The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks

Sonja Buchegger, Jean Le Boudec

The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks

Sonja Buchegger
IBM Zurich Research Laboratory
Säumerstrasse 4, CH-8803 Rüschlikon
sob@zurich.ibm.com

Jean-Yves Le Boudec
EPFL-DSC
CH-1015 Lausanne, Switzerland
jean-yves.leboudec@epfl.ch

Abstract:

Mobile ad-hoc networks rely on the cooperation of nodes for routing and forwarding. For individual nodes there are however several advantages resulting from noncooperation, the most obvious being power saving. Nodes that act selfishly or even maliciously pose a threat to availability in mobile ad-hoc networks. Several approaches have been proposed to detect noncooperative nodes. In this paper, we investigate the effect of using rumors with respect to the detection time of misbehaved nodes as well as the robustness of the reputation system against wrong accusations. We propose a Bayesian approach for reputation representation, updates, and view integration. We also present a mechanism to detect and exclude potential lies. The simulation results indicate that by using this Bayesian approach, the reputation system is robust against slander while still benefitting from the speed-up in detection time provided by the use of rumors.

1 Introduction

Unmanaged mobile ad-hoc networks do not have any infrastructure at their disposal to ensure correct behavior, the functioning of the network relies on the cooperation of all the nodes. Cooperation is needed more but harder to enforce than in an infrastructure-based network. Nodes can arbitrarily join or leave the network. For the lack of infrastructure, detection of misbehavior and subsequent isolation of a misbehaved node has to work in a distributed fashion. Several distributed reputation systems have been proposed to gather information about the behavior of nodes and to evaluate them for future cooperation in mobile ad-hoc or peer-to-peer networks, some of these approaches are described in Section 2. The rationale for using positive, negative, or both kinds of reputation as well as for using rumors or relying exclusively on first-hand observations are discussed in Section 3. In this paper we investigate the trade-off between robustness and efficiency of using rumors and propose a mechanism to filter out slander by using Bayesian statistics and excluding seemingly implausible opinions while retaining as much as possible of the detection speed-up given by second-hand information for a distributed reputation system in Section 4. In Section 5 we present our simulation methodology and results for an evaluation of the effect of rumours on a reputation system in mobile ad-hoc networks. Section 6 offers a discussion and future directions, and Section 7 concludes the paper.

2 Reputation Systems for Mobile ad-hoc and Peer-to-Peer Networks

Watchdog and pathrater components to mitigate routing misbehavior have been proposed by Marti, Giuli, Lai and Baker [11]. They observed increased throughput in mobile ad-hoc networks by complementing DSR with a *watchdog* for detection of denied packet forwarding and a *pathrater* for trust management and routing policy rating every path used, which enable nodes to avoid malicious nodes in their routes as a reaction. The nodes rely

on their own watchdog exclusively and do not exchange reputation information with others.

CONFIDANT (see our papers [4, 3]) stands for 'Cooperation Of Nodes, Fairness In Dynamic Adhoc NeTworks' and it detects malicious nodes by means of observation or reports about several types of attacks and thus allows nodes to route around misbehaved nodes and to isolate them from the network. Nodes have a *monitor* for observations, *reputation records* for first-hand and trusted second-hand observations, *trust records* to control trust given to received warnings, and a *path manager* for nodes to adapt their behavior according to reputation. Simulations for "no forwarding" have shown that CONFIDANT can cope well even with half of the network population acting maliciously. The protocol uses also second-hand information, i.e. observations by others, which can be a vulnerability in the presence of liars.

A reputation-based trust management has been introduced by Aberer and Despotovic in the context of peer-to-peer systems [1], using the data provided by a decentralized storage method (P-Grid) as a basis for a data-mining analysis to assess the probability that an agent will cheat in the future given the information of past transactions.

CORE, a collaborative reputation mechanism proposed by Michiardi and Molva [12], also has a *watchdog* component; however it is complemented by a reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior), which are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. Reputation values are obtained by regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request. Again, nodes only rely on first-hand observations and do not exchange reputation information. A performance analysis by simulation is stated for future work.

