III
FEEDBACK REPUTATION MECHANISM OPERATION

1) At the formation of the network, a signature and key management mechanism is employed, guarantying the proper authentication of each node;
2) During the broadcast of HELLO messages to ensure knowledge of one and two-hop neighbors, only properly authenticated nodes (through the signature mechanism) are considered;
3) For each authenticated node found, a new entry in the rating table is added with value α for the secondary rating and ρ for the primary rating;
4) Using the information from HELLOs, each node performs the selection of their MPR set, which is announced in subsequent HELLO messages;
5) Using this information, each node constructs its MPR selector set with the nodes that selected it as a MPR;
6) A TC is periodically flooded in the network by each node, declaring its MPR selector set;
7) A mechanism based on the already described watchdog concept is employed to detect misbehavior through direct observation of TC retransmissions;
8) Upon receipt of a TC message, a feedback message containing the path traversed by the TC message may be sent back to the origin depending on the rate λ of feedback message transmission;
9) Using the information of the TCs received, each node maintains a topology table which consists of entries with an identifier of a destination (a MPR selector in the TC message), an identifier of a last-hop node to that destination (the originator of the TC) and a MPR selector set sequence number;
10) When a feedback message is received, it is processed according to the Algorithm 1 for processing of feedback messages;
11) The topology table is then used by the routing table calculation algorithm to compute the routing table at each node. Details about this procedure may be found in [2].

changes in the secondary rating, and the feedback mechanism which produces changes in the primary rating.

C. Watchdog

The watchdog mechanism is based on having each node promiscuously hearing to its MPRs transmissions in the following way. When a node sends a TC message to the network it keeps listening to its MPRs transmissions. If a node detects that a MPR does not relay its packet, it decreases the MPR secondary rating by τ. Otherwise its secondary rating is increased by γ. To encourage cooperation in forwarding, the punishment should be greater that the reward.

As we have mentioned before, this mechanism is error-prone and, therefore, we confine it to produce changes in the secondary rating which, as will be shown later, is only used to determine how fast a node recovers from a misbehavior state.

D. Feedback Mechanism

The most relevant contribution of our work is the feedback mechanism. It is a reliable monitoring mechanism based on feedback messages generated in response to routing control traffic, which in the case of OLSR correspond to TC messages. When a feedback message is received it is processed as shown in Algorithm 1. The algorithm states that if a certain node is declared to have generated false routing information (step 3), then its primary rating is decreased by a punishment value PV (step 4). Otherwise, if the node was detected to have generated proper routing information its reputation rises (steps 6-9). Details about the mechanisms for detection of misbehavior (step 3), punishment of misbehaving nodes (step 4) and their recovery (steps 6-9) are subsequently presented.

1) Detection of false HELLO generation: The detection of false HELLO generation relies on the correlation of two sources of information: the paths obtained from feedback messages, and the local information obtained from HELLOs and kept in the *neighbor 2-hop set*. Since HELLO messages are only exchanged between direct neighbors and only the MPRs of a node relays its traffic, for this mechanism the nodes under scrutiny in Algorithm 1 are the MPRs of the current node.

Let us consider the scenario of Figure 2 in which node *C* generated a TC message and is now receiving a feedback message from one node in the network. Let *M* be a MPR of *C* which lies in the path of the feedback message (i.e. *M* was the forwarder of the TC from *C* which originated the current feedback message). The procedure to detect false HELLO generation is the following.

1) *C* receives a feedback message which holds the path of a TC message sent by him to the network;
2) *C* checks, for every node *T* two or more hops away from *M*, if there is an entry in the neighbor 2-hop set stating that *M* has direct connectivity to *T*;
3) If so, then *M* is a misbehaving node because he announced direct connectivity to *T* through HELLO messages and *T* is not directly reachable by *M*;
4) Otherwise *M* is considered a well-behaving node;
5) Taking in consideration if *M* is a misbehaving node or not, the reputation of *M* changes properly as shown in Algorithm 1.

