Guaya-Delgado et al. EURASIP Journal on Wireless Communications and Networking (2019) 2019:77
https://doi.org/10.1186/s13638-019-1375-7

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A novel dynamic reputation-based source routing protocol for mobile ad hoc networks

Lenin Guaya-Delgado[1], Esteve Pallarès-Segarra[1*], Ahmad Mohamad Mezher[2] and Jordi Forné[1]

Abstract

Routing in mobile ad hoc networks is based on the cooperation of the network's nodes. The presence of selfish nodes that do not cooperate in this task drastically reduces the number of delivered packets. In order to find the better paths that include nodes willing to cooperate, we propose a new routing algorithm based on the reputation of the nodes. In our proposal, each node locally assigns a reputation value to the rest of the nodes in the network and next it uses the assigned reputation values to find out the better routing paths, in order to minimize the overall packet loss ratio. We assume that nodes have a stationary routing behavior, but we also include a mechanism to detect changes in their behavior. Our approach has been evaluated in the presence of selfish nodes, in order to compare it with the dynamic source routing algorithm, obtaining a reduction in the packet loss ratio at the expenses of a small increase in the number of hops taken by the packets to reach their destinations.

Keywords: Reputation, Routing protocols, Selfish nodes, MANETs

1 Introduction

A Mobile Ad hoc NETwork (MANET) is a group of self-organized wireless mobile nodes (MNs) able to communicate with each other without the need of any fixed network infrastructure nor centralized administrative support. Furthermore, the transmission range in such mobile devices is limited; thus, a packet is forwarded in a multihop path relying on the nodes in the routing path. Due to that, MANETs need the cooperation of every node in the path to achieve a successful packet delivery. However, depending on the MANET application, nodes are willing to cooperate with each other (e.g., rescuing services) since they are controlled by an authority, or might be reluctant to cooperate (e.g., data sharing [1], traffic monitoring [2], emergency assistance services [3, 4], and multimedia data transmission [5]) trying to save their own resources. Since MANET nodes usually have limited power (i.e., battery) and scarce computational resources (CPUs), nodes might refuse to cooperate in order to save their limited resources. This misbehavior of nodes would drastically affect the routing protocol operation.

It is worth mentioning that the presence of just a few number of selfish nodes could severely degrade the MANET performance [6]. Therefore, detecting selfish nodes is crucial to ensure an effective and efficient MANET routing operation.

1.1 Contribution and organization

In this paper, we aim to enhance the overall performance of MANETs in the presence of selfish nodes by designing an efficient and dynamic reputation-based algorithm to be used in any ad hoc routing protocol. We would like to highlight the fact that our reputation-based algorithm can be applied over any routing protocol for ad hoc networks that stablishes end-to-end forwarding paths. Using our approach, the routing protocol will establish the forwarding path with the highest nodes' reputation. We introduce a simple metric to estimate the nodes' reputation according to their packet forwarding behavior.

It is worth noting that in this work, we refer to the node's reputation as its availability to forward packets. Due to the nature of their operation without infrastructure,

* Correspondence: esteve@entel.upc.edu
[1]Department of Network Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
Full list of author information is available at the end of the article

the functionality of ad hoc networks heavily depends on the collaboration among nodes. However, selfish nodes aim to save battery by not forwarding others' packets. Thus, avoiding selfish nodes which do not aim to collaborate lead to ensure a proper network performance. Recall that a collaborative node that temporarily causes packet losses due to, e.g., network congestion or medium collision will soon recover its reputation when it moves to another better area. Thus, our goal is to detect nodes with a persistent selfish behavior. It is out of the scope of the proposal to define any mechanism to reward or penalize the nodes of the network in order to force their cooperation. Actually, our approach consists in a reputation-based forwarding strategy for ad hoc routing protocols that avoid using forwarding paths that include selfish nodes that would not forward packets properly. Thus, the quality of service offered to the user improves.

In the following, we highlight the novel contributions of our paper:

- We propose a new method to score the forwarding behavior of MANET nodes based only on first-hand information of each source node. This feature is congruent with the fact that in networks without infrastructure such as MANETs, nodes gather only local information.
- We propose a novel method based on the well-known Exponentially Weighted Moving Average (EWMA) [7] to detect any change in the nodes' forwarding behavior. The peak signal-to-noise ratio of the second difference of the EWMA is calculated to set the parameters of the EWMA that optimize that peak signal-to-noise ratio. This allows us to detect the limits of the stationary time windows in which the reputation of a node remains static. This way, we can properly update the measure of the forwarding behavior of a node by averaging the reputation samples in those time windows.
- Additionally, we propose a reputation strategy based on estimating the paths' reputations with the objective of finding the best forwarding path to be used by the sender node. Our novel reputation-based forwarding strategy for ad hoc routing protocols avoids using forwarding paths that include selfish nodes that would not forward packets properly; thus, the percentage of packet losses decreases. Most of the routing protocols already proposed in the literature use a metric based on the minimum distance. Alternatively, our proposal uses a metric based on reputation. This reputation is estimated using only local information, thus avoiding having to rely on the opinions of third parties. Furthermore:
- We analyze different moving average techniques in order to reduce the noise of the reputation sampled scores and to be able to assign a true reputation forwarding value to the network nodes.
- We carry out several simulations to check the accuracy of the theoretical results.
- Finally, we test the proposed forwarding strategy operating over the well-known Dynamic Source Routing (DSR) protocol [8]. It is important to notice that the proposed strategy can be easily applied to any end-to-end routing protocol. Results show a significant reduction of packet losses in the network at the expenses of a small increase in the number of hops taken by the packets to reach their destinations.

The rest of the paper is structured as follows: Firstly, Section 2 includes some relevant related work concerning reputation systems in MANETs. Then, Section 3 gives a brief explanation of the proposed routing protocol based on reputation. Next, Section 4 describes the mathematical analysis of the different proposals to assign a reputation value to the MANET nodes. After that, experimental results and discussion are shown in Section 5. Finally, conclusions and future work are given in Section 6.

1.2 Methods and experimental

In Sections 3 and 4, we present our proposal, whereas in Section 5, we validate the model and show simulation results. In Section 3, we introduce our approach of a reputation-based routing protocol for MANETs, called *dynamic reputation-based source routing* (DrepSR). DrepSR includes algorithms to compute the reputation of nodes and paths. Section 4 includes our methodology to average the reputation forwarding values of the nodes, using a cumulative moving average (CMA) to average the reputation samples, and an exponentially weighted moving average (EWMA) to detect behavioral changes in the nodes. The methodologies used to validate our proposal are as follows:

- We have developed a mathematical calculation to attain the peak signal-to-noise ratio of the second difference of an EWMA. We use this calculation to define the parameter of the EWMA that maximizes the peak signal-to-noise ratio and optimizes behavioral changes in the nodes of a MANET (see Section 4.3).
- We perform simulations on a static topology in order to validate the mathematical model developed (see Section 5.1).

Once our theoretical model is properly validated in Section 5.1, showing a good accuracy level, we carry out

a performance evaluation in Section 5.2. To do so, we present:

- A representative set of simulations of a MANET for different scenarios with mobile nodes, in which our proposal DrepSR is compared with the DSR protocol. Complete details of this performance evaluation are described in Section 5.2, including simulation settings and simulation results.

2 Related work

The cooperation issue of nodes in MANETs has received much attention from the research community over the last years. In general, reputation systems can be classified into two categories: *price-based schemes* and *reputation-based schemes*. In the following, we present some representative works related to each category.

Price-based schemes treat packet forwarding services as transactions that can be paid for and introduce virtual benefits as a form of rewards to those nodes that have been participating in packet forwarding activities. This means that nodes obtain virtual credits in exchange of offering packet forwarding. The paper [9] proposes a stimulation mechanism based on a counter in each MANET node. The counter increases when the node forwards packets for others and decreases, following a function that depends on the estimated hop count to destination, when the node sends its own packets. A micro-payment scheme for multi-hop cellular networks is presented in [10] to encourage collaboration in packet forwarding by letting users benefit from relaying others' packets. Also, they propose mechanisms for detecting and rewarding collaboration, while detecting and punishing cheating. The work [11] presents a fair and efficient incentive mechanism to stimulate the node cooperation in MANETs by charging source and destination nodes when both of them benefit from the communication. To implement this charging policy efficiently, hashing operations are used in the ACK packets to reduce the number of operations. It has also pointed out that MANET nodes should cooperate and forward packets for others; otherwise, it would be impossible to achieve a pure ad hoc network [12].