A context-aware inference mechanism has been proposed by Paul and Westhoff [13], where accusations are related to the context of a unique route discovery process and a stipulated time period. The rating of nodes is based on accusations of others, whereby a number of accusations pointing to a single attack, the approximate knowledge of the topology, and context-aware inference are claimed to enable a node to rate an accused node without doubt. An accusation has to come from several nodes, otherwise the only node making the accusation is itself accused of misbehavior. While this mechanism discourages wrong accusations, it potentially also discourages correct accusations for fear of being the only denouncer.

3 Robustness vs. Efficiency

Relying exclusively on first-hand observations increases the detection time when compared to an approach that also uses reports from others, i.e., rumors. The more information is available, the faster the detection, however, rumors can destabilize a reputation systems when nodes make wrong observations or deliberatly lie to worsen the reputation of another node. The robustness problem caused by slander can potentially outweigh the benefit obtained by a shorter detection time.

Not considering the opinion of others is just one way to avoid destabilization by way of wrong accusations. To the same end, many reputation systems build on positive reputation only [16]. Some couple privileges to accumulated good reputation, e.g. for exchange of gaming items or auctioning [15].Positive reputation systems offer the implicit disincentive to change identifiers since reputations are built over time and having a long history of cooperation helps nodes to be chosen. Slander is not an issue in positive reputation systems, since no negative information is kept [10, 7]. Negative reputation systems offer more scalability under the assumption that misbehavior is the exception and not the norm.

We deem the combined use of both positive and negative reputation adequate for the context of mobile ad-hoc networks, as we are interested in the cooperation factor calculated as the frequency of misbehavior *relative* to the total activity of a node in a network. Moreover, the nature of the rumors should match the nature of first-hand observations or experiences. If a node keeps track of both positive and negative behavior of other nodes, the rumors considered should reflect the same kind of knowledge

in order not to introduce a bias in either direction.

As opposed to the Byzantine Generals problem, the nodes do not have to reach a consensus on which nodes misbehave. Each node can keep its own belief of the network denoted by the reputation system entries and it can choose to consider the beliefs of other nodes or to rely solely on its own observations. One node can have varying reputation records with other nodes across the network, and the subjective view of each node determines its actions. Byzantine robustness [14] in the sense of being able to tolerate a number of erratically behaving servers or in this case nodes is the goal of a reputation system in mobile ad-hoc networks. Here, the detection of misbehaved nodes by means of the reputation systems has to be followed by a response in order to render these nodes harmless.

4 A Bayesian Approach to Reputation Systems

4.1 Belief Representation

The main properties of a reputation system are the representation of reputation, how the reputation is built and updated, and for the latter, how the reputation views of others are considered and integretated. We propose to use a Bayesian approach for the representation and building of reputation as well as for subsequent decision-making depending on the reputation. Since the true probability of a node to act maliciously, say θ, is unknown, we make an estimation of θ by inference from the data obtained by direct or indirect observations. Bayes's Theorem is shown in Equation 1. It is used to calculate the probability of a random variable given an observation.

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)} \tag{1}$$

A so-called 'prior' distribution reflects the initial belief. Any up-front information can be fed into the prior to give it a head start. The prior, however, can also be chosen such that it reflects ignorance or indifference towards the initial situation. Given this prior, at each observation the information available is updated to reflect the added knowledge and to increase the precision of a belief. If the likelihood of a property is binomial, i.e., successes and failures occur independently, a good prior density is the Beta function. The Beta function is the conjugate prior for binomial likelihood and thus the posterior density is also Beta [2, 6]. The Beta function is used to reflect the prior belief. It is defined as follows.

$$f(\theta) = Beta(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{2}$$

$$\Gamma(x+1) = x\Gamma(x), \Gamma(1) = 1 \tag{3}$$

A binomial likelihood is assumed as $P(X) = \theta^n(1-\theta)^{1-n}$. The process of updating beliefs is as follows. First, choose a prior. To represent a non-informative prior and thus a uniform likelihood, we use $Beta(1,1)$. Then calculate the posterior distribution and update at each observation. We use s to represent the number of successes and f for the number of failures. Then, $Beta(\alpha,\beta)' = Beta(\alpha',\beta')$ with $\alpha' = \alpha + s$ and $\beta' = \beta + f$.

