REPSYS: A Robust and Distributed Reputation System for Delay-Tolerant Networks

Naercio Magaia, Paulo Pereira, Miguel Correia
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
naercio.magaia@tecnico.ulisboa.pt,prbp@inesc.pt,miguel.p.correia@tecnico.ulisboa.pt

ABSTRACT

Distributed reputation systems can be used to foster cooperation between nodes in decentralized and self-managed systems due to the nonexistence of a central entity. In this paper, a Robust and Distributed Reputation System for Delay-Tolerant Networks (REPSYS) is proposed. REPSYS is robust because despite taking into account first- and second-hand information, it is resilient against false accusations and praise, and distributed, as the decision to interact with another node depends entirely on each node.

Simulation results show that the system is capable, while evaluating each node's participation in the network, to detect on the fly nodes that do not accept messages from other nodes and that disseminate false information even while colluding with others, and while evaluating how honest is each node in the reputation system, to classify correctly nodes in most cases.

CCS CONCEPTS

• **Security and privacy** → *Trust frameworks*; • **Networks** → *Network simulations*; *Ad hoc networks*;

KEYWORDS

Reputation, Trust, Bayesian, Delay-Tolerant Network

1 INTRODUCTION

Delay-Tolerant Networks (DTNs) [8] are networks in which end-to-end connectivity between a source and target node is not guaranteed due to nodes' mobility or even because nodes can join or leave the network, for example, as a result of devices being turned off or running out of battery. The DTN routing strategy allows messages to be relayed among nodes until the destination is reached, or they are discarded.

To manage and organize decentralized and self-managed systems, *incentive schemes* [11] can be used, hence compensating for the nonexistence of a central or dedicated entity, e.g., for managing reputation and trust. In a *distributed reputation-based incentive scheme* [11], hereafter *reputation scheme*, which is more suitable for DTNs as no central authority is available, nodes' ratings are stored in a distributed fashion and the evaluation of reputation is based on subsets of information (e.g., information provided by neighbor nodes). In such systems, nodes collect reputation information. A reputation system that relies exclusively on first-hand information (direct evidence) may not take advantage of all the available information. However, second-hand information (indirect evidence) should be used with care since negative information, i.e., false accusations or praise, may be used to deceive the system. The collected information is evaluated in order to decide if the node should cooperate or not, based on the other node's reputation. After the nodes' interaction, the degree of cooperation between them is evaluated aiming to reward nodes that presented a good behavior by adequately increasing their reputation. As a result, nodes with bad reputation are isolated henceforward not receiving others' services.

This paper proposes REPSYS, a Robust and Distributed Reputation System for Delay-Tolerant Networks. REPSYS is both robust against false ratings and efficient at detecting nodes' misbehavior. It is based on a Bayesian approach that uses the Beta distribution. REPSYS can be integrated with any DTN routing protocol. It uses special feedback messages taking into account the network density. REPSYS proposed a new approach for obtaining first-hand information based on the attack signature and a modified decision criterion to avoid misclassification that takes into account recent observations.

2 RELATED WORK

Many reputation schemes have been proposed, but only a few for DTNs. In [3], a cooperative watchdog system to support selfish nodes detection was proposed. Each time a node participated in a contact opportunity, a reputation score was assigned to him. The proposed classification module does not learn as new observations are available. In [9], a Bayesian trust-based framework that can be integrated with single-copy data forwarding protocols was proposed. The proposed special message is not adequate for sparse DTNs. In [12], a Bayesian approach where each node also manages its reputation evidence and demonstrates it whenever necessary was proposed. Second-hand information was not considered in the latter. An iterative trust management and distributed malicious node detection mechanism for DTNs, where only the behavior of the nodes in terms of routing is evaluated, was proposed in [1]. A modified Bayesian approach for reputation and trust representation and update, and for second-hand information integration was proposed in [2]. Despite being similar to REPSYS, it was envisioned for MANET and P2P routing protocols therefore not being suitable for DTNs. Moreover, an offline classifier was used.