There is one important issue with this approach. The local information kept by OLSR nodes is based on a periodic exchange of control traffic. With nodes moving, there are transient states where the actual network state and the local in-

Algorithm 1 Feedback message processing

1: $SR_s \leftarrow$secondary rating of the node under scrutiny, *S*
2: $PR_s \leftarrow$primary rating of the node under scrutiny, *S*
3: **if** mechanism for detection of false HELLO or false TC generation has identified *S* as misbehaving node **then**
4: $PR_S \leftarrow PV$
5: **else**
6: **if** $SR_S < PR_S$ **then**
7: $SR_S \leftarrow SR_S$ + SRV
8: **else**
9: $PR_S \leftarrow PR_s$ + PRV
10: **end if**
11: **end if**

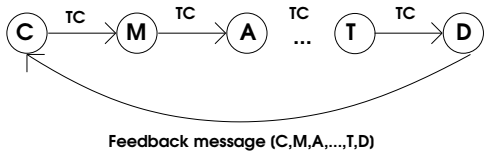

Fig. 2. Feedback message illustration

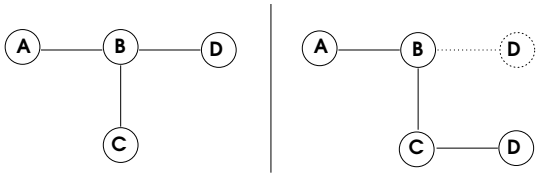

Fig. 3. MPR transient state

formation are not coherent. Regard, for example, the scenario of Fig. 3. Consider that on the left side, node B is a MPR of A, and C and D are MPRs of B. D is now moving and gets out of the transmission range of B and into the transmission range of C, becoming a MPR for C (right side of Fig. 3). In the meantime, the periodic exchange of control traffic did not occur and, therefore, A is still not aware of this topological change. A sends a TC to the network which follows the path A-B-C-D and D generates a feedback message containing this path. Since the local information of A states that B can reach D (because the local information of A was not updated yet), this results in a false positive of misbehavior detection where B is the misbehaving node.

For the scenarios considered in our simulations, one can see that these false positives are sparse and much less frequent than the correct detections of misbehavior. One possible solution to reduce them further, at the cost of more routing control traffic overhead, would be to decrease the intervals of control traffic generation. This results in a more frequent generation of control traffic which would ease a more up-to-date view of the actual network state and, subsequently, enable a reduction on the amount of false positives.

2) Detection of false TC generation: The detection of false TC generation is based on two sources of information: the paths obtained from feedback messages messages and the local information from TC messages kept in the *topology set*. Since TCs are flooded through all the nodes in the network, in this mechanism the nodes under scrutiny are all the nodes in the path of the feedback message. This allows us to detect nodes which have generated false TC messages through the following mechanism. Suppose that node *C* is a network node which is receiving a feedback message from the network. The procedure to detect false TC generation can be described as follows.

1) *C* receives a feedback message which holds the path of a TC message sent to the network by some node;
2) For every node *M* in the feedback message path and every node *T* three or more hops away from *M* also in the path, *C* checks if there is an entry in the *topology set* stating that *M* has direct connectivity to *T*;
3) If so, then *M* is a misbehaving node because he announced direct connectivity to *T* through TC messages and *T* is not directly reachable by *M*;
4) Otherwise *M* is considered a well-behaving node;
5) Taking into consideration whether *M* is a misbehaving node or not, the reputation of *M* is changed accordingly as shown in Algorithm 1.

The detection of false TC generation is also affected by the MPR transient state problem mentioned previously. One possible solution is to use the same technique of decreasing the intervals of control traffic generation at the cost of some more overhead. However, we have opted for a different approach, which avoids the increase in traffic overhead. Our approach to tackle this issue was already described in step 2 of the procedure to detect fake TC generation above and goes as follows. Instead of analyzing the connectivity of nodes which are two or more hops away from the eventual misbehaving node, we analyze it for nodes three or more hops away. This option successfully limits the number of false positives by reducing the number of occurrences of the MPR transient state, but it allows a misbehaving node to fake connections to nodes at two hops away. Nevertheless, we find this a reasonable compromise because of the very low number of false positives obtained and because, by faking connections to nodes at two hops away, a misbehaving node is only able to increase the path length by one, by faking connections to nodes at two hops away.

3) Punishment of misbehaving nodes: After detecting if a node is misbehaving, proper measures must be taken. As seen in step 4 of Algorithm 1, when a node is misbehaving its primary rating is set to a Punishment Value (PV). The primary rating ranges from 0 to 100, 100 being the best value for a node. In order to motivate nodes to behave well, the primary rating is used by the network nodes to determine their willingness to forward other nodes traffic. This is done by relaying other nodes traffic according to their primary rating. For instance, a node A that finds B to have a primary rating of 40 will only relay 40% of the packets from B.