Reputation-based schemes basically focus on evaluating each node's behavior and on detecting misbehaving nodes according to their reputation values. In [13], authors propose a watchdog method to identify misbehaving nodes and a path rater technique that helps routing protocols to avoid choosing those nodes to forward any packet. Simulation results prove the benefits of using those two techniques by increasing the throughput to at least 17%. Authors in [14] propose a protocol called CONFIDANT (cooperation of nodes: fairness in dynamic ad hoc networks) able to detect and isolate misbehaving nodes. They show that using CONFIDANT, a network can perform well even with a high percentage of malicious nodes. In [15], the authors present a generic mechanism based on reputation to impose cooperation among MANET nodes to avoid selfish behavior. Authors in [16] propose the observation-based cooperation enforcement in ad hoc networks (OCEAN) to detect and mitigate misleading routing behavior in ad hoc networks. Simulation results show that OCEAN works very well in terms of throughput. However, OCEAN is not fully able to penalize misbehaving nodes severely even though it performs at least equally well in comparison to another more complex schemes. The proposal [17] employs a trust-based system to counteract malicious node's behaviors that consists of a reputation system and a watchdog technique. Their watchdog technique uses a positive feedback message (PFM) as an evidence of the forwarding node's behavior. Regarding [18], the authors proposed a reputation-based trust management for detecting and preventing MANET vulnerabilities. Results of their performance evaluation show scalability and robustness using the proposed scheme.

In [19], authors propose a novel routing strategy that works with nodes that were considered trustworthy, but due to loss of power, they are reconsidered as selfish ones. Their strategy considers that selfish nodes are not malicious by default, and they are forced to provide their current energy level. This helps the routing strategy to avoid nodes with very low power based on the assumption that a node with low energy could drop packets to save its energy consumption. The proposal [20] proposes a trust mechanism based on a trust vector model (TV). The main goal of this mechanism is to detect malicious nodes to later take actions on them. It is implemented in two well-known routing protocols: dynamic source routing (DSR) [8] and ad hoc on-demand distance vector (AODV) [21], producing two proposals called TV-DSR and TV-AODV, respectively. Simulation results on both modified routing protocols show that they can effectively detect malicious nodes and mitigate their attacks. Authors in [22] propose a new mechanism to detect selfish nodes. They consider as a good strategy that selfish nodes drop control packets to avoid themselves being asked to forward data and in this way they save resources for their own use. Nodes are expected to contribute to the network within a time frame. Those nodes who fail to will undergo a test for their suspicious behavior. Simulation results support their scheme by obtaining good results.

Game theory and mechanism design have been widely applied for selfish routing. Authors in [23] propose an indirect reciprocity framework based on a game-theoretic solution to enforce cooperation among nodes.

Based on the proposed model, they obtain the threshold of benefit-to-cost ratio to ensure the convergence of cooperation. Simulation results demonstrate that their game-theoretical solution enforces cooperation among nodes when the benefit-to-cost ratio of the unselfish exceeds the critical condition. The proposal [24] addresses reducing energy consumption of energy-constrained wireless nodes through a game-theoretical energy-aware cooperative relaying scheme with fair distribution of the payoff among players, to keep them satisfied and discourage them from quitting the coalition. This solution also proposes a credit based system to reward cooperative players, hence excluding selfish users from cooperative coalitions.

The approach presented in [25] consists of a dynamic reputation management system to detect and isolate misbehaving nodes in MANETs. They introduce a novel direct monitoring technique that evaluates the nodes' reputation in the network. This technique ensures that misbehaving nodes will be detected and isolated from the network, whereas the rest of the nodes that spend their energy in forwarding data will be allowed to accomplish their network activities. Simulation results show the effectiveness of their model in restraining and mitigating the effects of misbehaving nodes in MANETs. Authors in [26] show how the use of their proposal called neighborhood compressive sensing (NCS) helps in the reduction of resource consumption and at the same time protects the network from attacks and misbehavior. The NCS model compresses sparse data such as routing tables updates. In addition, as individual nodes only accept routing tables updates if it is coming from the leader node's processed information, NCS also prevents the network from attacks and misbehavior. Simulation results show how the NCS model outperforms DSR in terms of energy consumption, lifetime, and packet dropping ratio.

To the best our knowledge, the approach of considering the reputation of candidate MANET nodes to conform the forwarding scheme, using only first-hand information of the nodes to score their reputation and using an exponential weighted moving average model to detect any changes in nodes' reputation, is novel. In the next section, we present our proposal of a reputation-based routing protocol for MANETs.

3 Proposed routing protocol based on reputation

We would like to highlight the fact that our reputation-based forwarding scheme could be implemented over any ad hoc routing protocol that performs an end-to-end forwarding scheme. For the sake of simplicity, in this work, we have selected the well-known DSR protocol, although similar benefits could be obtained over other ad hoc routing protocols. Briefly, DSR [8] looks for routes formed by intermediate nodes when a source needs to send a packet to a destination. To do so, DSR periodically sends route discovery packets to search paths to that destination. Route reply packets contain the whole route for that packet. To fulfill source routing, the routed packets contain the address of each node the packet will traverse. In the event of link breakages, route error packets are generated and alternative routes will be found looking for stored paths in the route cache or looking for new routes.

Our proposal, called dynamic reputation-based source routing (DrepSR), uses DSR as the routing protocol engine to discover the available routes to destination. In addition, DrepSR estimates the reputation of those discovered paths. Then, when a route stored in cache has to be chosen, DrepSR selects the most reputable one instead of the shortest one. The goal is to choose, among the set of available paths, those paths that produce the fewest packet losses. We assume that the main cause of packet loss is the non-forwarding behavior of selfish nodes present in the network. As mobile services increase (e.g., video-on-demand, gaming, social networks), users are worried with the battery lifetime of their smartphones and try to save battery. Relaying packets for others consumes battery; thus, users might feel reluctant to cooperate in the forwarding tasks because of worrying about offering their limited-energy battery. Anyway, although there were other factors for packet losses (e.g., congestion, collisions), it would also be beneficial to use our proposed algorithm. In DrepSR, each node locally assigns a reputation value to the other nodes with which it interacts, using only first-hand information based on its own experience. After that, the reputations of the available paths in cache are calculated using the estimated reputation of the individual nodes that form those paths. The reputation of a path is easily calculated as the product of reputations of those intermediate nodes that form that path. In this way, the algorithm assigns a reputation score to the paths discovered by the DSR mechanism. Finally, each time a node requires to send information to a specific destination and finds out more than one available path, the node will choose the path with the highest reputation.

To estimate the reputation of a node, it is necessary to have feedback of the percentage of packet losses in the path to which the node belongs. In fact, while a source node has no previous experience in the network, it cannot estimate the reputation of the other nodes. Initially, the source node estimates the fraction of packet losses (L_p) for each path p through which it transmits. To get the end-to-end percentage of packet losses, the feedback information of the transport layer is used. In the case of non-connection-oriented communications, the use of RTP/RTCP (Real Time Protocol/Real Time Control Protocol) over UDP (User Datagram Protocol) is

assumed. RTCP provides the percentage of packet losses through each available path. For the case of connection-oriented communications, the most widely used protocol is TCP (Transmission Control Protocol). In this case, losses are estimated from the number of retransmissions done by the source node regarding a same message. The main reason for making this choice is its simplicity, and although it may seem a coarse estimation, it is accurate enough to select the best path in the network. With this feedback information about the packet losses, the node calculates L_p as the ratio between the amount of lost packets and the amount of sent packets through path p. This way, our cross-layer proposal uses transport information to take forwarding decisions at the network layer. The L_p is calculated periodically, updating then the reputation of the nodes that belong to that path.

3.1 Algorithm to compute the reputation of nodes and paths

Once the estimation of the fraction of packet losses for a path has been done, the reputation of the intermediate nodes is estimated by sharing out the losses equally among nodes. That is, the algorithm considers that all the nodes of a path, loss the same amount of packets, which is not ever true. Therefore, if a well-behaving node belongs to a path in which there are several selfish nodes, its reputation will be penalized. Although it may seem unfair from the point of view of the nodes, it is not, since the goal of the algorithm is not to estimate an accurate reputation value, but to choose the path with fewer losses. Consequently, the design of schemes to reward or penalize nodes so that we can force their cooperation in the forwarding tasks is not our goal in this present paper. Anyway, it is important to notice that due to the inherent mobility of MANET nodes, a cooperative node wrongly penalized for having been in the same forwarding path than a selfish node will soon recover its reputation when it moves to another area.