3 THE REPSYS SYSTEM

REPSYS is both robust against false ratings and efficient at detecting nodes' misbehavior. REPSYS is robust because despite taking into

MSWiM'17, November 21–25, 2017, Miami, FL, USA

DOI: http://dx.doi.org/10.1145/3127540.3127573

account all the available information, it is resilient against false accusations and praise, and distributed, as the decision to interact with another node depends entirely on each node. It takes into account all the available information and uses Bayesian decision theory to classify nodes. REPSYS is based on a Bayesian approach that uses the Beta distribution, and can be integrated with any DTN routing protocol. There are three modules in REPSYS: reputation module (reputation collection module, reputation evaluation module), trust module and routing decision module (that uses Bayesian classification).

3.1 Assumptions

There is a network with several nodes, i.e., wireless devices held by people or in vehicles that may be moving. Each node has a Unique IDentifier (UID) that cannot be spoofed. Each node can only monitor its one-hop neighbors, i.e., can only monitor nodes that are directly connected to him. Akin to benign nodes, malicious nodes are also wireless devices. However, they may deviate from the protocol in the following ways: (*i*) *lying attacks (liars),* nodes that not having received a message return wrong confirmation that they have it in their buffer. In addition, these nodes may disseminate false first-hand information; (*ii*) *black-hole attacks*, nodes that do not forward others’ messages; and (*iii*) *collusion attacks*, where nodes may forward data to each other to earn reputation. The intensity of individual attacks can be augmented by collusion.

3.2 The modified Bayesian approach

Each node considers that there is a given parameter, θ, known as the state of nature such that another node misbehaves with probability θ, and that the outcome is drawn independently at each observation x. Furthermore, each node considers that there is a different θ for every other node. These parameters are unknown, hence modeled assuming that they are drawn according to a prior distribution, $\pi(\theta)$, which is updated as new observations become available.

The beta probability density function, $Beta(\alpha, \beta)$, is used as the prior as it represents probability distributions of binary events (e.g., good or bad) and the conjugate is also a Beta distribution [5]. The expectation of the Beta density is

$$\mathbb{E}\left[\text{Beta}\left(\theta|\alpha,\beta\right)\right] = \frac{\alpha}{\alpha+\beta} \tag{1}$$

The Bayesian process works as follows. Initially each node has the prior $Beta(1, 1)$, that is, the uniform distribution on $[0, 1]$, for all its neighbors. The $Beta(1, 1)$ prior represents absence of information as there are no observations. When a new observation is made, if a correct behavior is observed then $x = 1$; otherwise $x = 0$. The prior is updated according to $\alpha_{\text{new}} = \alpha_{\text{old}} + x$ and $\beta_{\text{new}} = \beta_{\text{old}} + (1 - x)$.

Due to the network dynamics, a node may change its behavior over time in contrast to the standard Bayesian framework that gives the same weight regardless of time of occurrence of the observation. The fading mechanism allows to forget gradually old observations, and it is defined as $y_\eta^\tau = y_\eta^{\tau-1}\eta + y^\tau$, where $y \in \{\alpha, \beta\}$ and y_η^τ is the accumulated rating of a given node at time period τ, y^τ is the new rating at time τ and η is the fading factor and $0 < \eta < 1$.

3.3 Information gathering

Each node is equipped with a pseudo-watchdog component that allows it to monitor the behavior of the neighbors with whom it interacts. Specifically, if node i forwards a message to node j, the behavior of j is evaluated in terms of two types of evidence, namely: (*i*) if j accepts messages of i and, (*ii*) if j forwards i’s messages of another node, say k. The former evidence is collected through direct communication between two nodes (i.e., through experience), meanwhile the latter, is through *Special Feedback Messages* (SFMs). Therefore, i waits for a SFM. Two types of SFMs, that take into account the network density, are proposed: (*i*) *type-1* that is created by k, which is 2 hops away from node i (which can be the source or forwarder of the message); and (*ii*) *type-2* that is created by the destination of the message. SFM *type-1* and shall suffice for dense networks. Each SFM contains the message identifier, the list of nodes the message traversed and the message digest.