The advantage of using the Beta function is that it only needs two parameters α and β that are continuously updated as observations are made or reported. These two parameters reflect the current belief, the higher the Beta curve, the more evidence samples have been taken in. The higher the peak and the narrower, the higher the confidence in the belief that there is a certain probability around which the observations center.

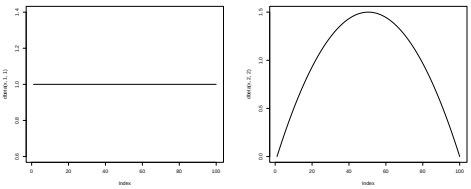
Figure 1(a) shows the non-informative flat prior of Beta(1,1), all probabilities of θ are equally likely. After some updates according to observations of successes and failures, the posterior density is depicted in Figure 1(d). The actual calculation of the density has been carried out here for illustrative purposes.

The Beta function offers moments that are simple to calculate.

$$\mathbb{E}(Beta(\alpha,\beta)) = \frac{\alpha}{\alpha+\beta} \tag{4}$$

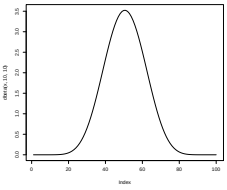
$$\sigma^2(Beta(\alpha,\beta)) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{5}$$

Applied to a reputation system, every node, say i, has a reputation component that receives as input first- or second-hand behaviour observations on

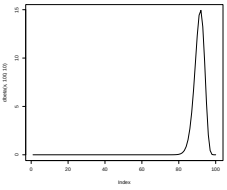


(a) Noninformative Prior Beta(1,1).

(b) Beta(2,2).



(c) Beta(10,10).



(d) Beta(100,10).

Figure 1: Density of the beta function of various observations.

other nodes, say j. It outputs decisions (misbehaving or not) for those js where node i feels able to say something. We call $R_{i,j}$ the summarized data that captures j's reputation, seen by i. $R_{i,j}$ is modified as observations are received according to the update of the Beta function as explained above.

4.2 Decision Making

In a mobile ad-hoc network, the point of keeping reputation records about other nodes of the network is to be able to make more informed decisions about whether to forward for another node, which path to choose, whether to avoid another node and delete it from the path cache, and whether to warn others about another node. Using the Bayesian approach, decisions can be made minimizing the risk for a loss, e.g., minimizing the risk of wrong classification of events, of deeming another node malicious, although it is not, or, vice versa, the risk of not recognizing a node as malicious although it actually misbehaves.

Loss can be represented as squared-error loss or 0-1 loss for classification, for instance, as depicted in equations 6 and 7.

$$L(\theta, \alpha) = (\theta - \alpha)^2 \tag{6}$$

$$L(\theta, \alpha_i) = \begin{cases} 0 & \text{if } \theta \in \Theta_i \\ 1 & \text{if } \theta \in \Theta_j, j \neq i \end{cases} \tag{7}$$

The decision-making process works as follows. First, the posterior according to all the given data is calculated. Then, for all actions the loss is calculated and weighted by its likelihood. Finally, the action δ^* with the smallest risk R (expected loss L) is chosen from $R(\theta, \delta^*) = E[L(\theta, \delta(X))]$.

In this paper, we apply the Bayesian approach to reputation updates, however, it can also serve for event classification of observations, i.e., whether they are regular protocol events or malicious attacks, as well as for trust classification to evaluate nodes according to their cooperation in the reputation system itself independent from their cooperation in the routing and forwarding according to the protocol.

4.3 Merging Models

To take advantage of rumors, i.e., to learn from observations made by others before having to learn by own experience, we need a means of incorporating the reputation beliefs into the view of an individual node.

In the particular case of misbehavior detection in mobile ad-hoc networks we want to give the most emphasis on reputation built by actually observed behavior, second-hand information should obtain less weight, since a node trusts its own observations more than a report from a random other node.

