

# SAFE: Securing pAcket Forwarding in ad hoc nEtworks

Yacine Rebahi, Vicente .E Mujica-V, Cyprien Simons and Dorgham Sisalem  
Fraunhofer Fokus, Kaiserin Augusta Allee 31, 10589 Berlin, Germany  
{rebahi, mujica, simons, sisalem}@fokus.fraunhofer.de

**Abstract**—Ad hoc networks are a new networking paradigm characterized by a dynamic topology and the absence of a predefined infrastructure. As a consequence, ad hoc networks are more vulnerable to attacks than traditional networks. The mechanisms used so far for protecting these networks are authentication and encryption. The latter, however, seem to be inefficient against malicious packet dropping and denial of service attacks. This paper addresses the malicious packet dropping problem and describes a mechanism to deal with. In fact, a trust model based on the reputation concept is developed. This mechanism provides two main functionalities: monitoring the behaviour of the neighboring nodes in the network and computing their reputations values based on the information provided by the monitoring. This paper also discusses in details how the reputation information is managed within the network. The proposed mechanism is also validated with some simulation work showing its feasibility and performance.

## I. INTRODUCTION

Ad hoc networks represent a new networking paradigm where communications are established between devices without the support of a centralized infrastructure. This new paradigm provides an interesting complementary (if not an alternative) to the traditional networks, especially that devices can easily be added or removed and the network will smartly reconfigure itself after any change. Unfortunately, the lack of infrastructure and the open accessibility of the ad hoc networks have made them more vulnerable to attacks than wired networks. Usually security in ad hoc networks is achieved through authentication and encryption. These techniques could only be considered as a first line of defense as they cannot for instance prevent a malicious node that has already joined the network to drop packets in order to carry out some denial of service attacks. This problem has led to the development of reputation systems to detect malicious nodes and discard them from the network.

In this work, we will be dealing with two kinds of misbehaviours: malicious packet dropping and false accusation propagation. In ad hoc networks, each node is assumed to be a router for the others, however, these nodes usually have limited capabilities which are considered as an excuse by some of them to behave in a selfishly way by using the resources of the other nodes and without offering any themselves. In fact, selfish nodes rely on the other nodes in the network to forward their packets, however, they only forward a certain amount of the received packets in order to save their resources. Malicious packet dropping may also be carried

out to disrupt the functioning of the network and degrade its performance. Some other malicious nodes may not drop packets, however, they will send false accusation to isolate some other nodes in order to disrupt the network performance.

In this paper, we develop a reputation system that detects misbehaving nodes and discard them from the network. Our mechanism builds trust between the nodes in the network by stimulating collaboration between them and discourage the malicious ones to carry out attacks and fraud. In our scheme, each node in the network, monitors the behaviour of its neighborhood and if a misbehavior is detected, the other neighboring nodes are informed in order to help them in protecting themselves.

Using reputation mechanisms to secure packet forwarding was proposed by many authors. For instance, in [1] and [2], failed and selfish behaviours are studied and the reputation here is simply bound to how "good routers" the nodes are. To be more precise, a good node refers to a node that forwards packets to the next node in the path even it has no interest in this transaction. Based on the number of successful forwarding transactions, a reputation value is assigned.

Two other reputation systems that work in the same way have also been proposed to help protect packet forwarding in ad hoc networks: the Collaborative Reputation (CORE) mechanism [3] and the Fairness in Dynamic Ad Hoc Networks (CONFIDANT) protocol [4].

The CORE scheme uses global reputation values that range from the value +1 through the value 0 to the value -1. This mechanism places more weight on past observations which may tolerate sporadically bad behaviours. In this case, a malicious node can build a good reputation, becomes a bottleneck node and then starts behaving maliciously which leaves in this case a deep impact on the network performance.

The CONFIDANT mechanism extends reactive routing protocols and uses only negative values for computing reputation. The CONFIDANT scheme uses Alarm messages to inform the other nodes in the network about a misbehaviour that occurred. This information is sent to a "Friends" list. This is restrictive and complex as a node's "friends" need to be determined and the corresponding information has

to be maintained. CONFIDANT allows malicious nodes recovery through timeouts when their entries in the black lists expire and are deleted. This way of allowing nodes recovery introduces a new vulnerability which is enabling malicious nodes to join again the network and repeat their attacks.

As for SAFE, it also extends reactive routing protocols. This mechanism uses positive reputation values varying between 0 and 1. This allows to describe in an easy way the proportions of packets forwarded by each node in the network. Contrary to CORE, the SAFE mechanism places more weight on the most recent observations, but without neglecting past observations. This way of computing reputation provides a more fair and seamless way in rating the other nodes. SAFE also allows misbehaviour information exchange between neighboring nodes without restriction, which provide a more robust mechanism for protecting the entire network. Our mechanism also prones nodes recovery, however, when the recovering nodes join again the network, they will be assigned “critical” reputation values which force them to behave correctly, otherwise they will be discarded again from the network.

This paper is organized as follows: section II describes the SAFE mechanism, its components as well as its functionalities. This section presents in particular how the reputation information is gathered, stored and evaluated. A simulation work validating the proposed scheme and showing its performance is described in section IV. Next, Section V describes some issues that can be addressed in the future and finally, section VI concludes the paper.