The first-hand information represents the parameters of the Beta distribution assumed by node i in its Bayesian opinion of node j’s behavior in the network. Each node keeps two data structures (records): *accept first-hand information* ($\mathcal{F}_{a_{\text{ij}}}$) for accepted messages and *forward first-hand information* ($\mathcal{F}_{f_{\text{ij}}}$) for forwarded messages. For each record there are two counters: α and β. Accept and forward first-hand information are given by $\mathcal{F}_{\mathfrak{x}_{\text{ij}}} = (\alpha_{\mathfrak{x}}, \beta_{\mathfrak{x}}) = (\alpha, \beta)_{\mathfrak{x}}$, where $\mathfrak{x} \in \{a, f\}$, and they are updated to identify *attacks’ signature* as follows: (*a*) α is incremented if a good behavior is observed when: (*i*) node j accepts messages of other nodes, e.g., node i. However, nodes that only accept messages may be performing *black-hole attack*s. Therefore, it is also necessary to ensure that node j forwards messages that it receives if the message is not destined to him; or (*ii*) node i receives a SFM from k because of a message i forwarded to j. It is assumed that among all neighbors of j, node k’s delivery likelihood to the destination is the highest one; (*b*) β is incremented if a misbehavior is observed when: (*i*) node j not being the destination of a message sent by node i, does not forward the message (no SFM was received neither did the message expire); or (*ii*) node j does not accept messages of other nodes, e.g., node i, may be an indication that j is performing a *lying attack.* Node j can only refuse to accept messages forwarded to him if he already has them in buffer. Moreover, node j must prove to node i that he has the message in buffer as follows: node i sends a message containing the message identifier (MID) and a nonce (N) to node j. If j has the message, it must reply with a digital signature containing the digest of the message with identifier MID and N.

Second-hand information corresponds to first-hand information published by other nodes. For instance, node i can gather node k’s first-hand information towards node j. Similarly to first-hand information, each node keeps two records: *accept second-hand information* ($\mathcal{S}_{a_{\text{ij,k}}}$) and *forward second-hand information* ($\mathcal{S}_{f_{\text{ij,k}}}$). Second- and first-hand information are related according to $\mathcal{S}_{\mathfrak{x}_{\text{ij,k}}} = \mathcal{F}_{\mathfrak{x}_{\text{kj}}}$.

3.4 Reputation rating

A reputation rating $\mathcal{R}_{\text{ij}}$, which is managed by the reputation module, is updated (*i*) when first-hand information is updated, and (*ii*) when received second-hand information is considered valid to be incorporated.

If accept and forward first-hand information that are kept by each node are available, they are combined to form a unique first-hand information, hereafter called first-hand information $\mathcal{F}_{\text{ij}} = (\alpha, \beta)_{\mathcal{F}}$, as follows: (*i*) if $\alpha_f > \alpha_a$ then $\alpha_{\mathcal{F}} = \alpha_f$. Otherwise, $\alpha_{\mathcal{F}} = \alpha_a$; (*ii*) $\beta_{\mathcal{F}} = \max(\beta_a, \beta_f)$. For replication-based approaches, an optimizations is proposed to penalize nodes that accept more messages than they forward: if $\alpha_a > \alpha_f$, $\alpha_a = \chi$ and $\alpha_f = 1$, increase $\beta_{\mathcal{F}}$ and decrease α_a. χ represents the number of evidences of accepted messages a node has while not having any evidence of messages that the node forwarded of other nodes.

The first-hand information record, which contains two counters, is never published since it is considered private. What is published is the first-hand information rating that is computed using Eq. 1.

When first-hand information is updated, an exponential weighted moving average (EWMA) is used to allow for reputation fading as follows

$$\mathcal{R}^{\tau}_{\text{ij}} = (1 - \phi)\mathcal{R}^{\tau-1}_{\text{ij}} + \phi\mathcal{F}^{\tau}_{\text{ij}} \tag{2}$$

where ϕ is the smoothing factor and $0 < \phi < 1$. Please note that first-hand information is equal to the accept first-hand information on the absence of forward first-hand information.