4) Recovery of misbehaving nodes: The recovery mechanism allows a node that stops misbehaving to recover from the misbehavior state. This is where the secondary rating (which varies according to the watchdog mechanism) is used. This mechanism entails a *slow recovery* of nodes which are found to refuse relaying control traffic on behalf of other nodes. The overall procedure, described in steps 6-9 of Algorithm 1, goes as follows. If the secondary rating of the recovering node is lower than its primary rating, only the secondary rating is increased by SRV (Secondary Recovery Value), until it reaches the value of the primary rating. This buffer of time delays the recovery of nodes which have been refusing to relay control traffic because only once the secondary rating reaches a value larger than the primary rating the misbehaving node effectively starts recovering by having an increase of PRV (Primary Recovery Value) to the primary rating.

We call this mechanism *direct interaction recovery* since it is only active when nodes interact directly, i.e. only when a node is near another he is able to recover from a misbehavior state. The reason for this is that, from our simulation results we were able to see that the amount of feedback messages leading to a detection of good behavior is much larger that the amount

of feedback messages leading to detection of misbehavior and, therefore, we needed to restrict the recovery, otherwise misbehaving nodes would recover too fast and would not be properly punished.

While this approach based on direct interaction may not be well suited for every kind of network (e.g. if two nodes do not move and accuse each other, they will never be able to recover if they are not within range of one another) we do not deem this to be a problem. In fact, one can add other type of mechanisms that are not based on the proximity of the nodes, e.g. a timeout mechanism could allow nodes to recover after a reasonable fixed amount of time.

E. Discussion

The feedback mechanism is a monitoring mechanism that relies on the paths stored in TC messages (in a way similar to that of on-demand routing protocols such as Dynamic Source Routing). When the TC reaches a certain node, a feedback message is sent back according to the rate defined to convey the accumulated path to the source node of the TC. This information is then used to determine misbehaving nodes as explained previously. If not properly protected, the path information may be used to perform blacklisting attacks, where legitimate nodes are accused of misbehavior. To assure the validity and integrity of this information we can take the following measures:

- *Authentication checks*, where every node in the path checks the signature information introduced by the previous node to determine whether it has included itself in the path being stored in the TC;
- *Route tampering protection* to protect the integrity of routes stored in messages flooded to the network. This type of mechanisms have been widely investigated within the scope of on-demand routing protocols, and schemes to protect against the tampering of those routes such as [14] already exist.

Additionally, if the path diversity in the network is low, it may happen that a feedback message is sent back through a path that includes the misbehaving node itself. This would allow the misbehaving node to drop the packet and that information to be lost, thus reducing the resulting punishment. One way to overcome this issue is the following.

1) Modify a redundancy parameter of OLSR (TC_REDUNDANCY) such that nodes advertise more neighbors than only the MPR selector set in TC messages (without considerable increased overhead because the TCs would still benefit from MPR forwarding);
2) With knowledge of an increase set of links in the network, each node can select alternative routes to send the feedback messages (e.g. by performing source routing for these specific messages).

V. Simulation Results and Discussion

The simulations presented in this paper where performed using the network simulator ns2 version 2.29.2 with a modified version of the UM-OLSR [15] implementation version 0.8.8 of OLSR (the used code is available in [16]). All the default values for the OLSR protocol from RFC3626 were used. The simulations were performed for 30 nodes with a transmission range of 250 meters, in an area of size 1500x300 meters during 900 seconds. The Random Waypoint Mobility model was used and mobility patterns were generated using the tool from [17] which provides a steady-state node mobility throughout a simulation. To average the results and diminish the choice of a favorable or unfavorable pick of scenarios, 5 independent replications were run, each with a set of 10 distinct mobility scenarios, which results in a total of 50 simulation runs for each set of parameters under evaluation. To exercise a network with walking mobile nodes, we considered the node speeds of 1.4m/s and 2.4m/s. Moreover, pause means of 1 and 5 seconds were also tested.

A. Attacker

The attacker, as implemented, performs two types of attacks: generation of false HELLOs and generation of false TCs.