The forwarding reputation value (r_n) assigned to a node n represents the forwarding probability for node n, that is, the ratio between the number of packets forwarded by the node and the total number of packets it has received. In the same way, we define the reputation value (R_p) assigned to path p as the ratio between the packets received by the destination node and the packets sent by the source node. Then, the reputation of a path is obtained from its fraction of packets loss given by Eq. (1)

$$R_p = 1 - L_p \tag{1}$$

A packet reaches its destination successfully if all the nodes of the path successfully forwarded it. Therefore, assuming that the forwarding behavior of each node is independent from the others, the reputation (R_p) of path p can be computed as the product of the reputations (r_n) of the nodes n that form that path p, see Eq. (2).

$$R_p = \prod_{n \in path\ p} (r_n) \tag{2}$$

As said before, the proposed algorithm estimates the reputation of the nodes ($\widehat{r_n}$) sharing out the losses equally between the nodes that form the path, except for the origin node that is not supposed to generate losses. In this case, Eq. (2) can be rewritten as follows, being $\widehat{r_n}$ the estimated reputation of node n and N_p the number of intermediate nodes in the path p.

$$R_p = (\widehat{r_n})^{N_p} \tag{3}$$

Each time the source node updates the fraction of packet losses for path p, the forwarding reputation value of each node belonging to that path is estimated. To do that, the reputation value of the path calculated in Eq. (1) is used in Eq. (3) to obtain Eq. (4).

$$\widehat{r_n} = \left(1 - L_p\right)^{\frac{1}{N_p}} \tag{4}$$

As mentioned before, the fact of dividing the losses equally between the nodes that compose the path could negatively affect the reputation of nodes that have a collaborative behavior. Furthermore, this also might hide a selfish behavior (i.e., the node obtains a higher reputation than it deserves). For this reason, the reputation of nodes must be averaged taking into account the estimations made in all the paths in which the node has previously participated. In the next section, we propose a methodology to average the reputation samples of the nodes.

The main goal is that each node has a table with the estimated reputations of the other nodes. This way, nodes can assess the reputation of the different discovered paths and choose the most reputable one. The reputation of the paths is computed by using the estimated reputations of the nodes in Eq. (2).

A drawback of the proposed algorithm is that nodes with a bad reputation can experience difficulties to update their reputation in case of changing their behavior and cooperate hereinafter. This happens because those nodes have not been selected as forwarding nodes for a while (since source nodes try to avoid them due to their selfish behavior), so they cannot provide new updated information of their current well behavior. To tackle this issue, we make nodes to periodically forget all they have learnt about the rest of the nodes, as a kind of reset in the framework.

In the following, we present the algorithms proposed to select the best forwarding path. Algorithm 3 is the main algorithm in our methodology. This

algorithm checks, among the set of active paths, if it has new information about the packet losses experienced in the path. If positive, it updates the reputation of the nodes that belong to that path using Algorithm 1 (*UPDATENODESREPUTATION*). In the case that it is necessary to find a route to a destination, a DSR route discovery is done to find out the available paths to that destination. Next, the reputation of each discovered path is calculated using Algorithm 2 (*GETPATHREPUTATION*), and the path with the highest reputation score is selected. Additionally, Algorithm 1 uses the *AVERAGE* function (described in Algorithm 4) to estimate the abovementioned nodes' reputations that will be explained in the next section.

4 Algorithm to average the reputation forwarding values

We assume that nodes have a stable forwarding behavior for an undetermined time. This entails that the average reputation forwarding value of the nodes is kept stationary during an arbitrary interval. Nevertheless, we also consider the possibility that the forwarding behavior of the nodes will change throughout time starting a new stationary interval. This may be due to the fact that the node either changes its behavior or its neighborhood, which affects its reputation.

As mentioned above, each time a node gets the reputation of a path, an estimation of the reputation of the nodes belonging to that path is updated. For convenience in the nomenclature, we name the set of k consecutive samples of the estimated behavior for a node n, as $r_1, r_2, \ldots, r_k$ instead of $\widehat{r_{n_1}}, \widehat{r_{n_2}}, \ldots, \widehat{r_{n_k}}$. For the sake of simplicity and clear explanations, we ignore the subscript n to refer to the node n under evaluation. To assign a reputation to that node, we use a moving average. The moving averages that we have considered are the cumulative moving average (CMA) and the exponentially weighted moving average (EWMA) [7].

4.1 Cumulative moving average

In the case of using a CMA [7], we compute the average using all the reputation history of a node:

$$c_k = \frac{1}{k} \cdot \sum_{i=1}^{k} r_i \tag{5}$$

In Eq. (5), c_k represents the CMA of node n after its k-th reputation sample has been estimated. Since past samples are weighted equally than the present ones, this average does not reflect the updated current behavior of the node in case it changed. If the behavior of the node were always the same, this would be a correct average to be implemented.

We assume that the node has a stationary behavior within a time window. Otherwise, it would not have sense to average the reputation samples. Under this assumption, we can recalculate the CMA each time a new stationary window starts. Now, the problem is to obtain the size of that window, that is, we need to detect when a change occurs in the behavior of the node, so that the average only includes samples that have the same stationary behavior. By means of a mechanism that allows us to detect those behavioral changes, we could define the window in which we would average the reputation samples. Each time a new window starts, we forget the old samples, that is, we reset the i counter in Eq. (5).

The set of reputation samples r_k of a node n that has a steady behavior can be expressed as a constant value $\bar{r}$ to which a noisy signal Δr_k is superimposed.

$$r_k = \bar{r} + \Delta r_k \tag{6}$$

The noisy signal is a zero mean stochastic process ($E[\Delta r_k] = 0$). Therefore, $\bar{r}$ is the expected value of the samples $E[r_k]$, which remains constant for all k samples because of the stationary behavior of the nodes. The second moment of Δr_k is the variance of the samples, which we assume to be constant for the temporary window of steady behavior:

$$\sigma_r^2 = E\left[(\Delta r_k)^2\right] \tag{7}$$

In the same way, we can express $c_k = \bar{c}_k + \Delta c_k$. Then, using Eq. (6) in Eq. (5), we obtain

$$\bar{c}_k = E[c_k] = \bar{r} \tag{8}$$

$$\Delta c_k = \frac{1}{k} \cdot \sum_{i=1}^{k} \Delta r_i \tag{9}$$

The expected value of the averaged samples using CMA does not depend on k and remains constant for the time window with steady behavior. Since Δr_k is a zero mean process, Δc_k is a zero mean process as well. In order to evaluate the deviation from the expected value, we calculate the variance of the CMA samples. We assume that forwarding is a memoryless process, that is, a node that forwards only the 50% of the packets throws a coin before each forwarding independently of the others. Under this assumption, we have that processes Δr_i and Δr_j are independent for $i \neq j$ and the variance is given by Eq. (10):

$$\sigma_c^2(k) = \frac{1}{k^2} \cdot \sum_{i=1}^{k} E\left[(\Delta r_i)^2\right] = \frac{\sigma_r^2}{k} \tag{10}$$

Concluding, for the CMA under the assumption of steady behavior in a well-defined temporary window, the

expected value of the reputation samples remains the same. Also, the variance of the reputation samples reduces inversely proportional to the number of samples.

4.2 Exponentially weighted moving average

Another widely used alternative to average samples is EWMA [7]. While averaging, more emphasis to recent samples is placed reducing exponentially the weight of past samples. We denote w_k as the value of the EWMA after processing the r_k sample:

$$w_k = (1-\beta)\cdot w_{k-1} + \beta \cdot r_k \tag{11}$$

In Eq. (11), β is a factor between 0 and 1 that weights the current sample r_k. The higher the β, the more emphasis is given to the present. In general, EWMA has less lag than CMA and is therefore more sensitive to changes.

To calculate the EWMA of a node, it is necessary to take an initial reputation value. A priori we consider that all nodes in the network are collaborative. Therefore, the initial reputation for all nodes must be one. Nevertheless, this proposal is defined in a generic way for any other initial value. Using Eq. (11) recursively, we obtain the moving average as a function of all the reputation samples exponentially weighted, where w_0 is the initial value of the moving average:

$$w_k = (1-\beta)^k \cdot w_0 + \beta \cdot \sum_{i=0}^{k-1} (1-\beta)^i \cdot r_{k-i} \tag{12}$$

In the same way that we have done with r_k in Eq. (6), we can express $w_k = \overline{w}_k + \Delta w_k$, being $\overline{w}_k$ the asymptotic behavior of the EWMA and Δw_k a stochastic process that superimposes to it. Using Eq. (6) in Eq. (12), we relate $\overline{w}_k$ with $\overline{r}$ and Δw_k with Δr_k.

$$\overline{w}_k = E[w_k] = (1-\beta)^k \cdot (w_0 - \overline{r}) + \overline{r} \tag{13}$$

$$\Delta w_k = \beta \cdot \sum_{i=0}^{k-1} (1-\beta)^i \cdot \Delta r_{k-i} \tag{14}$$

We can see that after k samples, the difference between the expected value of the sample and the asymptotic behavior of the moving average, that is $\overline{w}_k - \overline{r}$, has reduced by a factor $(1-\beta)^k$. Since $(1-\beta)^k < 1$, when the number of samples k grows, this difference tends to zero and $\overline{w}_k$ approaches asymptotically to $\overline{r}$. Therefore, the greater the β factor, the faster the asymptotic approach to the desired value $\overline{r}$.