When received second-hand information is considered valid to be incorporated, linear opinion pooling [4] is used for its integration. Assume two nodes *i* and *k* where *i* has its opinion on how honest node *k* is as an actor in the reputation system (i.e., the trust rating node *i* has on *k*, $\mathcal{T}_{\text{ik}}$), and *k* collects first-hand information about node *j*. A *recommendation* then consists in combining *i*'s opinion about *k* with *k*'s opinion about *j* in order for *i* to get its opinion about *j*. It resembles trust transitivity [6]. If *i* considers *k* trustworthy based on $\mathcal{T}_{\text{ik}}$, $\mathcal{F}_{\text{kj}}$ is used by node *i* for updating $\mathcal{R}_{\text{ij}}$ after performing the deviation test (Eq. 4) according to

$$\begin{aligned} \mathcal{R}^{\tau}_{\text{ij}} &= w_1\mathcal{R}^{\tau-1}_{\text{ij}} + w_2\mathcal{F}^{\tau}_{\text{kj}} \\ \omega_2 &= \varsigma\mathcal{T}_{\text{ik}},\ 0 < \varsigma < 1 \end{aligned} \tag{3}$$

where w_1 and w_2 are fixed non-negative weights with sum-total 1 and ς is the node's individuality factor. If ς is less than $\frac{1}{2}$, it means that a node trusts more its own experience hence guaranteeing that second-hand information carries less weight than first-hand information.

Moreover, first-hand information received from highly trusted nodes should carry more weight that the one received from nodes with low trust ratings. $\mathcal{T}_{\text{ik}}\mathcal{F}_{\text{kj}}$ allows to discount the first-hand information received as a function of the trust rating of the node that provided the information.

If *i* considers *k* untrustworthy, the accept and forward deviation tests are performed. The *deviation test* is computed as follows

$$\left|\mathcal{S}_{\mathfrak{x}_{\text{ij,k}}} - \mathcal{F}_{\mathfrak{x}_{\text{kj}}}\right| \geq d \tag{4}$$

where d is the deviation threshold. The deviation test allows comparing if nodes *i* and *k* have similar opinions about *j*.

If the results of accept and forward deviation tests are both negative, $\mathcal{F}_{\text{kj}}$ is incorporated using Eq. 3. Otherwise, (*i*) if both are positive, $\mathcal{F}_{\text{kj}}$ is not incorporated; (*ii*) if at least one of them is positive, $\mathcal{F}_{\text{kj}}$ is incorporated at most twice since one of the deviation tests mostly probably failed because of stale recommendations.

Any node *k*'s recommendations towards *j* are synthesized using the same moving average process as in Eq. 2, thus making the system resilient against false praise and accusation. Is it assumed that there is an acceptable number of misbehaving nodes. Second-hand information is integrated using

$$\mathcal{S}^{\tau}_{\mathfrak{x}_{\text{ij,k}}} = (1 - \phi)\,\mathcal{S}^{\tau-1}_{\mathfrak{x}_{\text{ij,k}}} + \phi\mathcal{F}^{\tau}_{\mathfrak{x}_{\text{kj}}} \tag{5}$$

3.5 Trust rating

The trust record, which is stored at the trust module, has also the form $\mathcal{T}_{\text{ij}} = (\alpha, \beta)_{\mathcal{T}}$. As it was previously mentioned, $(\alpha, \beta)_{\mathcal{T}}$ represents the parameters of the Beta distribution assumed by node *i* in its opinion about how honest node *j* is as an actor in the reputation system. When node *i* receives first-hand information from some node *k* about node *j*, an update is performed.