Once the weight has been determined, the entry of the node that misbehaved is changed accordingly. If the rating of a node in the table has deteriorated so much as to fall out of a tolerable range, the suspect node is declared "detected" and some action can be triggered.

There are several challenges for merging beliefs.

- False/fake belief models for deliberate deception and influence.
- Contradicting models. How to consolidate them, whom to believe, how to assign weights for significance.

- Privacy concerns. Nodes may not want to expose their opinions to others, also there is a reduction of uncertainty which might be beneficial to malicious nodes.
- With whom to share information. Who provides the most valuable information, who is trusted for their opinion, and, related to the privacy concern, whom can nodes show their beliefs without harm.

In their tutorial on Bayesian model averaging, Hoeting et al. [8] give the following methodology.

If Δ is the quantity of interest, such as an effect size, a future observable, or the utility of a course of action, then its posterior distribution given data D is:

$$pr(\Delta|D) = \sum_{k=1}^{K} pr(\Delta|M_k, D)pr(M_k|D). \quad (8)$$

This is an average of the posterior distributions under each of the models considered, weighted by their posterior model probability. $M_i, ..., M_k$ are the models considered. The posterior probability for model M_k is given by

$$pr(M_k|D) = \frac{pr(D|M_k)pr(M_k)}{\sum_{l=1}^{K} pr(D|M_l)pr(M_l)} \quad (9)$$

where

$$pr(D|M_k) = \int pr(D|\theta_k, M_k)pr(\theta_k|M_k)d\theta_k \quad (10)$$

is the integrated likelihood of model M_k, θ_k is the vector of parameters of model M_k, $pr(\theta_k|M_k)$ is the prior density of the parameters under model M_k, $pr(D|\theta_k, M_k)$ is the likelihood, and $pr(M_k)$ is the prior probability that M_k is the true model. All probabilities are implicitely conditional on $\mathcal{M}$, the set of all models considered.

In addition, Davison [5] lists the following, with z being the variable of interest, and y the data.

$$f(z|M_i, y) = \frac{\int f(z|y, \theta_i, M_i)f(y|\theta_i, M_i)\pi(\theta_i|M_i)d\theta_i}{f(y|M_i)} \quad (11)$$

Here θ_i is the parameter for model M_i, under which the prior is $\pi(\theta_i|M_i)$ and the prior probability of M_i is $Pr(M_i)$.

Berger [2] lists several methods for combining probabilistic evidence. To process different sources of information, he lists two ad-hoc systems.

Linear Opinion Pool. Assign a positive weight w_i (where $\sum_{i=1}^{m} w_i = 1$) to each information source π_i (supposedly to reflect the confidence in that information source), and then use

$$\pi(\theta) = \sum_{i=1}^{m} w_i \pi_i(\theta) \quad (12)$$

Independent Opinion Pool. When the information sources seem "independent", use, as the overall probability distributions for θ,

$$\pi(\theta) = k \left[\prod_{i=1}^{m} \pi_i(\theta) \right] \quad (13)$$

The alternative to the use of ad-hoc rules is, according to Berger, probabilistic modelling, i.e., obtaining the joint distribution of all random observables and unknown parameters of interest or, at least, determining enough to calculate the conditional (posterior) distribution of the desired θ given the observables. This is sometimes called the *super Bayesian* approach, to emphasize that it is a single decision maker (the super Bayesian) who is trying to process all the information to arrive at a distribution of θ which is consistent with probabilistic reasoning.

$$\pi(\theta_1|p) = \left[1 + \frac{(1-p)}{p}^{\alpha-\beta} \frac{\pi_2(\theta_2)}{\pi_2(\theta_1)} \right]^{-1} \quad (14)$$

In a first approach, we give the most weight to each nodes' own observations and we do not assume any a priori knowledge on the trustworthiness or expertise of a node. We thus weight second-hand reports equally among the neighbors that issue these rumors. The weight for each rumor at an exchange encounter is thus $\frac{1}{n}$, n being the number of neighbors at this particular instant. As a more advanced approach we could conceive of modeling the trust given to particular nodes and have the respective weight depend on it. A trust component qualifies the trust that node i puts on second-hand observations originated by other nodes, say k. We call $T_{i,k}$

the summarized data that captures the trust that node i places on node k . $T_{i,k}$ could first be configured by an external mechanism or be adaptive to the behavior in rumor spreading. When the reputation system receives second-hand observations (from k , about j), it would then use $T_{i,k}$ to decide how to update $R_{i,j}$ and to determine whom to send rumors to.