Since Δr_k is a zero mean process, Δw_k is a zero mean process as well. Note that according to our definition of r_k (see Eq. (6)), we have that $\Delta r_k = 0, \forall\, k < 1$.

In order to evaluate the deviation of the EWMA of the reputation samples, from the asymptotic behavior, we calculate the variance of the EWMA of the reputation samples. Using the same assumptions as in CMA, we attain the variance of the EWMA as a function of the variance of the reputation samples (see [7] for further details):

$$\sigma_w^2(k) = \frac{\beta}{2-\beta} \cdot \left[1-(1-\beta)^{2k}\right] \cdot \sigma_r^2 \tag{15}$$

Notice that although we suppose that σ_r^2 is constant in the stationary window, σ_w^2 depends on k. Since always <1, we have that $\sigma_w^2(k) < \sigma_r^2$. This means that the EWMA reduces the deviation of the samples from their mean value. The smaller the β factor is, the smaller the variance $\sigma_w^2(k)$ is. If we compare the variances of EWMA and CMA, we can notice that for CMA, this variance tends to zero when the number of samples increases (Eq. (10)). In contrast, for EWMA, when the number of samples grows, the variance tends to a value that depends on β which is shown in Eq. (16):

$$\sigma_w^2 \approx \frac{\beta}{2-\beta} \cdot \sigma_r^2 \tag{16}$$

Additionally, for EWMA, there is a trade-off between the asymptotic approaching speed and the variance. High values for β entail a quick approach to the average stationary value, but with a low variance reduction. Conversely, low values for β reduce considerably the variance, although with a slow approach.

4.3 Stationary time windows detection

CMA is the best option for averaging the reputation samples, in case the behavior of the nodes changes steeply and we are able to detect the moments in which the stationary time windows start and end. To do so, we propose a method to detect changes in the nodes' behaviors and delimit these stationary time windows. Our proposal is based on the combined use of an EWMA to detect the change of the nodes' behavior and a CMA to average their reputation values.

Figure 1 shows the EWMA for the reputation of a node that changes, defining two stationary time windows. If the asymptotic behavior of the EWMA was a continuous function, we would have a discontinuity in the derivative each time the stationary behavior of the node changes and there is a Dirac delta function in the second derivative. Since we have a discrete time function, we use the second difference of the EWMA instead of the second derivative in order to detect the end of a stationary time window. We define the first difference of the EWMA as:

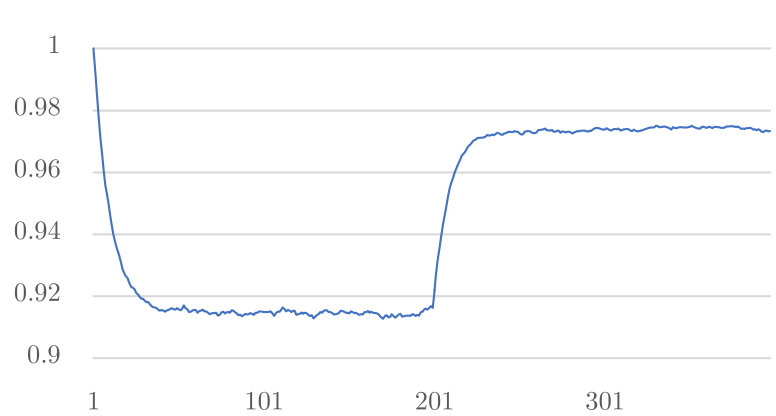

Fig. 1 Average of the reputation samples of a node if we use an EWMA when the node changes its behavior defining two stationary windows

$$f_k = w_k - w_{k-1} = \beta \cdot (r_k - w_{k-1}) \tag{17}$$

and the second difference

$$d_k = f_k - f_{k-1} = \beta \cdot (r_k - r_{k-1}) - \beta^2 \cdot (r_{k-1} - w_{k-2}) \tag{18}$$

With this definition, if a peak for d_k occurs at k_0, it means that the sample r_{k_0} belongs to a new stationary time window. The above expression has no sense for $k < 3$, since at least three samples are necessary to compute the second difference.

In order to find the values of β that optimize the peak detection, we relate the second difference d_k with the samples r_k. Using Eq. (12) on Eq. (18), we have:

$$d_k = \beta \cdot r_k - \beta \cdot (1+\beta) \cdot r_{k-1} + \left(\frac{\beta}{1-\beta}\right)^2 \cdot \left[\beta \cdot \sum_{i=2}^{k-1} (1-\beta)^i \cdot r_{k-i} + (1-\beta)^k \cdot w_0\right] \tag{19}$$

Under the assumption that in the stationary window $r_k = \overline{r} + \Delta r_k$, we can write $d_k = \overline{d_k} + \Delta d_k$, being

$$\overline{d_k} = \beta^2 \cdot (1-\beta)^{k-2} \cdot (w_0 - \overline{r}) \tag{20}$$

$$\Delta d_k = \beta \cdot \Delta r_k - \beta \cdot (1+\beta) \cdot \Delta r_{k-1} + \frac{\beta^3}{(1-\beta)^2} \cdot \left[\sum_{i=2}^{k-1} (1-\beta)^i \cdot \Delta r_{k-i}\right] \tag{21}$$

To relate the noise of the second difference of the EWMA with the noise of the samples, we calculate the standard deviation of d_k. As we have said above, we assume that $E(\Delta r_i \cdot \Delta r_j) = 0, \forall\, i \neq j$ and $E(\Delta r_i^2) = \sigma_r^2, \forall i > 0$ in the stationary window. In this case:

$$\sigma_d^2 = E(\Delta d_k^2) = \beta^2 \cdot E(\Delta r_k^2) + \beta^2 \cdot (1+\beta)^2 \cdot E(\Delta r_{k-1}^2) + \frac{\beta^6}{(1-\beta)^4} \cdot \left[\sum_{i=2}^{k-1} (1-\beta)^{2i} \cdot E(\Delta r_{k-i}^2)\right]$$

$$\sigma_d^2 = \beta^2 \cdot \left[1 + (1+\beta)^2 + \beta^3 \frac{1-(1-\beta)^{2(k-2)}}{2-\beta}\right] \cdot \sigma_r^2 \tag{22}$$

Plotting the factor that relates both variances as a function of β, using a different number of samples (see Fig. 2), we get practically the same graph being impossible to distinguish them, then we can approximate the above expression for the case $k \to \infty$(in this case we approximate $(1-\beta)^{2(k-2)} \approx 0$, since $\beta < 1$).

$$\sigma_d^2 \approx 2\beta^2 \cdot \frac{2+\beta}{2-\beta} \cdot \sigma_r^2 \tag{23}$$

For large values of β, the noise σ_d^2 increases, being greater than that of the samples σ_r^2. Therefore, we have to choose a small value for β if we want to reduce the noise and optimize the detection.

On the other hand, the peak of the second difference will be higher the value of β, being easier to detect. We must compare the value of the peak with the noise and find the value of β that optimizes the peak signal-to-noise ratio. To calculate the peak value, we assume that the reputation of the node changes from a value $\overline{r_1}$ to a value $\overline{r_2}$, and this change occurs between the sample $k_0 - 1$ and k_0 so that $r_{k_0-1} = \overline{r_1} + \Delta r_{k_0-1}$ and $r_{k_0} = \overline{r_2} + \Delta r_{k_0}$. Using Eq. (19) we get:

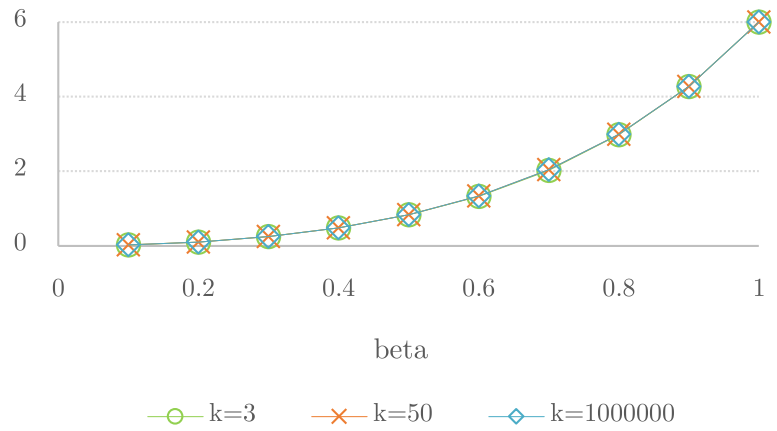

Fig. 2 Plot of the factor that relates the variance of the samples σ_r^2 and the variance of the second difference of the exponentially weighted moving average of the samples σ_d^2, as a function of β for different number of samples k (see Eq. (22))

$$\overline{d_{k_0}} = \beta \cdot (\overline{r_2} - \overline{r_1}) + \beta^2 \cdot (1-\beta)^{k_0-2} \cdot (w_0 - \overline{r_1}) \tag{24}$$