Prior to incorporating the second-hand information, a deviation test is executed. On the one hand, it is used to update the trust rating node *i* has of *k*, and on the other hand, in addition to the latter, it is also used to decide whether to update the reputation rating node *i* has on *j*. $\alpha_{\mathcal{T}}$ is incremented if both deviation tests are positive. If both deviations tests are negative or if at least one is positive and $\mathcal{F}_{\text{kj}}$ was incorporated at most twice, then $\beta_{\mathcal{T}}$ is incremented. The trust rating is computed using Eq. 1.

3.6 Bayesian classification

In classification problems, Θ is discrete and the goal is to estimate θ given an observation x. To address the task of finding suitable nodes to forward messages in DTNs, two binary classification problems are considered: the node's behavior ($\mathfrak{P}_1$) and trustworthiness ($\mathfrak{P}_2$) classification problems.

Let $\theta \in \Theta = \{\theta_0, \theta_1\}$ be the unknown state of nature: for $\mathfrak{P}_1$: $\theta = \{\theta_0 = \text{good/normal}, \theta_1 = \text{bad/misbehaving}\}$ and for $\mathfrak{P}_2$: $\theta = \{\theta_0 = \text{trustworthy}, \theta_1 = \text{untrusworthy}\}$. Let $X \in \mathcal{X}$ be a random variable with $\{f(x|\theta),\ x \in X\}$. Let $\pi(\theta) > 0$ and $\sum_{\theta\in\Theta}\pi(\theta) = 1$ be the prior probability mass function. Let $a \in \mathcal{A} = \{a_0, a_1\}$ be the allowed decision or action: for $\mathfrak{P}_1$: $a = \{a_0 = \text{FORWARD}, a_1 = \text{DO\_NOT\_FORWARD}\}$ and for $\mathfrak{P}_2$: $a = \{a_0 = \text{TRUST},\ \text{a}_1 = \text{DO\_NOT\_TRUST}\}$. Let the "0/1" loss function be used for classification. It assigns zero cost to any correct decision and unit cost to any wrong decision. The optimal Bayesian decision is given by

$$\delta_{\text{Bayes}}(x) = \begin{cases} \theta_0 \longleftarrow l(x) \geq t \\ \theta_1 \longleftarrow l(x) < t \end{cases} \tag{6}$$

where $l(x) = \frac{f(x|\theta_0)}{f(x|\theta_1)}$ is the likelihood ratio and $t = \frac{\pi(\theta_0)}{\pi(\theta_1)}$ is the decision threshold.

The likelihood function is given by the Bernoulli distribution $f(x|\theta) = \theta^r(1-\theta)^{n-r}$, where $r = \sum_{i=0}^{n} x_i$, and r denotes the number of outcomes representing correct behavior.

In the beginning, if the only information available is the conditional probability density function of the observation given the true θ, the maximum likelihood decision criterion (δ_{ML}) [5] is used. δ_{ML} is defined as

$$\delta_{\text{ML}} = \begin{cases} \theta_0 \longleftarrow l(x) \geq 1 \\ \theta_1 \longleftarrow l(x) < 1 \end{cases} \tag{7}$$

In the *node's behavior classification problem*, after each interaction between two nodes, the sender updates the reputation rating of the other node based on the result of this interaction. Each node clusters the other nodes to whom it interacted in two groups: normal nodes, if $\mathcal{R}_{\text{ij}} \geq 1/2$, and misbehaving nodes, if $\mathcal{R}_{\text{ij}} < 1/2$. The prior probabilities $\pi(\cdot)$ of these clusters, which allow determining the decision threshold, are coefficients of the convex combination of the number of nodes in these clusters. The optimal Bayesian decision is computed using Eq. 6 given the prior probabilities. However, if a correct behavior is observed and $\pi(\theta_1) > \pi(\theta_0)$, one may incur in false positives, i.e., a misclassification, while using the optimal Bayesian decision criterion, because of the higher weight of the decision threshold in comparison to the likelihood ratio.