## II. THE SAFE SCHEME

### A. Overview

Establishing trust in ad hoc networks needs to pass through detecting intruders and discard them from the network. However, if we rely only on self-detecting misbehaviors, this could be insufficient because a node currently not detecting any fraud, could not be sure that all its one-hop neighborhood are trustful. Indeed, a node that is actually not sending packets cannot detect selfish nodes in its neighborhood. As a consequence, collaboration between neighboring nodes is mandatory to provide a global mechanism for protecting the entire network. For achieving that, each node in the network monitors the packet forwarding of its neighborhood and upon detecting a misbehavior from one of them, it broadcasts this information to the others. The reputation values that each node computes is based on the results of the undertaken monitoring and will express the amount of trust that this node has in the ones that it is relying on to forward further its packets. The mentioned reputation information broadcast will be referred along this paper as sending an accusation. Also, a node A neighborhood will refer to the nodes that are only one hop away from node A. This simply means that accusations are sent only one hop away and the reputation information is

gathered from the nodes that are only one hop away from the requester node as it will be discussed in section III-A

Building trust between nodes in an ad hoc network requires that these nodes possess some correct identities valid over a reasonable period of time. The more resistant to spoofing these identities are, the more robust the reputation system is. Our mechanism simply uses IP addresses to identify the nodes and can be used in concert with any other mechanism, for instance the Secure Routing protocol (SRP) [5], for dealing with IP spoofing.

In ad hoc networks, packet dropping has various reasons behind it. A node may drop packets to save its resources or to carry out a denial of service attack. However, it may also happen that a node drops packets because it simply does not have enough resource to do so. SAFE considers the excessive packet dropping as a misbehaviour regardless the intension behind. A malicious node in this paper means that this node was assigned a reputation value less than a the threshold defined in section III-B. These reputation values are tightly bound to the number of packets dropped by a node. Thus, when a threshold is reached, the packet dropping becomes a serious threat and the node responsible for this dropping must be avoided.

SAFE also advocates smart collaboration between the nodes in the network. Suitable methods for implementing such collaboration constitute epidemic algorithms [6]. The latter transmit information when nodes get in direct contact, similar to the transmission of an infectious disease between individuals. In fact, when a misbehaviour regarding a node A is detected, this information is broadcasted to the neighboring nodes, which will store it for a period of time. If it happens that these nodes move to another network and this misbehaviour information is needed for some reasons such as the recent joining of node A to this new network, this information can be used to help in detecting and discarding node A. The way the reputation repositories (see section II-B) are built in SAFE, are also based on some methods used in the epidemic algorithms [6], which makes them more suitable for highly dynamic networks.

### B. Building Blocks

SAFE builds trust through an entity, called the SAFE Agent, that runs on each node in the ad hoc network. Each node is in charge of detecting local misbehaviours independently. However, collaboration between all the nodes is required in order to secure the entire network. The SAFE Agent architecture is depicted in figure 1 and comprises the following functionalities,

*The Monitor:* The Monitor is in charge of observing the node's neighbourhood packet emission. For instance, if node A is monitoring the behaviour of node B, node A will keep a ratio of the number of packets that node B has forwarded

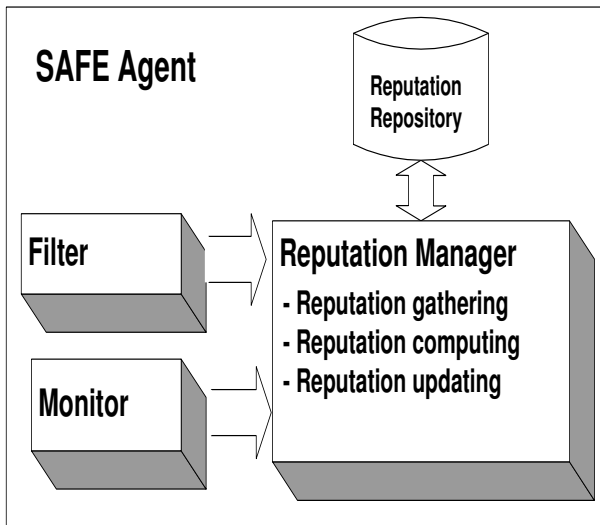


Fig. 1. Trust Management Architecture

and the total number of packets that node A has transmitted to node B to forward them further. The monitoring results are regularly communicated to the Reputation Manager, which will update the reputation value of the node being monitored and store it in the reputation repository.

*The Filter:* As it will be described in section III-A, the SAFE mechanism adds a reputation header to the underlying routing protocol in order to facilitate the exchange of the reputation information between the different SAFE agents. The Filter can be considered as a complementary module that helps in distinguishing the packets containing some reputation information from the rest of the received packets. In fact, when a packet is received by a node, it passes first through the Filter that checks whether that packet involves a reputation header. If it is the case, the packet will be forwarded to the Reputation Manager. Otherwise, it will be processed as a normal packet.

*The Reputation Manager:* The Reputation Manager is the main component of the SAFE Agent. This module is in general responsible for the manipulation of the reputation information. To be more precise, the Reputation Manager gathers, computes and maintains the reputation information related to the neighbourhood. The Reputation Manager takes input from either the Monitor or the Filter. The first case reflects that a certain node packet emission is being monitored, so, the Reputation Manager must be contacted to update the corresponding reputation information. However, the input received from the Filter is related to some reputation information exchange between some neighbouring nodes regarding a malicious/accused node (see section III-A for more details). The Reputation Manager computes the new reputation value of the node under consideration using the metrics described in section III-B and stores this value in its reputation repository.

*The Reputation Repository:* SAFE assumes that each node in the network maintains a Reputation Repository where the neighbouring reputation values computed according to the metrics described in section III-B are stored. These repositories are filled by the reputation values computed through a direct monitoring or through the accusations sent by some nodes in the network. To be more concrete, SAFE stores the reputation information in form of (key, value) pairs. The key field refers to the IP address of the node that the reputation is being computed for, however, the value field refers to a vector comprising the REP\_VAL which is the reputation value of the node being monitored and the TTL which is the Time To Live value. The TTL indicates the period of time for which the entry is valid and when it expires the entry is automatically removed from the Reputation repository. The TTL is used for two purposes.