Comparing this result with Eq. (20), we observe that the peak that appears in the second difference caused by the change of behavior is:

$$P = \beta \cdot (\overline{r_2} - \overline{r_1}) \tag{25}$$

Using Eqs. (23) and (25), we can define the peak signal-to-noise ratio as a function of β:

$$\frac{P^2}{\sigma_d^2} = \frac{2-\beta}{2 \cdot (2+\beta)} \cdot \frac{(\overline{r_2} - \overline{r_1})^2}{\sigma_r^2} \tag{26}$$

Since $(2-\beta)/2(2+\beta)$ is a decreasing function of β, we can conclude that the lower the β value, the better the detection of a new stationary time window. Obviously, we cannot use $\beta = 0$ because in this case, we were not averaging samples.

4.4 Proposed averaging algorithm

Our proposal is to average the samples of the reputation of a node using a CMA in the stationary period of steady behavior for each node. We use this average as the reputation of the nodes in order to find the best forwarding routes in the MANET. To delimit this stationary period, in parallel, we calculate the EWMA of the samples using a low value for β, whose value is discussed in the simulation results section. Also, we compute the second difference of that EWMA average obtaining the values d_k. We suppose that the samples d_k represent an ergodic process and therefore its mean value $\overline{d_k}$ and its standard deviation σ_{d_k} are calculated as follows:

$$d_k = \frac{1}{k} \cdot \sum_{i=1}^{k} d_i \text{ and}$$
$$\sigma_{d_k} = \sqrt{\overline{d_k^2} - \overline{d_k}^2} \text{ being } \overline{d_k^2} = \frac{1}{k} \cdot \sum_{i=1}^{k} d_i^2 \tag{27}$$

In Eq. (27), k represents the number of samples that we have in the stationary time window. The more samples, the more accurate the result we achieve.

When the absolute value of a new sample d_{k+1} is higher than a certain number of standard deviations added to the mean value, a peak detection event occurs. In this case, we start a new CMA with the new samples that belong to a new stationary time window. The peak detection condition is shown in Eq. (28):

$$|d_{k+1}| > |\overline{d_k}| + F \cdot \sigma_{d_k} \tag{28}$$

Since we do not know if the reputation of the node will increase or decrease, we do not know if the peak of the second difference will be positive or negative. Therefore, we use the absolute value of d_{k+1} and $\overline{d_k}$. The factor F determines how much higher the peak should be with respect to the noise. A small F value may be the cause of false positives so that the noise in the samples may seem like a peak. On the other hand, a large F value may be the reason why small changes in the behavior of a node are not detected. As seen in Eq. (25), the amplitude of the peak depends on the difference of both reputations, the old and the new one.

To sum up, the proposed algorithm uses a CMA to average the reputation samples and the second difference of the EWMA with a small β to detect the limits of the stationary time windows.

5 Experimental results and discussion

In this section, we first validate our proposal in Section 5.1, and afterwards, we carry out a performance evaluation in Section 5.2.

5.1 Validation of the theoretical model

In order to validate the theoretical results in the detection of behavioral changes in the MANET nodes and to be able to define the temporal windows in which nodes have a stationary behavior, several Matlab R2017b [27] simulations have been performed with the topology shown in Fig. 3. It depicts a source node (S) that transmits to three different destinations (D_i, $i = 1, 2, 3$). Every 1000 packets transmitted through a path, the loss probability for that path is calculated and a reputation sample is obtained for each one of the intermediate nodes that form that path. In total, 100,000 packets have been transmitted to each destination. All nodes forward all packets they received except node 2, which does not forward a 30% of the packets during the first half of the simulation and a 10% during the second half. That is, its real reputation goes from 0.7 to 0.9. Node 4 only participates in paths S-D_2 and S-D_3 while nodes 1, 2, and 3 participate in all paths. Thus, node S should assign the same reputation to nodes 1, 2, and 3.

Initially, for different values of β, we check if the behavioral change of node 2 is clearly detected, and we also verify if node S correctly assigns their reputation value to nodes 1 and 4. The detection threshold has been set with $F = 2$, since this value showed a very good detection rate after conducting many simulations. This way, a detection takes place if $|d_{k+1}| > |\overline{d_k}| + 2 \cdot \sigma_{d_k}$. In Figs. 4, 5, 6, 7, 8, 9, and 10, the horizontal axis indicates the number of samples. Each time a peak of $|d_k|$ gets above the threshold, the average of the reputation samples is reset.

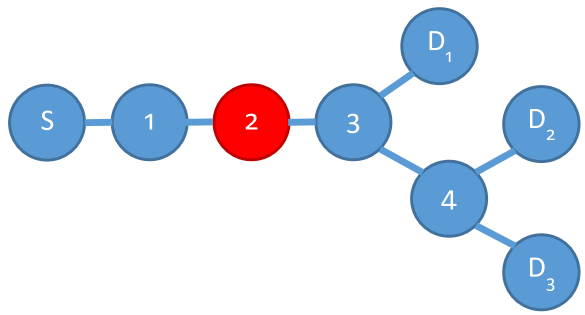

Fig. 3 Network topology used in the simulation with a source node (S) and three destinations (D_1, D_2, and D_3). In the simulation, all nodes forward 100% of the packets, except node 2 that forwards only a 70% of the packets during the first half of the simulation. Afterwards, the node changes its behavior and forwards a 90% of the packets during the last half of the simulation

Since node 1 participates in different paths, its samples are noisier than those of node 4. The noise makes it difficult to detect the change in the behavior of node 2 if the threshold of detection is too high. In Fig. 4, it is observed that for a value of $\beta = 0.01$, the detection can perfectly be done, although false positives are obtained. A false positive occurs when $|d_k|$ gets above the threshold but actually, there was not a change in the behavior of the node. One of the false positives has been marked with a red circle in Fig. 4 in sample number 125. As it can be seen, a false positive produces an oscillation in the average value of the reputation due to the fact that the CMA is reset (see the graph in the top left in Fig. 4). Since the variance of CMA decreases inversely proportional to the number of samples (see Eq. (10)), when the CMA is reset, its variance is the same than the variance of the reputation samples, and it decreases as we average more reputation samples. In Fig. 5, we get similar results using a value of $\beta = 0.1$. Increasing the value of β to 0.3 (see Fig. 6), we are in the limit of detection. However, for $\beta = 0.7$ (Fig. 7) and $\beta = 0.9$ (Fig. 8), the peak-to-noise ratio is lower so that no change in behavior occurs for node 1 and the averaging is slower. As already mentioned in the previous section, high β values worsen the peak-to-noise ratio, which makes it difficult to detect a change in the node's behavior.

The value of the threshold F must be set experimentally. Similar problems can be found in the literature (e.g., control charts [7]) and researchers adjust this parameter based on experiments. To do so, several simulations have been performed with $\beta = 0.1$, changing the detection threshold value F in each simulation: $F = 1$ in Fig. 9, $F = 2$ in Fig. 5, and $F = 3$ in Fig. 10. We can see that low F thresholds produce false positives that generate small oscillations in the averaging of the reputation of the nodes (see the top graphs in Fig. 9). Despite this, it is preferable to detect false positives than not being able to detect the behavior change of the nodes. In case that false positives always occur, reputation samples are not averaged, since the CMA is constantly reset, but the changes of behavior are always detected. Therefore, it is more convenient to use low F values. On the other hand, in Fig. 10, we can see that higher values for F do not allow to detect the change of the behavior of the node with noisier reputation samples, and its reputation is not updated properly. As mentioned above, since node 4 only participates in one path, its reputation samples are less noisy than those of the other nodes, being possible in this case to use higher F thresholds in order to avoid the risk of committing false positives (what happens for low F values). After many representative simulations we have conducted, we have obtained an adequate value for the threshold $F = 2$.