A modified optimal Bayesian decision, which is an *online classifier*, is proposed as the workaround. It consists in finding attenuation parameters α and β of the *posterior mean Bayesian estimator* ($\hat{\theta}_{\text{PM}}$) [5] and computing an attenuated decision threshold. $\hat{\theta}_{\text{PM}}$ is given by

$$\hat{\theta}_{\text{PM}} = \frac{\alpha + r}{\alpha + \beta + n} \tag{8}$$

For the minimum possible case, i.e., one correct behavior being observed and two clusters, one with 2 misbehaving nodes and the other with 1 normal node, $l(x)$ is $4/3$. If $\alpha = \beta$, the Bayesian attenuation parameters are given by

$$\alpha \geq 3n - 7r \tag{9}$$

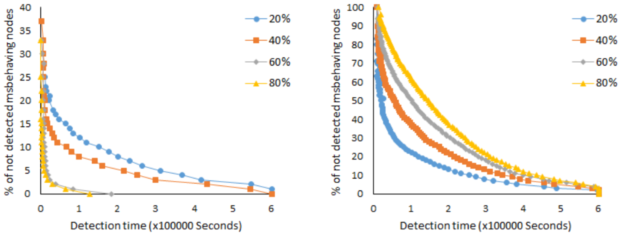
For the case above, $\alpha = 2$ and the decision threshold is equal to the likelihood ratio. If instead the *maximum a posteriori Bayesian estimator* [5] was used, the decision threshold would be greater than the likelihood ratio which would lead to misclassification.

In the *trustworthiness classification problem*, each node also clusters nodes that sent first-hand information to him in two groups: trustworthy, if $\mathcal{T}_{\text{ij}} > 1/2$, and untrustworthy, if $\mathcal{T}_{\text{ij}} < 1/2$, based on the result of the deviation test. The deviation test is performed after the bootstrapping of the trust module. During bootstrapping, nodes' recommendations are synthesized (Eq. 5). Ideally, the bootstrapping period should not be inferior to the time necessary for the distributed reputation system to converge, i.e., for each node's routing decision module engine to be able to classify correctly all the nodes with which it interacted.

In the same way to the node's behavior classification problem, the modified optimal Bayesian decision is computed (see Eqs. 8 and 9) to avoid misclassifications.

4 SIMULATION MODEL

REPSYS was implemented on the Opportunistic Network Environment (ONE) simulator [7]. The simulation model consisted of a network with 150 pedestrians. A map-based mobility model of the Helsinki City over an area of 4.5 × 3.4 Km was used. The pedestrians were moving in a speed varying between 0.8 to 1.4 m/s. The communication range between nodes was 10 m, and the communication is bidirectional at a constant transmission rate of 2 Mbit/s. Every 1 to 2 minutes, a source node randomly chosen can generate one message to a randomly chosen destination. Nodes do not change their behavior (malicious or not) over time. The TTL attribute of each message was 12 h, and the message size varies from 500 kB to 1 MB. The pedestrians' devices had a buffer size of 20 MB for DTN traffic.The simulation time was 7 days with an update interval of 1.0 s. The deviation threshold was set to $1/6$. The individuality factor was set to $2/3$, which means that first-hand information weights $2/3$, that is, the double of the second-hand information weight, i.e., $1/3$. The same goes to the smothing factor. The nodes misbehavior considered for evaluation were liars and black-hole attacks. It was considered that misbehaving nodes were also colluding, i.e., they increased α of misbehaving nodes and β of normal nodes. The effects of nodes' misbehavior was examined on Epidemic similarly to [10]. The percentage of liars and nodes that performed black-hole attacks varied from 20% to 80% with increments of 20%.