- Limit the size of the repositories especially that some nodes may leave the network and do not join again for a long time, so it is useless to keep their entries in the reputation repositories if these entries are not used
- The nodes that are discarded from the network because they excessively dropped packets, whatever the reason behind it, can recover and join again the network. The nodes that failed to forward packets because of physical problems will join again the network and participate actively in the packet forwarding operation. The malicious nodes can also join the network after being discarded. However, every node that joins the network will be assigned a reputation value close to a predefined threshold as described in section III-B. This will force the malicious nodes to behave correctly, otherwise their reputation will get down and reach the mentioned threshold very quickly and thus, they will be discarded again from the network

The way the reputation repositories are built in SAFE is similar to some mechanisms used in epidemic algorithms [6]. This helps the nodes to react faster when exchanging reputation information in highly dynamic ad hoc networks.

### III. THE PROTOCOL DESCRIPTION

Although the SAFE scheme introduces some new features that can be incorporated easily with the underlying routing protocol, this scheme still remains lightweight and can be integrated further with a protocol such as the Secure Routing Protocol (SRP) [5] to deal with secure route discovery. In this case, the network is more secure as it is enabled to detect a wider set of attacks.

#### A. Reputation Information Exchange

SAFE provides a novel way of exchanging reputation information between neighbouring nodes in the network. This exchange is achieved through the SAFE Header depicted in

the figure below.

REP_TYPE	ACCUSED_ID	REP_VAL
----------	------------	---------

The reputation information exchange occurs in two situations: First, when a node detects a misbehaviour from another node through a direct monitoring. In this case, the node detecting the fraud has to broadcast this information to its neighborhood in order to help them in protecting themselves. The second situation occurs when a node A receives an accusation about a node B from another node in their neighborhood. Node A will not directly update the reputation value of node B with the information received from the accuser, however, it will query the other nodes in the neighborhood for their opinions regarding the accused node. Based on the received information, the reputation value of the suspicious node will be updated. The mentioned features introduced by SAFE require the addition of a header comprising the fields depicted in the figure above. The SAFE Header is incorporated into the underlying protocol header as an additional IP option or as separate header inserted after the IP header and before the next upper-layer protocol. The SAFE Header consists of three values; the reputation type, the IP address of the node whose reputation is being reported and the reputation value itself. Below, a description of these different values is provided.

- **REP\_TYPE:** Different SAFE messages are distinguished with the help of the REP\_TYPE field. The latter takes one of the following values,
  - **REP\_ACCUSATION:** this value is used when a node detects a misbehavior and wishes to share this information with its neighborhood
  - **REP\_REQUEST:** this value is used when a node receives an accusation regarding another node and wishes to query its neighboring nodes about their opinions
  - **REP\_RESPONSE:** when a node receives a SAFE message requesting an opinion about a certain node, the value REP\_RESPONSE is used to reply to this kind of requests
- **ACCUSED\_ID:** this field contains the IP address of the accused node
- **REP\_VALUE:** this field contains the reputation value of the accused node

### B. Reputation Evaluation

The reputation values used in this work, are defined to be real numbers ranging from 0 to 1. In the sequel, we will distinguish between the direct reputation value, denoted by  $r_{direct}$  and which is the outcome of the packet forwarding monitoring, and the residual reputation value, which is a weighted sum of  $r_{direct}$  and the reputation value already

existing in the reputation repository (denoted by  $r_{reptab}$ ) of the node being performing the rating. The direct reputation values are tightly linked to the number of packets forwarded during a transaction. For instance, if during a given period of time, a node A has received 10 packets, however, it forwarded only 7, this node will be assigned the direct reputation value 0,7. The reputation of a node varies according to its behaviour, it may increase or decrease depending on the number of packets forwarded.

#### Initial reputation value

If two nodes are in the transmission range of each other, just after the discovery phase, each of them creates an entry in its reputation repository for the other and assigns it the reputation value  $\lambda_{init}$ . This value is slightly above a threshold value  $\lambda_{thre}$ . The latter simply expresses that if a certain node has a reputation less than this value, this node is declared to be malicious. To be more concrete, if the threshold value is 0,8, the initial reputation value will be for instance 0,82.

#### Reputation computing using direct monitoring

Monitoring the packets emission of a node is achieved according to the number of packets that this node has forwarded. In other words, if node A is relying on node B for forwarding its packets, node A computes node B reputation using the following metric,

$$r_{direct}(A, B) = \frac{\#forwarded}{\#sent} \quad (1)$$

The reputation value resulting from the formula (1) will be combined with the reputation value already existing in the reputation repository of node A regarding node B ( $r_{reptab}(A, B)$ ). This can be achieved through the following formula,

$$r(A, B) = (1 - a) \cdot r_{reptab}(A, B) + a \cdot r_{direct}(A, B) \quad (2)$$

$a$  is a value ranging between 0 and 1. This formula is a weighted sum consisting of two parts. The first part describes the node B reputation value already figuring in the node A's reputation repository. If node A did not meet node B before, the reputation value  $r_{reptab}(A, B)$  is set to the value  $\lambda_{init}$  as mentioned earlier. The second part reflects that some changes regarding node B reputation occurred and the corresponding reputation value needs to be updated. So node A computes the new node B reputation value and adds it to the previous one ( $r_{reptab}(A, B)$ ), in the way described in formula (2), before storing the sum again in its reputation repository. By taking into account a node reputation history, the evaluation will be consistent and seamless. Indeed, a "good" router, which met a physical problem for a short time, will not be punished and discarded as its reputation is going to increase again if we still rely on it for forwarding data packets. On the other hand, a node reputation should seamlessly vary. If the reputation, for instance, ripples, discovering a new routing path is frequently

invoked and the node power is quickly consumed. Despite the contribution of the historical reputation, the most recent reputation value will always be considered more "heavy" by setting the weight  $a$  to a sufficiently big value.