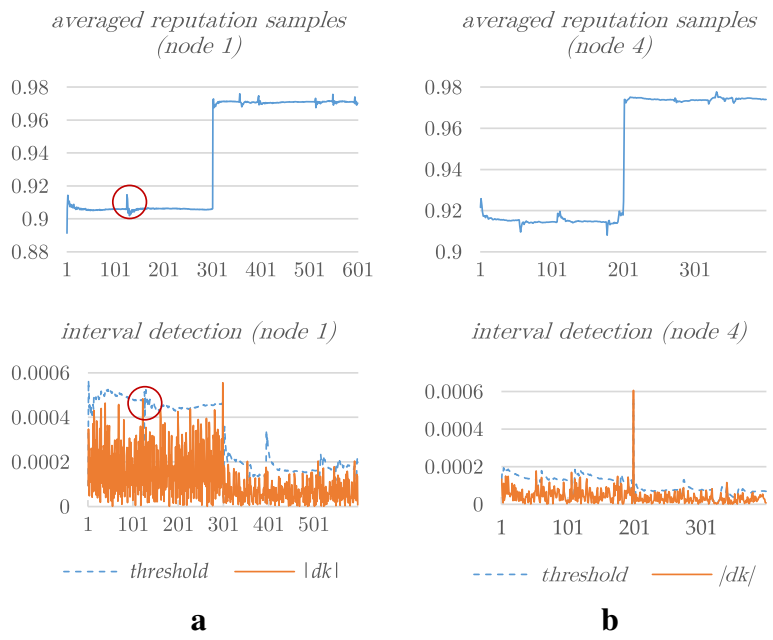

Fig. 4 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.01$ and $F = 2$: **a** for node 1 and **b** for node 4

5.2 Performance evaluation of the proposed routing protocol

To verify the operation of the proposed routing algorithm, a simulator of a MANET has been developed using Matlab R2017b [27]. The simulator implements DSR and also our proposed DrepSR forwarding algorithm. This way, we can compare the performance of DSR in both cases, either by using the minimum number of hops (vanilla DSR) or, conversely, by using our algorithm to select the route with the highest reputation (DrepSR). To perform the simulations, ten independent scenarios of 200 m × 200 m have been generated with 16 nodes

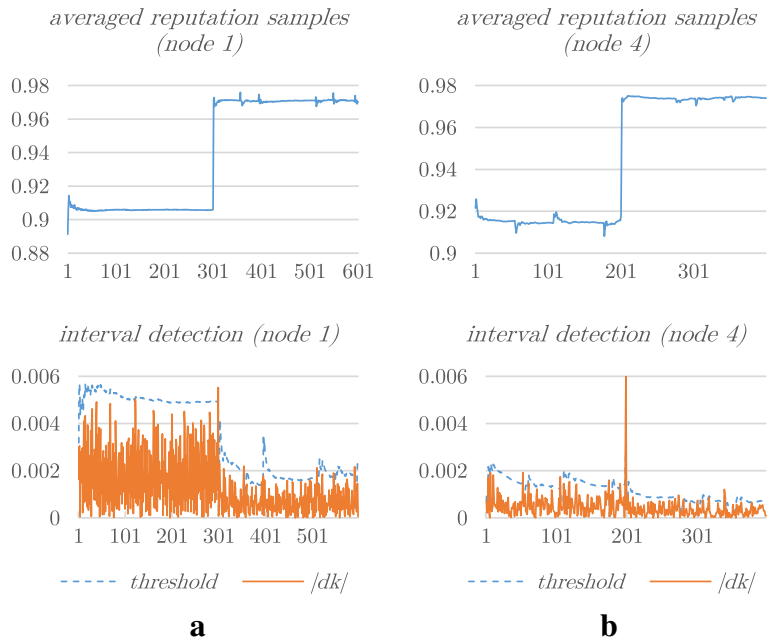

Fig. 5 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.1$ and $F = 2$: **a** for node 1 and **b** for node 4

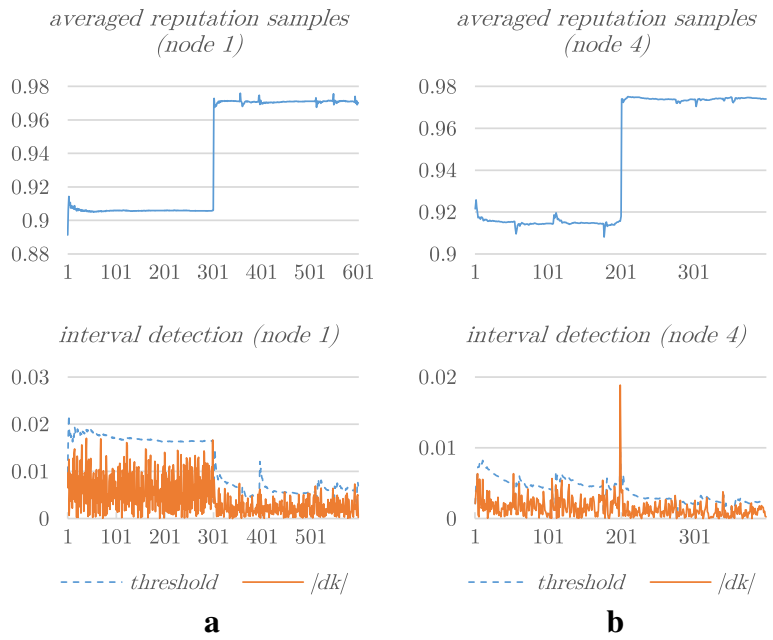

Fig. 6 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.3$ and $F = 2$: **a** for node 1 and **b** for node 4

following a random waypoint movement pattern. Half of the nodes have been configured so that they behave selfishly by dropping packets they should forward. Here, we show results with selfish nodes that only forward 50% of the packets they should forward. The simulation settings are shown in Table 1.

For each simulation, we have obtained the packet loss probability in the MANET and the average number of hops, both for DSR and DrepSR routing protocols. These results are shown in Table 2.

Averaging the ten simulations, we obtain that the proposed DrepSR reduces losses by 20% (DSR shows a

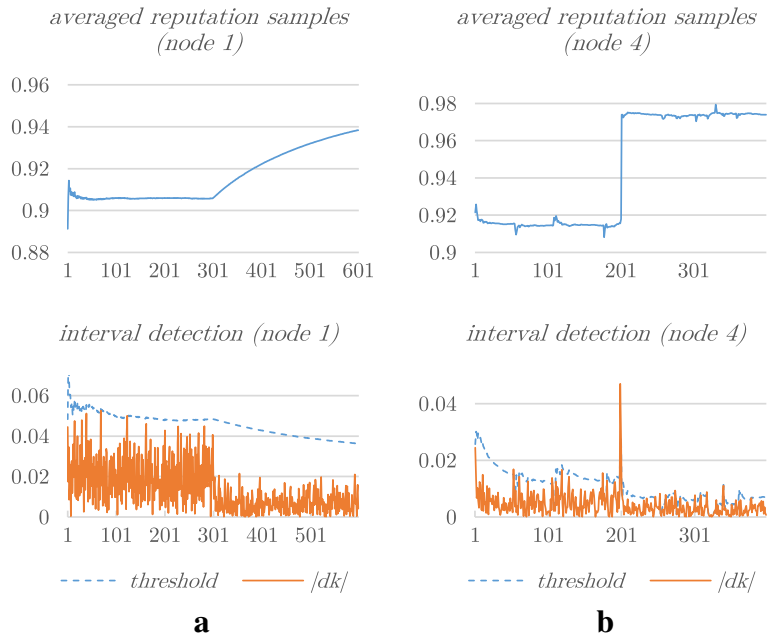

Fig. 7 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.7$ and $F = 2$: **a** for node 1 and **b** for node 4

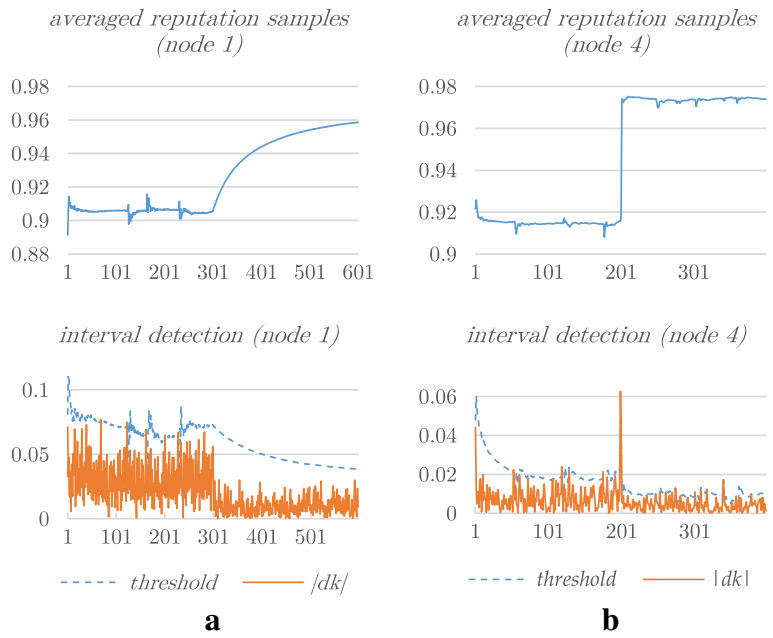

Fig. 8 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.9$ and $F = 2$: **a** for node 1 and **b** for node 4

packet loss probability of 21% and DrepSR 16%) with respect to DSR. Also, we can see that the number of hops barely increases and can be considered negligible. The average of the results and the 95% confidence interval are shown in Fig. 11.