4.4 Robustness Against Wrong Accusations

The question is how to detect and avoid wrong accusations. Our approach is to exclude those $R_{k,j}$ for which there is a large incompatibility between $R_{k,j}$ and $R_{i,j}$ for some j. As a simple means to express that $R_{k,j}$ makes a strong case that j is bad, whereas $R_{i,j}$ does not, we exclude $R_{k,j}$ from the model merging if it deviates from $R_{i,j}$ by more than u, the deviation threshold, in either direction.

As mentioned previously, dynamic trust adaptation according to the congruency metric given by this deviation could be useful. However, we use the simpler approach of not discriminating between nodes and thus treating each rumor on a case-by-case basis and evaluate its utility solely on the grounds of how much it deviates from the belief the recipient already has. Trust management is thus rendered obsolete in this particular approach.

5 Simulation

5.1 Goals and Metrics

By means of simulation, we want to investigate the robustness and efficiency of a distributed reputation system in a mobile ad-hoc network. The key questions addressed are

- How long does it take until a misbehaved node is detected, using first-hand observations only, using also second-hand information, i.e., the first-hand observations of others, or even more indirect rumors?
- What is the effect of wrong accusations and can they be detected?
- What is the effect of varying trust models?
- How robust is the system to wrong observations?
- With whom should information be exchanged – with neighbors or remote nodes? And, what is the effect of mobility?

5.2 Methodology, Algorithms, and Parameters

5.2.1 Setup

The simulation was implemented in R [9, 17]. To simulate good and bad behavior, neighborhood, observation mistakes, movement, and trust updates, we used a grid of nodes. We investigated and compared the effect of using first-hand observations only, using also second-hand information in a network with no slander, and using also second-hand information in a network with liars but discarding too deviant opinions.

The nodes were placed in a grid, to simulate a communications range of one hop, and they observed the behavior of their neighborhood. Depending on its position in the grid, a node has up to 8 neighbors. A node can only directly observe neighbors, i.e., node i at row j and column k, denoted as i_{jk}, can observe any neighboring node n in its row $n_{j,<k+1|k-1>}$, in its column $n_{<j+1|j-1>,k}$, or diagonally one hop away $n_{<j+1|j-1>,<k+1|k-1>}$. Periodically, nodes move around. We emulate this with the following algorithm. We pick a node at random, say node $i_{j,k}$ and randomly select a new location (j', k') for it such that $j' = [j-2, j+2]$ and $k' = [k-2, k+2]$ to keep the movement reasonably local. We then repeat this with the node that we find at (j', k') and so on, until the new location is the original (j, k) and the permutation cycle is completed.

Before each move, the nodes exchanged reputation information in the form of Beta parameters with their neighbors. The liars reversed the reputation information before giving it to the neighbors. This process was iterated until all of the malicious nodes were classified as detected by all of the nodes in the network, which was the case when the expected value of the reputation, $E(theta = R_{i,j})$ exceeded

a threshold of 0.75. As a rehabilitation mechanism to mitigate the effect of slander, the nodes periodically reviewed their reputation opinions and reversed their opinion from "detected" to "regular" when the reputation was substantially better than the detection threshold.

The threshold used to determine when to exclude a suspect liar's opinion depends on the priorities. As is typical for diagnosis systems, there is a trade-off between minimizing false positives or false negatives. We chose a threshold of 50% deviation to err on the side of false positives, i.e., the mechanism excluded some true information but reliably prevented slander. This way the robustness is maintained at the price of an unused detection speed-up potential. A side-effect of the emphasis on robustness was that, given the nature of a node did not change throughout the simulation time, the rehabilitation mechanism provided for the strategy of excluding liars was never required.