#### *Reputation computing using accusations*

In this context, accusations refer to the reputation information that a node in the network has broadcasted regarding another node. These accusations may be true if the sender has really detected a misbehaviour and wished to share this information with its neighbourhood. However, sending false accusations can also be used for disrupting the network performance. A malicious node, instead of dropping packets, may send false accusations in order to discard some nodes from the network and subsequently decrease its performance. To prevent a malicious node to fetch a false accusation, the SAFE mechanism introduces the following measures.

Accusations from malicious nodes are ignored. This means that an accusation sent by a node that has a reputation value less than the threshold is ignored

If a node A receives an accusation regarding node B, node A will first query its neighbourhood for their own opinions regarding the accused node. If the queried nodes have any information regarding the accused node, they will send it to node A. Here also an accusation number threshold  $\theta_{accu}$  is introduced. This means that if the number of accusations received by node A is less than the threshold value  $\theta_{accu}$ , the accusation against node B will be ignored, otherwise, the reputation of node B will be updated. To be more concrete, let us assume that  $p$  ( $p > \theta_{accu}$ ) accusations regarding node B have been received by node A from the nodes  $N_1, \dots, N_p$ . In this case, the node B reputation value will be updated as follows,

$$r(A, B) = \tau \cdot r_{reptab}(A, B) + (1 - \tau) \cdot \frac{\sum_{i=1}^p r_{reptab}(A, N_i) \cdot r_{reptab}(N_i, B)}{\sum_{i=1}^p r_{reptab}(A, N_i)} \quad (3)$$

Let us note that node A treats the received replies based on its trust in the sender nodes. This forces the nodes in the network to behave correctly as the more trustful the node is, the more valuable its reply is. The historical reputation is again considered here through the value  $r_{reptab}(A, B)$ . However, the weight  $\tau$  will be chosen small enough to make the currently computed reputation more "heavy".

#### IV. SIMULATION

The network simulator *ns-2* [9] was used to simulate the SAFE Agent architecture described in section II-B. The network simulator *ns-2* is an object-oriented and discrete event-drive network simulator that, in the recent years, has incorporated powerful tools, protocols and modules in the area of ad hoc networks. Considering the *ns-2* simulator is not a restriction as the modular architecture of the SAFE

Agent allows the use of any other network simulator.

Each mobile host has an omni-directional antenna having unity gain. The wireless interface (WaveLan) is modeled as a shared-media radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 m [10]. Each node is assumed to have a buffer of size 64 packets. The simulation study was carried out by considering the "pause time" parameter, which represents the agent mobility in ad hoc networks. For instance, when the pause time increases, the nodes tend to remain stationary. Otherwise, nodes are frequently in motion with low pause times (i.e full motion when the pause time is 0). The random waypoint model [11] is selected as the mobility model in a rectangular field (600 x 300 meters). The movement scenario files are characterized by the pause time (PT) parameter which varies from 0 through 20, 30, 50, 100, 200, 300, 400 to 500 seconds. The nodes' speed is uniformly distributed between 0 and a maximum value of  $10m.s^{-1}$ .

To evaluate the performance of the SAFE mechanism, 15 total nodes are used in the rectangular field, where source and destination nodes are located in opposite sides and one malicious node is in the middle of the transmission. The rest of the nodes move randomly in the field. The source node transmits packets with a sending rate of 4 pkts/s and a packet size of 64 bytes. The Dynamic Source Routing (DSR) [12] is used as the underlying protocol because of its maturity and efficiency. Simulations were run during 900 seconds taking into account the above pause times.

The malicious node is created using a two-stage Markov chain machine. In the "good" stage, the nodes behave without dropping any arriving packet. However, in the "bad" stage, malicious nodes drop packets based on a "dropping" function. The latter is defined as a random number between a maximum (MAX\_RATE) and a minimum (MIN\_RATE) dropping rates. The Markov chain machine oscillates between both states during a period of time ( $t_{trans}$ ) that can be a fixed or a random value.

We investigate in this simulation work the packet delivery ratio, the routing overhead, the throughput and the average end-to-end delay (See [10], [11] for more details). The packet delivery ratio represents the fraction of data packets delivered to the destination. On the other side, the total number of the routing packets transmitted during the simulation are measured to compute the routing overhead. In this case, each forwarded packet counts as one transmission. The Throughput refers to the actual measured performance of a system when the delay is considered. In the simulation results, the metrics of the Throughput are related to the average value per node. Finally, the average delay shows the average one-way latency observed between transmitting and receiving a packet. It is important to mention that for each configuration, reported measurements are the mean of at least 20 runs with different random seeds.

The simulation work is divided in two parts. The first one compares between the performance of DSR and DSR enhanced with the SAFE mechanism. However, the second part discusses the improvement achieved by the enhanced DSR when it uses Passive Distributed Indexing (PDI) II.

In the rest of this paper, metrics related to the original DSR implementation are represented by *DSR*, which indicates that the DSR protocol performs without the SAFE mechanism. When the latter is activated, the measurements are identified as *Enhanced DSR*.