(a) Liars

(b) Black-hole

Figure 1: The time necessary to correctly classify misbehaving nodes as DO_NOT_FORWARD **for Epidemic with 20, 40, 60 and 80% of liars and black-hole nodes**

5 SIMULATION RESULTS

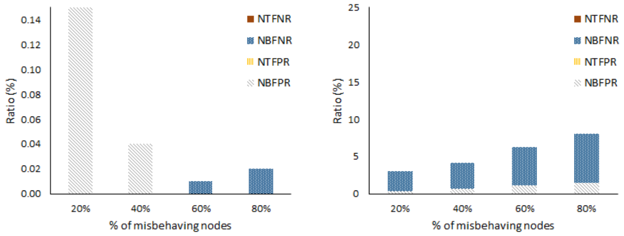
The evaluation of the performance of REPSYS consisted in appraising the reputation and trust modules, similarly to previous work [2]. Additionally, Bayesian classification at the routing decision module was also evaluated. For each setting, i.e., protocol-percentage pair, thirty independent simulations using different seeds were conducted, and the results averaged, for statistical confidence.

The following metrics were considered for the evaluation of REPSYS: *detection time of misbehaving nodes*, which corresponds to the simulation time that took all normal nodes to correctly classify all misbehaving nodes they came in contact with, starting at the detection instant of the first misclassification and *robustness* against false accusations (false negatives) and false praise (false positives), namely Node's Behavior False Positives Ratio (NBFPR), Node's Behavior False Negatives Ratio (NBFNR), Node's Trustworthiness False Positives Ratio (NTFPR) and Node's Trustworthiness False Negatives Ratio (NTFNR).

5.1 Detection time of misbehaving nodes

Figures 1 presents the time necessary for each good node to classify correctly all misbehaving nodes it met as DO_NOT_FORWARD for the Epidemic routing protocol considering four percentages of misbehaving nodes, i.e., 20, 40, 60 and 80%.

The detection time was directly influenced by the routing layer, i.e., the algorithm used to disseminate messages across the network. Ideally, the goal of any reputation system would be to correctly classify all the other nodes with whom a given node interacts (e.g., for



(a) Liars

(b) Black-hole

Figure 2: Node's behavior and Trustworthiness false positives and negatives ratios for Epidemic with 20, 40, 60 and 80% of liars and black-hole nodes

the simplest case, if the node accepted and forwarded the message it received) with the least possible number of contacts. However, overhead causes nodes to interact many times with the same node or group of nodes.

Epidemic's performance was most of the times affected by the overhead of the protocol, therefore increasing the detection time of liars. Nonetheless, the presence of liars improved the performance of Epidemic since less message copies were created, as liars did not accept them and by not accepting they were detected thus reducing the detection time. For the black-hole attack, REPSYS must penalize nodes that only accepted but did not forward messages given that evidence that these messages were not forwarded expired. Even if a small penalization was given, misbehaving nodes performing black-hole attacks would be detected. However, good nodes that behaved similarly to misbehaving nodes would be also isolated from the network, although temporarily, because of the fading mechanism or if they started forwarding messages.

For SFM *type-2*, since an evidence has, by default, the same TTL of a message that originated it, there is a tradeoff between the TTL and the detection time. If the goal is for REPSYS to converge sooner (i.e., to have a small detection time) then the TTL should not be too high. Otherwise, SFMs might not have enough time to be effectively disseminated over the network, which would increase the number of misclassifications as a consequence of a too small TTL. Nevertheless, REPSYS took more time to detect an increasing percentage of nodes performing black-hole attacks in contrast to liars where an increased number of liars took less time to detect, mainly because of forward first-hand information.

5.2 Robustness

In Figure 2, four metrics were considered to measure REPSYS's robustness against false accusations and praise for the lying and black-hole attacks. One can conclude, by analyzing these figures, that there were no misclassified bad nodes for the node's trustworthiness problem. The use of second-hand information may lead to false accusations and praise, but even with the optimal Bayesian decision criterion, it did not have any influence on the metrics considered. There are two reasons for that: (*i*) the bootstrapping of the trust module and (*ii*) the tolerance to nodes that failed the deviation test (Eq. 4). Additionally, there is also a tradeoff between false positives and negatives. By attempting to isolate misbehaving nodes (that is, to reduce the false positives ratio), good nodes that up to a given instant only accepted messages will be misclassified as DO_NOT_FORWARD, therefore increasing the ratio of false negatives.