Optionally, researchers can use open source datasets for MANETs to carry out the performance evaluation of proposals, like the one presented in [28]. This kind of traces generated from real experiments can be very useful to evaluate protocols and algorithms accurately.

6 Conclusion and future work

In this paper, we have presented a novel forwarding strategy for ad hoc routing protocols called DrepSR.

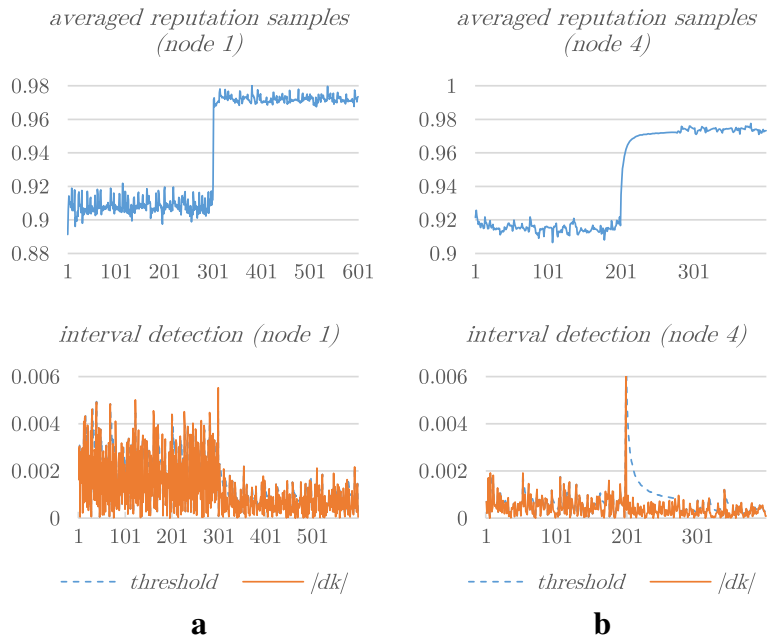

Fig. 9 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.1$ and $F = 1$: **a** for node 1 and **b** for node 4

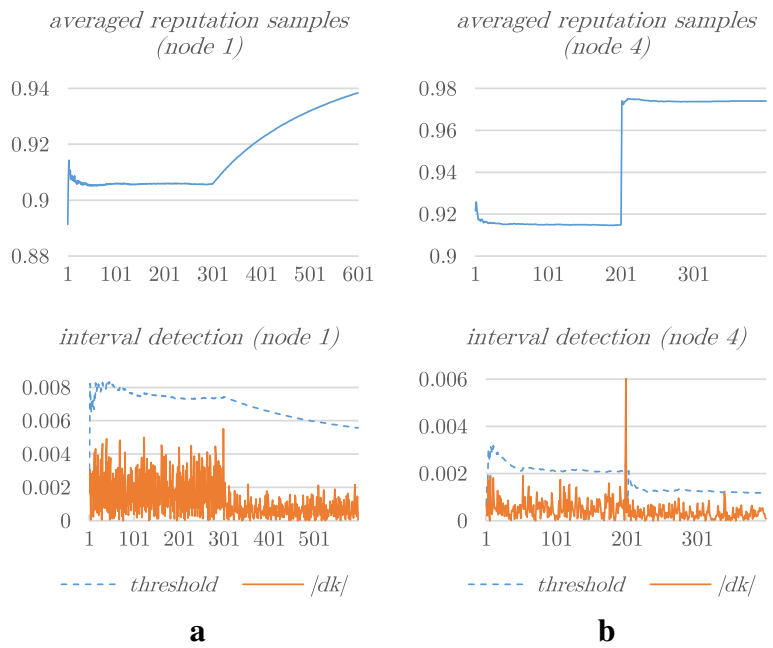

Fig. 10 Average value of the samples (figures at the top) and interval detection (figures below) when $\beta = 0.1$ and $F = 3$: **a** for node 1 and **b** for node 4

DrepSR is a modification of the vanilla DSR that includes our novel reputation-based forwarding algorithm. Our approach is based on the reputation of MANET nodes where we consider the presence of selfish nodes which do not cooperate in the routing tasks. Each source node uses the well-known DSR protocol as the routing engine to discover the available routes to destination. Then, the source node estimates the reputation of each path selecting the most reputable one using only first-hand information.

The advantage of this methodology is its simplicity, and although it may not calculate the real reputation of the single nodes, it is able to find those paths with less losses, which was our initial goal. Our main objective is to minimize the packet loss ratio when sending packets through different routes. Therefore, it is natural to use the observed packet loss ratio of each node as a metric of its reputation. Of course, packet losses may not always be caused by the selfish behavior of the nodes, but could also be due to traffic congestion or the random nature of wireless communications, or even because of the limited capabilities of nodes. However, the actual cause of a packet loss is not captured by our approach and we suppose it is always related to a selfish behavior, directly affecting the reputation of the nodes involved. We would like to highlight that although our reputation metric is not an accurate measure of selfish behavior for any individual node, it

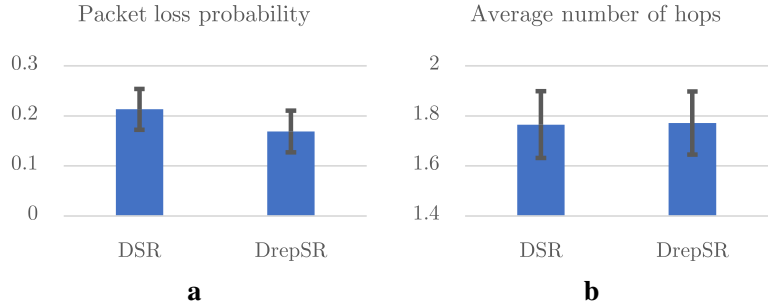

Fig. 11 Comparison of the simulated results for the DSR protocol and the proposed DrepSR routing protocol. **a** Packet loss probability; **b** Average number of hops. Figures show a 95% confidence interval using 10 independent realizations per point

Table 1 Simulation settings

Parameter	Value
Simulation time	1000 s
Number of nodes	16
Movement pattern	Random waypoint
Area	200 m × 200 m
Minimum speed	0.5 m/s
Maximum speed	1.5 m/s
Maximum pause time	60 s
Transmission range	120 m
Traffic type	CBR at 20000 bps
β	0.1
F	2
Number of selfish nodes (50% losses)	8
Source nodes (destinations)	1 (2, 3, 4, 5, 6, 11)
	2 (3, 7, 12)
	3 (4, 8,12)
	4 (5, 9, 14)
	5 (1, 10, 15)

is a good estimation that helps choosing the best available routing path.

We have validated our theoretical results through extensive simulations using Matlab R2017b [27]. After validation, the novel strategy was implemented over DSR using Matlab R2017b and many simulations were carried out for different scenarios with mobile nodes. We compared the probability of packet loss and the average number of hops for the DSR protocol (routing using the shortest path) and the DrepSR protocol (routing using the most reputable path). Results show that DrepSR reduces the probability of packet losses by 20%, while it slightly increases the average number of hops by 8%.

Table 2 Simulation results

Scenario	DSR		DrepSR	
	Packet loss probability	Mean number of hops	Packet loss probability	Mean number of hops
1	0.1729	1.6431	0.13	1.6632
2	0.2514	1.9605	0.2224	1.9624
3	0.2034	1.7846	0.1578	1.8177
4	0.2562	1.8872	0.2023	1.904
5	0.186	1.7013	0.1521	1.6515
6	0.2137	1.6815	0.1642	1.6936
7	0.1424	1.5611	0.1158	1.6012
8	0.1693	1.6354	0.1241	1.6498
9	0.3696	2.248	0.3243	2.2141
10	0.1671	1.5476	0.0952	1.5557

As a future work, we are planning to use Expectation Maximization model to accurately estimate the paths' reputation and compare it to our proposed strategy developed in this work. We are also considering the use of open source datasets for MANETs to carry out the performance evaluation of our protocol and algorithms.