### A. Evaluation of the SAFE Performance

Figure 2 shows the delivery ratio for the above simulation scenario. As the pause time increases, the percentage of data packet is constantly improved. For example, for pause times above 100 seconds, the achieved delivery ratio for the Enhanced DSR is around 10% better than the normal DSR. When considering highly dynamic scenarios ( $PT \leq 100$ ), the Enhanced DSR performance is similar to the DSR's one.

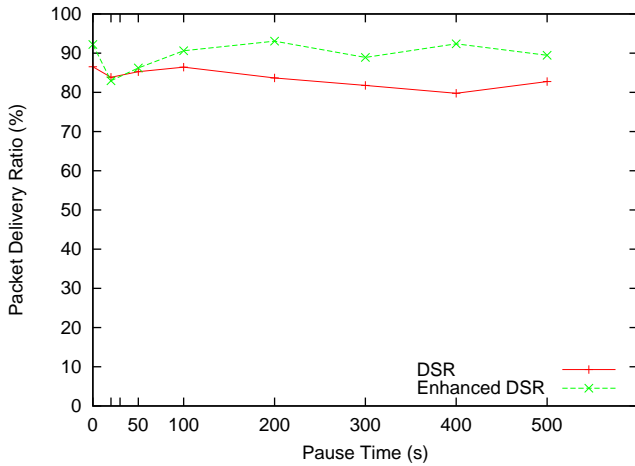


Fig. 2. Packet Delivery Ratio Comparison

The efficiency of the SAFE mechanism with respect to the throughput is similar to the one obtained for the delivery fraction depicted in figure 3. For medium and low mobility scenarios (medium:  $100s < PT \leq 300s$ , low:  $PT > 300s$ ), the throughput average is close to 235 byte/s for Enhanced DSR and 215 byte/s for DSR without enhancement. Thus, the SAFE mechanism allowed the detection of the bad behavior of the malicious node and the setup of a new path to avoid the malicious node. This means that a less percentage of data packets were dropped by the malicious nodes when the SAFE mechanism has been used. This situation is depicted in figure 2 which also shows that the delivery ratio has increased.

Based on the monitoring of the routing overhead depicted in figure 4, SAFE duplicates in almost all the cases the routing overhead in the network. This is quite clear because

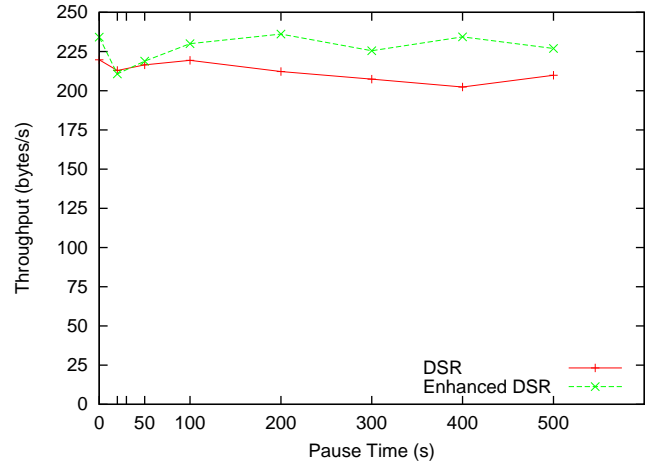


Fig. 3. Throughput

SAFE uses some new packet types. However, the SAFE mechanism also allows the detection, more frequently, of malicious node and the improvement of other parameters such as the delivery ratio, throughput and delay. In the sequel, we will show how the extra routing overhead generated by the SAFE mechanism is reduced without any degradation of the improvements achieved by this mechanism on the other metrics side.

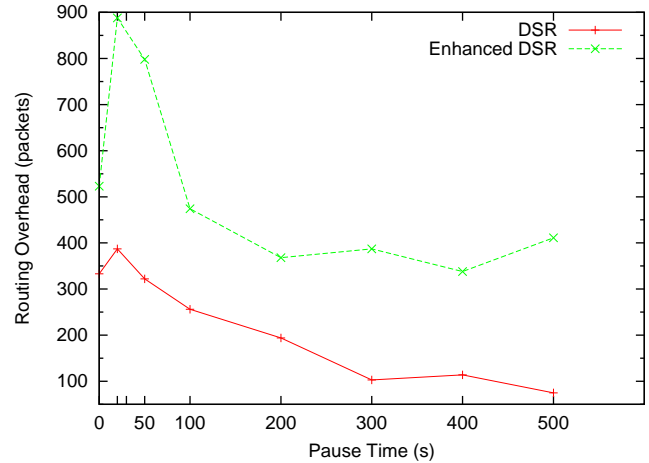


Fig. 4. Overhead

Figure 5 depicts the total number of packets dropped by the malicious node during the simulation. For the Enhanced DSR, the number of packets dropped has considerably decreased for the majority of the pause times (except for  $PT = 20s$  and  $PT = 50s$ ). This behavior is related to the ability of the SAFE mechanism to detect the malicious nodes and look at for an alternative path. In fact, establishing a new path that avoids the malicious node requires some modifications regarding the underlying protocol(DSR). This part was not totally implemented yet because we wished to

investigate the impact of mobility on the paths establishment after the detection of a misbehaviour. So far, the SAFE mechanism allows a node that detects a misbehaviour to inform the source node, which will simply launch a new route discovery. This technique is similar to the one used by DSR to setup a new path when a link failure is detected. As the nodes move, re-establishing a new path may come out either with a new path excluding the malicious node or with the previous path again. The simulation results we have got regarding this issue, are quite interesting and are discussed below.