6 CONCLUSIONS

In this paper, a robust and distributed reputation system for DTNs was proposed. REPSYS takes into account all the available information and uses Bayesian decision theory to classify nodes. The system is robust because despite taking into account all the available information, it is resilient against false accusations and praise, and distributed, as the decision to interact with another node depends entirely on each node.

ACKNOWLEDGMENTS

A warmly thanks to Mário Figueiredo for his numerous inputs. This work was partially supported by Fundação Calouste Gulbenkian and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

REFERENCES

[1] Erman Ayday and Faramarz Fekri. 2012. An Iterative Algorithm for Trust Management and Adversary Detection for Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing* 11, 9 (sep 2012), 1514–1531. DOI:https://doi.org/10.1109/TMC.2011.160

[2] Sonja Buchegger and Jean-Yves Le Boudec. 2004. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*. http://infoscience.epfl.ch/record/519

[3] J. A. F. F. Dias, J. J. P. C. Rodrigues, F. Xia, and C. X. Mavromoustakis. 2015. A Cooperative Watchdog System to Detect Misbehavior Nodes in Vehicular Delay-Tolerant Networks. *IEEE Transactions on Industrial Electronics* 62, 12 (Dec 2015), 7929–7937. DOI:https://doi.org/10.1109/TIE.2015.2425357

[4] Franz Dietrich and Christian List. 2016. Probabilistic Opinion Pooling. In *The Oxford Handbook of Probability and Philosophy*, Alan Hájek and Christopher Hitchcock (Eds.). Oxford University Press, Chapter 25, 832. https://global.oup.com/academic/product/the-oxford-handbook-of-probability-and-philosophy-9780199607617?cc=pt

[5] Mário A. T. Figueiredo. 2004. *Lecture notes on Bayesian estimation and classification.* Technical Report October. Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa. 172 pages. https://fenix.tecnico.ulisboa.pt/downloadFile/1126518382172510/Bayes

[6] Audun Josang. 1999. Trust-based decision making for electronic transactions. In *Proceedings of the Fourth Nordic Workshop on Secure Computer Systems (NORDSEC'99)*. 496–502.

[7] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. 2009. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09)*. Article 55, 10 pages. DOI:https://doi.org/10.4108/ICST.SIMUTOOLS2009.5674

[8] Maurice J Khabbaz, Chadi M Assi, and Wissam F Fawaz. 2012. Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges. *IEEE Communications Surveys & Tutorials* 14, 2 (jan 2012), 607–640. DOI:https://doi.org/10.1109/SURV.2011.041911.00093

[9] Na Li and Sajal K. Das. 2013. A trust-based framework for data forwarding in opportunistic networks. *Ad Hoc Networks* 11, 4 (jun 2013), 1497–1509. DOI:https://doi.org/10.1016/j.adhoc.2011.01.018

[10] Naercio Magaia, Carlos Borrego, Paulo Rogério Pereira, and Miguel Pupo Correia. 2017. PRIVO: A PRIvacy-preserVing Opportunistic routing protocol for Delay Tolerant Networks. In *IFIP Networking*. 1–9. http://dl.ifip.org/db/conf/networking/networking2017/1570333245.pdf

[11] Naercio Magaia, Paulo Rogério Pereira, and Miguel P. Correia. 2015. Security in Delay-Tolerant Mobile Cyber Physical Applications. In *Cyber-Physical Systems: From Theory to Practice*, Danda B. Rawat, Joel J. P. C. Rodrigues, and Ivan Stojmenovic (Eds.). CRC Press, Chapter 15, 373–394. DOI:https://doi.org/10.1201/b19290-22

[12] Lifei Wei, Haojin Zhu, Zhenfu Cao, and Xuemin Shen. 2013. SUCCESS: A secure user-centric and social-aware reputation based incentive scheme for DTNs. *Ad-Hoc and Sensor Wireless Networks* 19, 1-2 (2013), 95–118. DOI:https://doi.org/10.1007/978-3-642-22450-8_14