5.2.2 Scenarios

- First-hand observations: $n_t(i)$ denotes the nodes that node i can observe during the time interval t, i.e. the grid neighbors. Each node j issues a sequence of bits out of $[0, 1]$ according to a distribution that depends on whether a node is good $output_{good}$ or bad $output_{bad}$. Node i sees the bits correctly with probability $correctObservation$.

  1. Place nodes in the grid.
  2. $\forall$ nodes, select $type \in \{good, bad\}$ and according probability distribution of output $output_{type}$.
  3. repeat
     (a) $\forall$ nodes output byte according to $output_{type}$.
     (b) $\forall$ nodes i, observe neighbors n correctly with probability p.
     (c) $\forall$ nodes i, n update $R_{i,n}$ using the Beta function.
  4. until $t > o$, o being the number of observations at each location.
  5. Pick node, move until cycle completed. Repeat 1–3.

  until end of simulation, then $\forall$ nodes i and j evaluate $R_{i,j}$ and compare to the $type_j$.

- Truthful second-hand observations: $S_t(i)$

  1. Iterations of the algorithm above.
  2. Periodically $\forall$ nodes i and j output $R_{i,j}$.
  3. $\forall$ nodes i and j update $R_{i,j}$ by integrating local $R_{i,j}$ and $R_{k,j}$, the neighbors' $R_{i,j}$. A variant is to use only the delta between the $R_{k,j}$ received at the last encounter and the current $R_{k,j}$, this scenario is termed 'deltas only'.

- Contaminated second-hand information: With lies and excluding lies. info only We use probability distributions $tellTruth_good$ (probability of telling the truth as a regular node) and $tellTruth_bad$ (probability of telling the truth when a node is a liar). Independent of its status as a good or bad type, nodes can be liars or truthful.

  1. Iterations of second-hand algorithm, but drawing from the probability distribution to tell a lie or the truth. When a node k lies, it swaps the α and β of its $\mathcal{B}_{k,j}(\alpha, \beta)$ represented by $R_{k,j}$ before disclosing it to the neighbors for model comparison.
  2. Compare $R_{i,j}$ with all neighbors k, weight $R_{k,j}$ by $\frac{1}{n}$ and integrate with $R_{i,j}$.
  3. For another scenario, termed 'with lies', include the contaminated information regardless. For the 'liars excluded' scenario, when comparing, only use $R_{k,j}$s according to the congruency metric, deviating less than u from $R_{i,j}$, with u being the deviation threshold and $R_{i,j}$ the accumulated reputation of j as seen by node i.

5.2.3 Parameters

Our parameters are network size, number of misbehaved nodes, number of liars, threshold for detection (default: 0.75), number of observations before moving (default: 10), information type (first-hand, second-hand, third-hand information, second-hand only considering deltas since last encounter, including lies, allowing for liars but trying to exclude

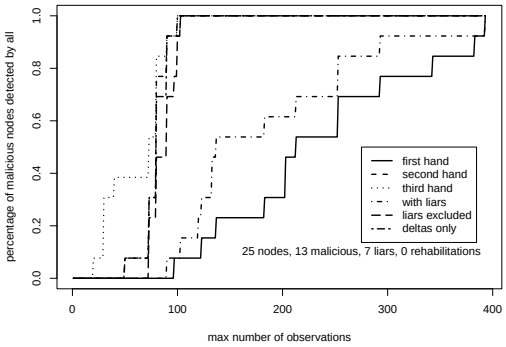


Figure 2: Percentage of malicious nodes detected vs. maximum detection time.