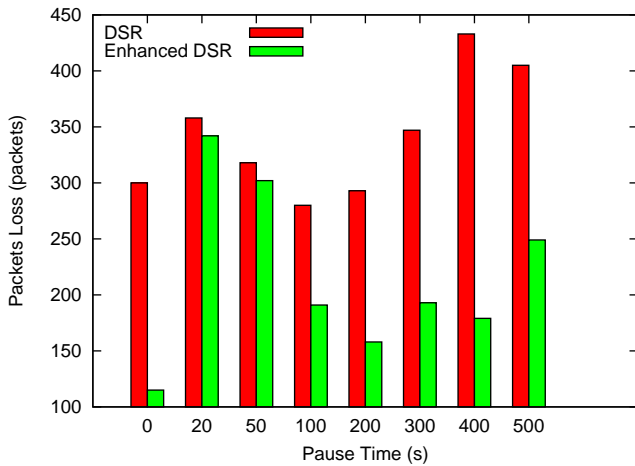


Fig. 5. Data Packets dropped by the Malicious node

The number of times for which the malicious node is detected is depicted in figure 6. Note that for a full motion of the network ( $PT = 0$ ), the detection of malicious nodes is more difficult due to the highly dynamic changes of the network. This simply means that the malicious node moves very fast and does not have enough time to drop packets. However, for high mobility configurations ( $PT = 50s$  and  $PT = 100s$ ), we obtain a “medium” level of detection which might be due to the fact that when the malicious node is detected and when a new path is established, the malicious node is again part of the new path. The low delivery packet ratios (See fig. 2) are the result of the number of link failures that are due to the huge changes occurred to the topology of the network. In the case of low mobility scenarios (except for  $PT = 500s$ ), the achieved detections are within the range (10,18). As for the static configurations, the malicious node was identified, however, it was during the whole simulation time part of the path between the source and the destination as the network topology did not change. To be more concrete, for  $PT = 500s$ , the nodes remain stationary during the simulation time and the malicious node was detected 34 times.

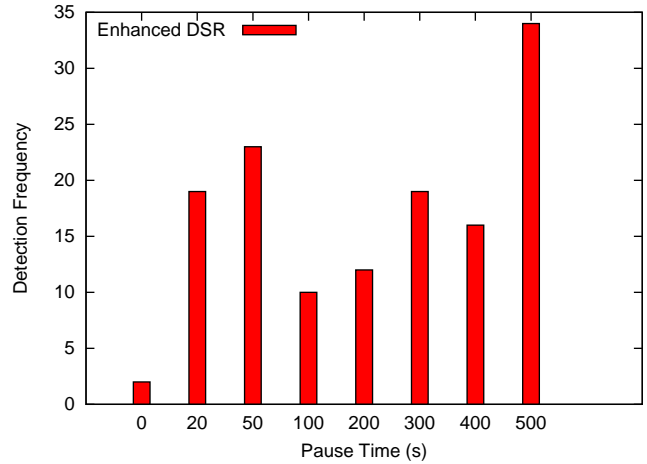


Fig. 6. Detection Frequency of the Malicious Node

### B. Performance of SAFE when using Passive Distributed Indexing (PDI)

In this part, the reputation repositories maintained by the SAFE Agents are enhanced through an epidemic indexing procedure described in section II. First, figure 7 shows that the packet delivery ratio has been improved when using the mentioned procedure. The enhanced DSR, in this case, outperforms the delivery ratio with more than 10% for all the pause times. In comparison with the previous metrics, for instance figure 2, the percentage of the packets correctly received by the destination has significantly increased for the high mobility scenarios ( $PT \leq 100s$ ).

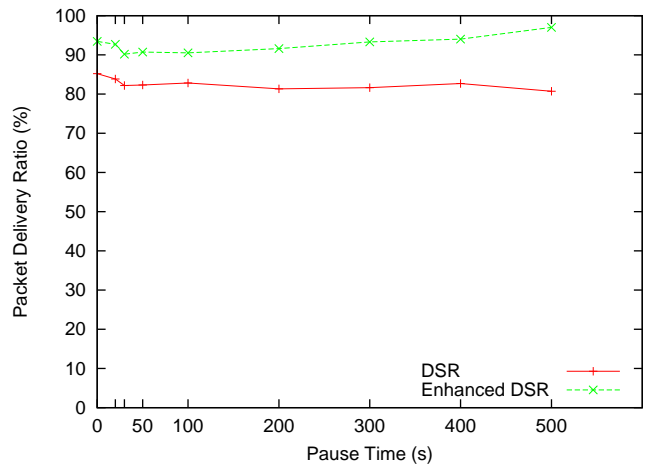


Fig. 7. Packet Delivery Ratio Comparison

The throughput is depicted in figure 8 that shows that some improvement was also achieved for high and low mobility scenarios. Due to the low delay generated by SAFE when using epidemic algorithms (See figure 9) and the high delivery ratio (See figure 7), the throughput of the Enhanced DSR oscillates within the “optimal” range (227, 249 bytes/s).

As for the DSR without enhancement, the throughput has decreased to a value below  $215\text{bytes/s}$  because there is no way to avoid malicious packet dropping within the normal DSR.

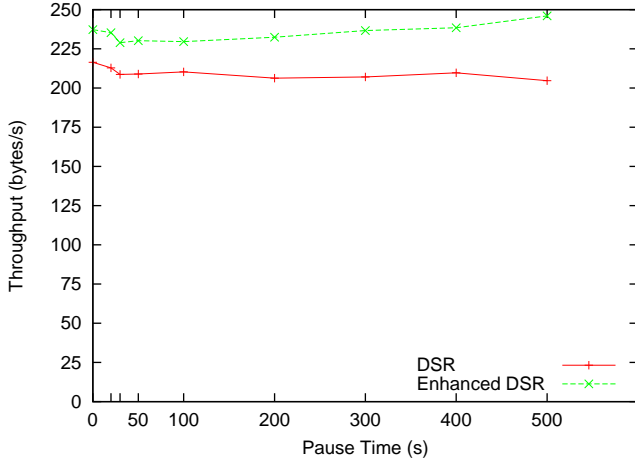


Fig. 8. Throughput

The obtained average delay seems to be similar to the one generated by the normal DSR as it is shown in figure 9. This means that the SAFE mechanism does not generate any extra delay for the end-to-end latency.