1 Appendix

Table 3 Notation and symbols

r_n	Node reputation for node n
R_p	Path reputation for path p
L_p	Packet loss probability for path p
$\widehat{r_n}$	Sample of estimated reputation for node n
N_p	Number of intermediate nodes in path p
r_k	k-th sample of a node reputation
$\bar{r}$	Mean value of r_k
Δr_k	Deviation from the mean value of the sample r_k
σ_r	Standard deviation of r_k
c_k	Cumulative moving average (CMA) of the k samples of r_k
$\overline{c_k}$	Mean value of c_k
Δc_k	Deviation from the mean value of c_k
σ_c	Standard deviation of c_k
w_k	Exponentially weighted moving average (EWMA) of the k samples of r_k
β	Weight for the next sample to be averaged in the EWMA
w_0	Initial value for the EWMA
$\overline{w_k}$	Mean value of w_k
Δw_k	Deviation from the mean value of w_k
σ_w	Standard deviation of w_k
f_k	First difference of the w_k values
d_k	Second difference of the w_k values

Abbreviations

AODV: Ad hoc on-demand distance vector; CMA: Cumulative moving average; CONFIDANT: Cooperation of nodes, fairness in dynamic ad hoc networks; CPU: Central processing unit; DrepSR: Dynamic reputation-based source routing; DSR: Dynamic source routing; EWMA: Exponentially weighted moving average; MANET: Mobile ad hoc network; MN: Mobile node; NCS: Neighborhood compressive sensing; OCEAN: Observation-based cooperation enforcement in ad hoc networks; PFM: Positive feedback message; RTCP: Real Time Control Protocol; RTP: Real Time Protocol; TCP: Transmission Control Protocol; TV-AODV: Trust vector- ad hoc on-demand distance vector; TV: Trust vector; TV-DSR: Trust vector- dynamic source routing; UDP: User Datagram Protocol

Acknowledgements

Not applicable

Funding

This work was supported in part by the Spanish Government under projects INRISCO (TEC2014-–54335-C4-1-R) and MAGOS (TEC2017-84197-C4-3-R).

Lenin Guaya Delgado is the recipient of a full scholarship from the Secretaria Nacional de Educación Superior, Ciencia y Tecnología (SENESCYT).

Availability of data and materials
Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Authors' contributions
LGD implemented the code in the simulator, conducted all the simulations, and carried out the analysis of results. EP developed the proposal of the study, guided the entire research process as well as the mathematical analysis of the proposal, and took care of most of the writing. AMM collaborated in the development of the mathematical models and the writing and the analysis of results. JF made detailed corrections of the manuscript and assisted with the review of the simulation results. All authors participated in the discussions to develop the proposal and made a general review of the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Department of Network Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. [2]Department of Electrical and Computer Engineering, University of New Brunswick (UNB), Fredericton, Canada.

Received: 28 September 2018 Accepted: 21 February 2019
Published online: 25 March 2019

References
1. A. Duran, C. Shen, Mobile ad hoc P2P file sharing. Proc. IEEE Wireless Comm. and Networking Conf. (WCNC) **1**, 114-119 (2004)
2. S. Goel, T. Imielinski, K. Ozbay, Ascertaining viability of WiFi based vehicle-to-vehicle network for traffic information dissemination. Proc. IEEE Int'l Conf. Intelligent Transportation Systems (ITS), 1086-1091 (2004)
3. H. Wu, R. Fujimoto, R. Guensler, M. Hunter, MDDV: mobility-centric data dissemination algorithm for vehicular networks. Proc. ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET), 47-56 (2004)
4. A.M. Mezher, M.A. Igartua, L.J. de la Cruz Llopis, E. Pallarès-Segarra, C. Tripp-Barba, L. Urquiza-Aguiar, J. Forné, E.S. Gargallo, A multi-user game-theoretical multipath routing protocol to send video-warning messages over mobile ad hoc networks. Sensors **15**, 9039–9077 (2015)
5. H. Gharavi, Multichannel mobile ad hoc links for multimedia communications. Proc. IEEE **96**(1), 77–96 (2008)
6. S. Marti, T.J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks. Proc. ACM MobiCom, 255-265 (2000)
7. C. Douglas, *Montgomery, "Introduction to statistical quality control", 6th Edition* (Wiley, New York, 2009) ISBN: 978–0–470-16992-6
8. D.B. Johnson, D.A. Maltz, J. Broch, in *Ad hoc networking*. DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks (Addison-Wesley Professional, Boston, 2001), pp. 139–172
9. L. Buttyán, J.P. Hubaux, Stimulating cooperation in self-organizing mobile ad hoc networks. Mobile Netw. Appl. **8**, 579–592 (2003). https://doi.org/10.1023/A:1025146013151
10. M. Jakobsson, J.P. Hubaux, L. Buttyán, *A micro-payment scheme encouraging collaboration in multi-hop cellular networks* (International Conference on Financial Cryptography, LNCS Springer, Berlin, 2003), pp. 15–33
11. M.E. Mahmoud, X. Shen, FESCIM: fair, efficient, and secure cooperation incentive mechanism for multihop cellular networks. IEEE Trans. Mob. Comput. **11**(5), 753–766 (2012). https://doi.org/10.1109/TMC.2011.92
12. E. Chiejina, H. Xiao, B.A. Christianson, in *Proceedings of the 6th York Doctoral Symposium on Computer Science & Electronics, York, UK, 2013*. Candour-based trust and reputation management system for mobile ad hoc networks (University of York, York, 2013)
13. S. Marti, T.J. Giuli, K. Lai, M. Baker, *Mitigating routing misbehavior in mobile ad hoc networks* (Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, 2000), pp. 255–265
14. S. Buchegger, J.Y. Le Boudec, in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Lausanne, Switzerland, 9*. Performance analysis of the CONFIDANT protocol (ACM, New York, 2002), pp. 226–236
15. P. Michiardi, R. Molva, in *Proc. CMS*. Core: a collaborative reputation mechanism to enforce node cooperation in MANETs (2002), pp. 107–121
16. S. Bansal, M. Baker, *Observation-based cooperation enforcement in ad hoc networks* (CS Dept., Stanford Univ., Stanford, 2003) Tech Rep
17. N. Li, S.K. Das, A trust-based framework for data forwarding in opportunistic networks. Ad Hoc Netw. **11**, 1497–1509 (2013)
18. A. Banerjee, S. Neogy, C. Chowdhury, in *Proceedings of the 2012 Third International Conference on Emerging Applications of Information Technology (EAIT), West Bengal, India*. Reputation based trust management system for MANET (2012), pp. 376–381
19. S.G.C. Kumar, A novel routing strategy for ad hoc networks with selfish nodes. J. Telecommun. **3**, 23–28 (2010)
20. W. Gong, Z. You, D. Chen, X. Zhao, M. Gu, K.Y. Lam, Trust based routing for misbehavior detection in ad hoc networks. J. Netw. **5**, 551–558 (2010)
21. C.E. Perkins, E.M. Royer, in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*. Ad-hoc on-demand distance vector routing (New Orleans, 1999), pp. 90–100, IEEE Computer Society.
22. K.A.A. Bakar, J.A. Irvine, in *Proceedings of the 2010 6th International Conference on Wireless and Mobile Communications* (ICWMC), Scheme for detecting selfish nodes in MANETs using OMNET++, published by Conference Publishing Services (CPS), (Valencia, 2010), pp. 410–414
23. C. Tang, A. Li, X. Li, When reputation enforces evolutionary cooperation in unreliable MANETs. IEEE Trans. Cybernetics **45**(10), 2190–2201 (2015). https://doi.org/10.1109/TCYB.2014.2366971
24. F.B. Saghezchi, A. Radwan, J. Rodriguez, Energy-aware relay selection in cooperative wireless networks: An assignment game approach. Ad Hoc Netw. **56**, 96–108 (2017). https://doi.org/10.1016/j.adhoc.2016.12.001.
25. E. Chiejina, H. Xiao, B. Christianson, A dynamic reputation management system for mobile ad hoc networks. Computers **4**, 87–112 (2015)
26. M.A. Khusru Akhtar, G. Sahoo, Enhancing cooperation in MANET using neighborhood compressive sensing model. Egyptian Inform. J. (2016). https://doi.org/10.1016/j.eij.2016.06.007
27. MathWorks. Available online: https://www.mathworks.com/. Accessed 9 July 2018.
28. A. Karygiannis, K. Robotis, E. Antonakakis, in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)*. Creating offline MANET IDS network traces (2007). https://doi.org/10.1109/WOWMOM.2007.4351704