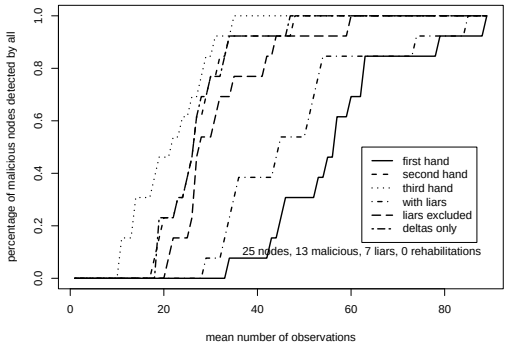


Figure 3: Percentage of malicious nodes detected vs. mean detection time.

them), partners for information exchange (default: neighbors, planned: fixed set of nodes (friends), random set of nodes), weight for model averaging (default $\frac{1}{n}$ (n being the number of neighbors), planned: weight according to adaptive trust function, full Bayesian model averaging), and detection (threshold of $E(\theta) - u$, loss function). We have probability distributions for *type* (good or bad), $output_{good}$, $output_{bad}$, $correctObservation$, $tellTruth_{good}$, $tellTruth_{bad}$, and $liar - type$.

5.3 Results

Figure 2 shows the maximum detection time, i.e., the time in the simulation when the last node detected a particular malicious node, vs. how many of the malicious nodes were detected by all at that time, Figure 3 shows the mean detection time for all nodes. These examples are representative of the results obtained by the simulation. We chose to show individual representative examples instead of mean outcomes over several runs, since the type of a node both concerning the cooperation and the lying properties are drawn from probability distributions and not explicitely specified, thus the portion of malicious nodes or liars varies.

We used true second-hand information, even third-hand information (which is not independent but reinforcing beliefs by potentially mirroring them back to the originator, we only showed it for comparison), the delta to previously received second-hand information only, contaminated second-hand information (which has the side effect of wrong accusations), and contaminated second-hand information but excluding deviating views.

Using the full set of second-hand observations or using only the difference between already received second-hand information and the current second-hand information consistently perform very similarly and very well. Exchanging the full set of observations when nodes encounter repeatedly considers information as new that has been integrated already and thus can bias the belief, whereas keeping track of the last exchanged information, albeit only two parameters per reputation, can add up to a significant storage requirement in large mobile networks.

When nodes not only exchange their own first-hand observations but hand on rumors of a deeper transitivity level, their own opinions once voiced can be reflected to them at a later time, thus reinforcing their original opinion. Although using this 'third-hand' information consistently outperforms all other strategies, it is not a valid choice since these observations are not independent.

As can be seen from Figures 4 and 5, the performance of the Bayesian approach of liar exclusion improves when the number of liars is small and approaches the performance of truthful second-hand information. In the presence of many liars, the

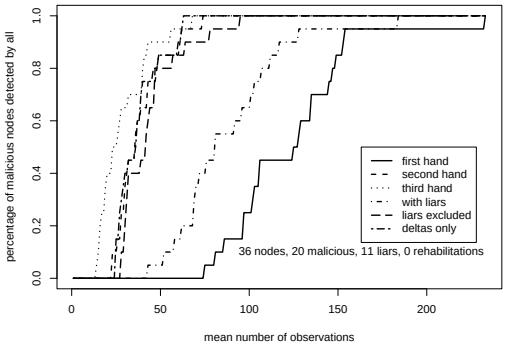


Figure 4: Percentage of malicious nodes detected vs. mean detection time.

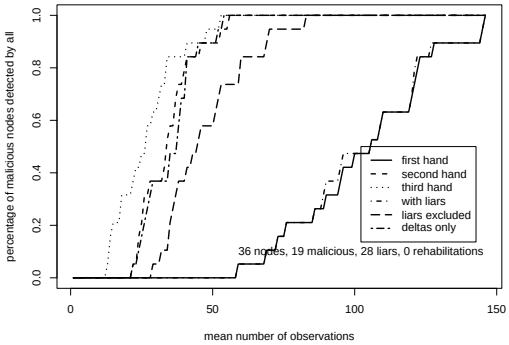


Figure 5: Percentage of malicious nodes detected vs. mean detection time.

performance degrades gradually but is still better than relying only on first-hand observations. In all the figures, the scenario 'with lies', i.e., integrating contaminated second-hand information regardless, performs better than relying on first-hand observations only, yet the price for this speed-up in detection time is that innocent nodes are also being classified as 'detected' by many nodes due to the effect of wrong accusations. This has consistently been avoided by the 'liars excluded' scenarios throughout the entire simulation.