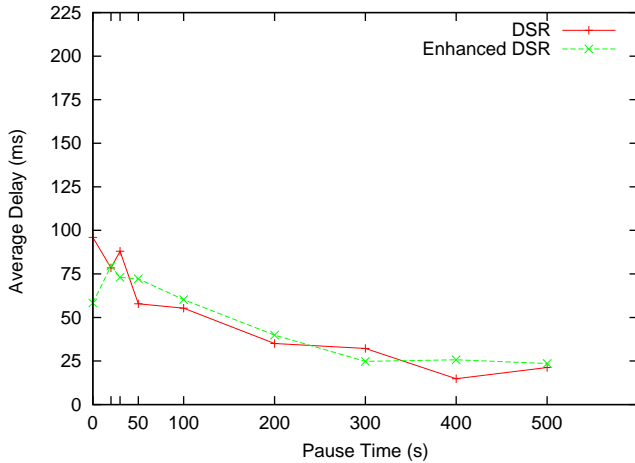


Fig. 9. End-to-End Average Delay

Similar to Passive Distributed Indexing (PDI) [6], SAFE provides effective means (See fig. 10) through local monitoring for coping with stale index entries due to weak connectivity, node failure, and modified data. This local monitoring has potentially less overhead than the previous version as shown in figure 4. We can see that the overhead of SAFE without any PDI enhancement remains constant between 400ms and 500ms for the pause times  $PT \geq 100s$  as depicted in figure 4. However, with this enhancement (See

figure 10), the reputation repositories are optimized. As a consequence, for the same pause times, an improvement is observed between 650 ms and 200 ms for 100s and 500s, respectively. It is important to mention that the routing overhead was also improved for high mobility scenarios. For instance, using the pause times 20s and 50s, the previous overhead is around 900 and 800 packets respectively as indicated in figure 4. However, when SAFE uses epidemic algorithms, the overhead values are 700 and 650 packets respectively as shown in figure 10.

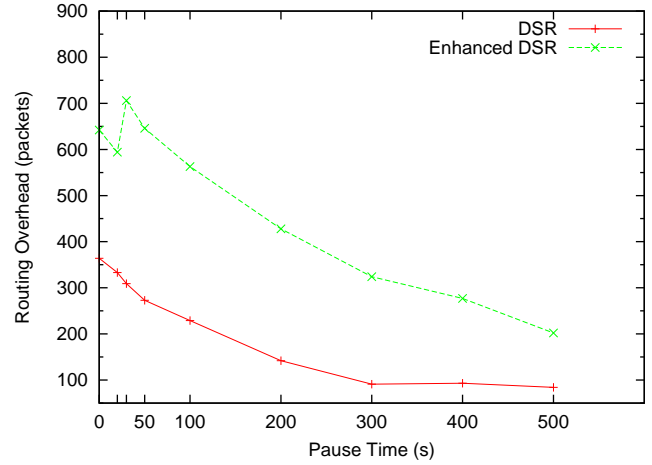


Fig. 10. Overhead

The simulation results also show (figure 11) the efficiency of our approach in reducing the number of packets dropped by the malicious node. We observe, in general, that the number of the dropped packets has decreased more than half for all the analyzed pause times. Again, we obtain a considerable improvement for high mobility configurations when comparing SAFE with the PDI enhancement and without this enhancement (figure 5). For instance, for the pause times 20 and 50 seconds, the malicious node has only dropped 150 and 200 packets respectively (See figure 11) in comparison with the previous performance results of SAFE depicted in figure 5 where the malicious node has dropped 345 and 300 packets respectively.

According to figure 12, when SAFE is used with the PDI enhancement, the malicious node is detected (in the case of low mobility configurations) less times than when SAFE is used without the PDI enhancement (figure 6). For instance, for the pause time 500 seconds, the malicious node was identified 5 times as shown in figure 12. Without PDI enhancement, the malicious node is detected 34 times (See figure 6).

Using the Passive Distributed Indexing (PDI) [6], SAFE analyses most queries locally without sending messages outside the radio ranges of the inquiring node. The simulation results show an improvement in the percentage of data packet delivery, high throughput, and low end-to-end delay. The main