For the generation of false HELLO, the attacker node adds the false information that he is able to reach all of his two-hop neighbors with the intent of forcing the selection as MPR. This attack may be harmful in two ways: (a) it can cause the selection of a wrong MPR set and (b) messages sent by the attacked node may not reach some of his two-hop neighbors. From our simulations this attack has shown not to be very effective in forcing the selection as a MPR, therefore to exercise a successful attacker we have set an OLSR flag on the attacker node which forces its selection as a MPR node.

For the generation of false TC, the attacker node randomly chooses a node which is at three or more hops away from him and announces direct connectivity to it. This attack may be harmful because it introduces conflicting routes and promotes loss of connectivity and increase in path lengths in the network.

Both types of attackers and the corresponding detection mechanisms were tested separately. In our simulations the attacker starts generating false control traffic after 50 seconds of the beginning of the simulation and restarts behaving correctly at 300 seconds.

B. Feedback Reputation Mechanism Parameters

Since the goal of the feedback reputation mechanism is to properly punish the generation of false routing control traffic, independently of whether a node refuses to relay traffic, the parameters related to traffic relay refusal were set to the default values of $SRV = 1$ (secondary rating recovery value), $\gamma = 1$ (secondary rating increase) and $\tau = 2$ (secondary rating decrease). This resulted, as expected, in very high secondary ratings because no traffic relay refuser was used.

As for the remaining parameters, the initial values of the primary ρ and secondary α node ratings where both set to the top value of 100, in other words we consider the nodes in the network to be honest. Since it is hard to choose an adequate value for the feedback message rate λ, we performed

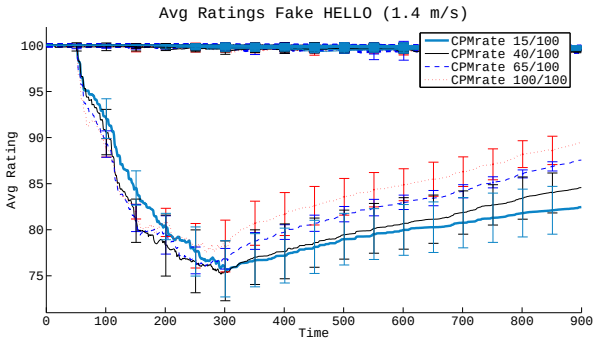

Fig. 4. Average rating of nodes (false HELLO, 1.4m/s)

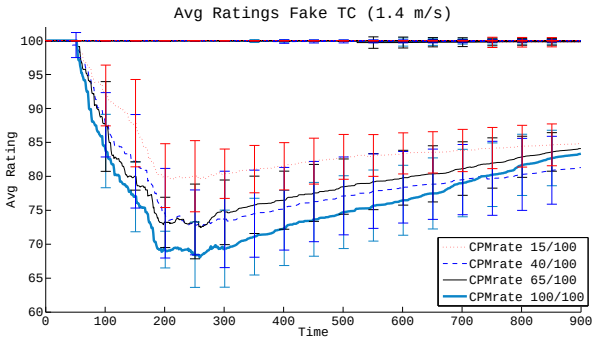

Fig. 5. Average rating of nodes (false TC, 1.4 m/s, 1 false link)

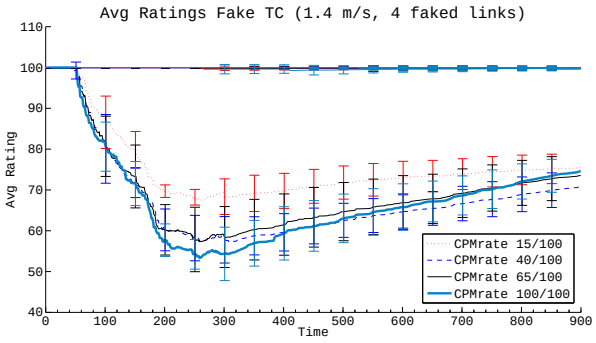

Fig. 6. Average rating of nodes (false TC, 1.4 m/s, 4 false links)

several sets of simulations and analyzed the results for several feedback message rates. The results are shown in Section V-C.

The punishment value PV and the primary recovery value PRV must be set in correlation in order to allow a correct punishment of misbehaving nodes but also a reasonable recovery for nodes that start behaving correctly after misbehavior. Our simulations show that the number of false positives are more frequent in the detection of false HELLO than in the detection of false TC, therefore we used a more severe punishment value $PV = 0$ for false TC detection and a less severe $PV = primary\_rating/2$ for the detection of false HELLO messages. Regarding the primary recovery value, a value of $PRV = 1$ has delivered very satisfactory results in terms of punishment vs. recovery of the nodes. For both PV and PRV, setting them to higher values will allow better recovery but worse punishment, and vice-versa.