Over the course of the simulation, it has emerged that using the 'liars excluded' Bayesian scenario significantly improves on the performance of the mean detection time when compared to the 'first hand' scenario, yet the performance gain is even higher in the worst case, namely the maximum detection time, i.e., the maximum time it takes for a malicious node to be deemed 'detected' by all the nodes of the network.

Another observation is that, as one would expect, the detection improvement given by the use of second-hand information even in the presence of liars, but given the attempt to discard the wrong accusations by means of our Bayesian approach, in fact increases with the network size. The larger the network, the higher the probability of receiving information about nodes before actually encountering them as neighbors and being able to observe their behavior.

6 Discussion and Future Work

In addition to more simulation runs to get confidence intervals (or Bayesian credible sets) and exploring a larger space of parameters, we are working on an implementation of the Bayesian approach in a more realistic mobile ad-hoc network environment and extending the existing CONFIDANT protocol by the new Bayesian approach to incorporate the insights gained to make the protocol robust against wrong accusations yet reasonably fast in detection.

With a more realistic setting and mobility model, we can then investigate the effect of changing the partners with whom to exchange rumors. Exchanging information exclusively with the current neighbors has the advantage that rumors do not have to be relayed by other nodes and thus keep the traffic local, however, the effect of receiving rumors from more remote nodes, fixed or random, could add the benefit of learning more about other nodes before having to learn from experience but potentially more useless information could get distributed than in a local exchange scenario.

An important next step is to come up with a detailed model of the adversary. For instance an adversary could try to create instabilities by lying only so much as to not be discarded at the model merging stage, yet sufficiently to worsen another nodes' reputation gradually over time. The current

system is dampening the effect of wrong accusations by forcing liars to lie more to have a fast effect, but then detection is easier. To combat the slow deliberate degradation of reputation, we intend to introduce an aging mechanism of reputation into the simulation.

7 Conclusions

Using second-hand information can significantly accelerate the detection and subsequent isolation of malicious nodes in mobile ad-hoc networks. If nodes are deceived by wrong observations or slander, the robustness of the reputation system is endangered. We found that, enabled by our Bayesian approach, by exluding opinions that deviate substantially from first-hand observation and the majority opinion of second-hand opinions gathered over time, the robustness of the reputation system remains intact even with a large number of liars in the network, while the detection speed still improves over merely using first-hand observations and, with a decreasing portion of liars, approximates the ideal case of using truthful second-hand information.

References

[1] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2001)*, 2001.

[2] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition edition, 1985.

[3] Sonja Buchegger and Jean-Yves Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403 – 410, Canary Islands, Spain, January 2002. IEEE Computer Society.

[4] Sonja Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002. IEEE.

[5] Anthony Davison. *Bayesian Models*. Chapter 11 in Manuscript, 2002.

[6] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, Inc., 1970.

[7] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 150–157, 2000.

[8] D. Hoeting, J. A.and Madigan and C.T. Raftery, A.E.and Volinsky. Bayesian model averaging: A tutorial (with discussion). *Statistical Science*, 44(4):382–417, 1999.

[9] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[10] Peter Kollock. The production of trust in online markets. *Advances in Group Processes, edited by E. J. Lawler, M. Macy, S. Thyne, and H. A. Walker*, 16, 1999.

[11] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255–265, 2000.

[12] Pietro Michiardi and Refik Molva. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia., 2002.

[13] Krishna Paul and Dirk Westhoff. Context aware inferencing to rate a selfish node in dsr based adhoc networks. In *Proceedings of the IEEE Globecom Conference*, Taipeh, Taiwan, 2002. IEEE.

[14] Radia Perlman. Network layer protocols with byzantine robustness. PhD. Thesis Massachussetts Institute of Technology, 1988.

[15] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. Working Paper for the NBER workshop on empirical studies of electronic commerce, 2001.

[16] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[17] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S-Plus. Third Edition*. Springer, 1999. ISBN 0-387-98825-4.