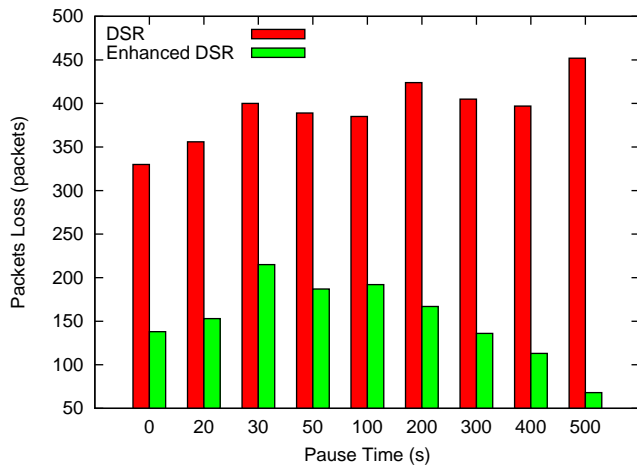


Fig. 11. Data Packets dropped by the Malicious node when PDI is used

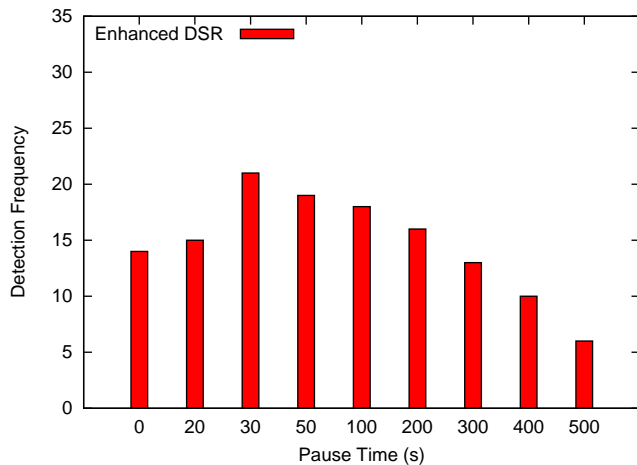


Fig. 12. Detection Frequency for SAFE

characteristic of SAFE is the explicit control over a local perspective of the information delivery and the minimisation of the routing overhead.

## V. FURTHER WORK

In the future, the following issues will be addressed,

- We mentioned earlier that establishing a new path that avoids a malicious node when this one is detected, is partly implemented. Due to the mobility and the initial topology of the network, the same path might be setup again when trying to avoid the malicious node if the DSR protocol is not modified. For this reason, we will modify the DSR protocol in order to enable the route discovery re-establishment to avoid the malicious node when setting up a new path
- We will evaluate the extensibility of the SAFE mechanism when the network is larger (i.e 50, 100 nodes) and the number of malicious nodes is bigger (10,20,...).

- we will also compare the performance of the SAFE mechanism with the other reputation systems such as CORE and CONFIDANT

## VI. CONCLUSION

So far, ad hoc networks are secured using authentication and encryption. These mechanisms seem to be inefficient against some other kind of attacks such as malicious packet dropping and denial of service. The mechanism we suggested enabled routing protocols to detect packet dropping frauds. In fact, the nodes in the network monitor independently the behavior of each other, however, they need to collaborate in order to identify the intruders. As this mechanism is based on the reputation concept, functionalities such as reputation information gathering and reputation computing are also discussed. Our proposal was also validated with some simulation work showing both its feasibility and performance.

## REFERENCES

- [1] P. Dewan et al, "Trusting Routers and Relays in Ad hoc Networks", In the International Conference in Parallel Processing Workshops, Kaohsiung, Taiwan, October 06-09, 2003
- [2] P. Feldman, "A Practical Scheme for Non-Interactive Verifiable Secret Sharing", In Proceedings of 28th IEEE Symposium on Foundations of Computer Science (Focs 87), IEEE Computer Science, 1987, pp. 427-437
- [3] P. Michiardi et al, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks". In Proceedings of the IFIP Conference, B. Jerman-Blazic and T. Klobucar editors, Kluwer academic, vol 228, Portoroz, Slovenia, September 26-27, 2002, pp. 107-121
- [4] S. Buchegger, J. Y. Le Boudec, "Performance Analysis of the COFIDANT Protocol: Cooperation Of Nodes- Fairness In Dynamic Ad hoc NeTworks", In IC/2002/01
- [5] P. Papadimitratos et al, "Secure Routing for Mobile Ad Hoc networks", In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002
- [6] C. Lindemann et al, "Exploiting epidemic data dissemination for consistent lookup operations in mobile applications". SIGMOBILE Mob. Comput. Commun. Rev., V8 N3. ACM Press, pp. 44-56 2004
- [7] Y. Zhang, B. Singh, "A Multi-Layer IPsec Protocol", In Proceedings of the 9th USENIX Security Symposium, Denver, Colorado, August 14-17, 2000
- [8] D. Senn, "Reputation and Trust Management in Ad Hoc Networks with Misbehaving Nodes", Diploma Thesis DA-2003.27, July 2003
- [9] UC Berkeley and USC ISI, "The network simulator ns-2", Part of the VIN T project, Available from <http://www.isi.edu/nsnam/ns>, 1998
- [10] C. E. Perkins et al, "Performance comparison of two on-demand routing protocols for ad hoc networks", In IEEE Personal Communications, V8 N1, February 2001, pp. 16-28
- [11] J. Broch et al, "A Performance Comparison of Multi-hop Wireless Ad Hoc Networks Routing Protocol", In the 4th annual ACM/IEEE international conference on Mobile computing and networking, ACM Press, 1998, pp. 85-97
- [12] D. Johnson et al, "The Dynamic Source Routing protocol for mobile ad hoc network", draft-ietf-manet-dsr-10.text, July 2004
- [13] A. Alexa, "Reputation Management in P2P Networks: The EigenTrust Algorithm", Available at [http://www.mpi-sb.mpg.de/units/ag5/teaching/ws03\\_04/p2p-data/01-20-paper2.pdf](http://www.mpi-sb.mpg.de/units/ag5/teaching/ws03_04/p2p-data/01-20-paper2.pdf)
- [14] L. Xiong, L. Liu, "A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities", In Proceedings of the IEEE International Conference on E-commerce (CEC'03), 2003
- [15] Y. Huang, W. Lee, "A cooperative Intrusion Detection System for Ad Hoc Networks", In Proceedings of the 1st ACM Workshop Security of Ad hoc and Sensor Networks, Virginia 2003
- [16] A. Burg, "Ad Hoc Network Specific Attacks", Seminar Ad hoc networking, Technische Universitaet Muenchen, 2003