C. Results

In this section, we discuss a set of simulation results underlining the effectiveness of our security scheme and the cost in terms of traffic overhead. Two type of plots are shown: plots with the average ratings of the nodes and plots with the overhead induced by the feedback message and OLSR operation.

The plots with the average ratings of the nodes show the ratings for all the nodes in the network. The lines on top correspond to the average ratings of all the well-behaved nodes and the lines in the middle correspond to the average rating of the misbehaving node, for all the feedback message rates considered. The average rating *R* of a certain node *A* tells us that, if the traffic in the network is evenly distributed, the punishment mechanism will allow, in average, *R* % of the traffic originated in *A* to be delivered to the next destination.

The overhead plots basically allow a comparison of the overhead of the feedback mechanism introduced by our security scheme and the overhead of the regular OLSR operation.

From the plots in Figs. 4 we can see that the behavior of the mechanism for detection of false HELLO does not significantly change with the variations in the feedback message rates. What does change is the recovery mechanism which is faster for higher values of feedback message rate.

As of the detection of false TC, from Fig. 5, we can see that this mechanism is already more subject to changes in the feedback message rate used. For both node speeds tested, the average rating of the misbehaving node drops faster and to a lower bottom value for higher feedback message rates, and the recovery mechanism is once again much faster for higher feedback message rates. The results for the speed of 2.4 m/s are omitted because they are very similar to these ones.

It may seem that these ratings could be more severe, although it is important to notice that the average ratings presented consider the nodes from the whole network therefore eventually including nodes with which the misbehaving node does not interact (e.g. because they do not become MPRs and, therefore, do not relay traffic, which results in keeping high ratings for misbehaving nodes). Additionally, for the detection of false TC, the tests performed consider an attacker that announces a single fake link. As the number of fake links increases, the average primary ratings drop even further. See for example Fig. 6 where with 4 fake links the primary ratings drop to lower values than in the previous plots, reaching a minimum around 55 points for a feedback message rate of 100%.

In all the plots shown so far, one second was used for the mean pause value. From the results on Fig. 7, where a mean pause of 5 seconds was used, we can see that (1) the misbehaving node is more severely punished, and (2) the well-behaved nodes have slightly worse average ratings. This is in line with the fact that with larger pause times, nodes will naturally interact less with each other and, therefore, the recovery mechanism (which is based on direct interaction) will be less effective. This fact resulted in a sharp change in the evolution of the average primary rating when (at 300s) the

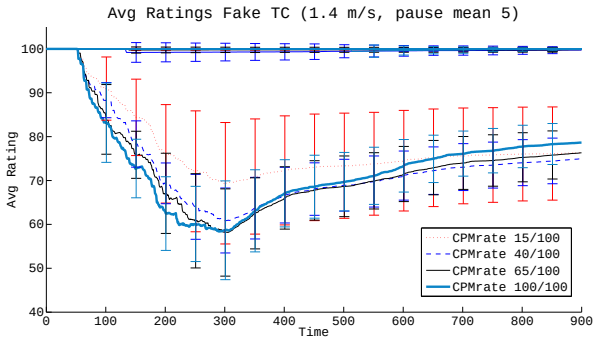

Fig. 7. Average rating of nodes (false TC, 1.4 m/s, 1 false link, pause mean 5s)

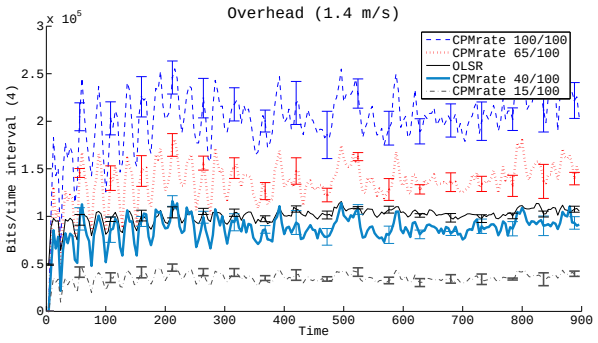

Fig. 8. Overhead of Feedback Mechanism vs OLSR (1.4 m/s)

attacker node stops misbehaving. This behavior was less clear in the previous plots with one second mean pause, because a large amount of interactions between nodes increases the impact of the recovery mechanism on the ratings.

In terms of the overhead results presented in Fig. 8, as expected there is a high overhead of our security scheme if a feedback message rate of 100% is used and, naturally, as the feedback message rate gets lower so does the overhead, becoming very reduced in the case of a feedback message rate of 15%. The results for the node speed of 2.4 m/s are omitted because they are very similar to these ones.

In summary, since the mechanism for detection of false HELLO behaves arguably well for the considered rates, we believe that a feedback message rate of 15% would be the wisest choice. In terms of the mechanism for detection of false TC, a feedback message rate in between 15% and 40% would provide reasonable punishment for the misbehaving node with a rating around 75-80 for the weakest attacker (a single fake link) while keeping a reasonably low overhead in terms of network traffic.

VI. Conclusions

We presented a feedback reputation mechanism for securing the OLSR protocol. The mechanism deals with the generation of fake HELLO and fake TC messages, two attacks which so far have not had a satisfactory solution. Beyond providing a natural solution for these security problems, our practical scheme is resistant to general problems of reputation systems. Specifically, we are able to eliminate the dissemination of reputation information throughout the network, and make it impossible for nodes to accuse or praise other nodes falsely — this would require them either to generate false feedback messages (which can be protected by cryptographic mechanisms) or to repeat old feedback messages (which are protected by a time-stamp mechanism). As part of our ongoing work we are studying how to better tackle the false positives obtained in the detections of misbehavior (so that all well behaving nodes are guaranteed to maintain maximum ratings at all times), how to develop more effective recovery mechanisms, and how to counter the traffic relay refusal attack.

References

[1] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Vennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. of IEEE International Multitopic Conference (INMIC 2001)*, 2001.

[2] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr), rfc 3626," 2003, October 2003.

[3] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler, and D. Raffo, "Securing the OLSR protocol," in *Proceedings of Med-Hoc-Net*, Mahdia, Tunisia, June 2003.

[4] C. Adjih, D. Raffo, and P. Mülethaler, "Attacks against OLSR: Distributed key management for security," in *2005 OLSR Interop and Workshop*, Ecole Polytechnique, Palaiseau, France, July 28–29 2005.

[5] C. Adjih, T. Clausen, A. Laouiti, P. Mühlethaler, and D. Raffo, "Securing the OLSR routing protocol with or without compromised nodes in the network," HIPERCOM Project, INRIA Rocquencourt, Tech. Rep. INRIA RR-5494, February 2005.

[6] D. Raffo, C. Adjih, T. Clausen, and P. Mühlethaler, "An advanced signature system for OLSR," in *SASN '04: Proceedings of the 2nd ACM Workshop on security of ad hoc and sensor networks*. New York, NY, USA: ACM Press, 2004, pp. 10–16.

[7] D. Dhillon, T. S. Randhawa, M. Wang, and L. Lamont, "Implementing a fully distributed Certificate Autorithy in an OLSR MANET," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Atlanta, Georgia, USA, March 21–25 2004.

[8] L. Buttyán and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc wans," in *MobiHoc '00: Proceedings of the 1st ACM international symposium on mobile ad hoc networking & computing*. Piscataway, NJ, USA: IEEE Press, 2000, pp. 87–96.

[9] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *INFOCOM*, 2003.

[10] L. Buttyán and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, 2003.

[11] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 255–265.

[12] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2002, pp. 226–236.

[13] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. of the IFIP-Communication and Multimedia Security Conference*, Copenhagen, June 2002.

[14] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Wireless Networks*, vol. 11, no. 1, pp. 21–38, 2005.

[15] F. J. Ros, "UM-OLSR," obtain via: http://masimum.dif.um.es/.

[16] J. P. Vilela, "CSS-OLSR - Code for simulations," obtain via: http://www.dcc.fc.up.pt/~joaovilela/css-olsr/.

[17] S. PalChaudhuri, J.-Y. L. Boudec, and M. Vojnovic, "Perfect simulations for random trip mobility models," in *Annual Simulation Symposium*. IEEE Computer Society, 2005, pp. 72–79. [Online]. Available: http://dx.doi.org/10.1109/ANSS.2005.33