A Robust Reputation System for Mobile Ad-hoc Networks

EPFL IC Technical Report IC/2003/50

Sonja Buchegger
EPFL-IC-LCA
CH-1015 Lausanne, Switzerland
sonja.buchegger@epfl.ch

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

***Abstract*—Reputation systems in mobile ad-hoc networks can be tricked by the spreading of false reputation ratings, be it false accusations or false praise. Simple solutions such as exclusively relying on one's own direct observations have drawbacks, as they do not make use of all the information available. We propose a fully distributed reputation system that can cope with false disseminated information. In our approach, everyone maintains a reputation rating and a trust rating about everyone else that they care about. From time to time first-hand reputation information is exchanged with others; using a modified Bayesian approach we designed and present in this paper, only second-hand reputation information that is not incompatible with the current reputation rating is accepted. Thus, reputation ratings are slightly modified by accepted information. Trust ratings are updated based on the compatibility of second-hand reputation information with prior reputation ratings. Data is entirely distributed: someone's reputation and trust is the collection of ratings maintained by others. We enable node redemption and prevent the sudden exploitation of good reputation built over time by introducing re-evaluation and reputation fading. We present the application of our generic reputation system to the context of neighborhood watch in mobile ad-hoc networks, specifically to the CONFIDANT [3] protocol for the detection and isolation of nodes exhibiting routing or forwarding misbehavior. We evaluate the performance by simulation.**

***Index Terms*—System design, Simulations, Statistics**

I. Introduction

A. Motivation

Reputation systems have been proposed for a variety of applications, among them are the selection of good peers in a peer-to-peer network, the choice of transaction partners for online auctioning, and the detection of misbehaved nodes in mobile ad-hoc networks. There is a trade-off between efficiency in using the available information and robustness against false ratings [4]. If the ratings made by others are considered, the reputation system can be vulnerable to false accusations or false praise. However, if only one's own experience is considered, the potential of learning from experience made by others goes unused. Using only positive or only negative information reduces the vulnerability to only false praise or only false accusations.

Our goal is to make neighborhood watch systems both robust against false ratings and efficient at detecting misbehavior. We propose a mechanism that makes use of all the available information, i.e. both positive and negative, both from own and from others' experience. To make the reputation system robust we include a way of dealing with false ratings.

In the remainder of this paper we refer to the entities in the reputation system as nodes, since we apply it to nodes in a mobile ad-hoc network.

B. Solution Overview

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 ratings of others are considered and integrated. The reputation of a given node is the collection of ratings maintained by others about this node. In our approach, a node i maintains two ratings about every other node j that is cares about. The *reputation rating* represents the opinion formed by node i about node j's behavior as an actor in the base system (for example, whether node j correctly participates in the routing protocol). The *trust rating* represents node i 's opinion about how honest node j is as an actor in the reputation system (i.e. whether the reported first hand information summaries published by node j are likely to be true). We represent the ratings that node i has about node j as data structures $R_{i,j}$ for reputation and $T_{i,j}$ for trust. In addition, node i maintains a summary record of *first hand information* about node j in a data structure called $F_{i,j}$.

To take advantage of disseminated reputation information, i.e., to learn from observations made by others before

having to learn by own experience, we need a means of incorporating the reputation ratings into the views of others. We do this as follows. First, whenever node i makes a first hand observation of node j's behavior, the first hand information $F_{i,j}$ and the reputation rating $R_{i,j}$ are updated. Second, from time to time, nodes publish their first-hand information to their neighbors. Say that node i receives from k some first hand information $F_{k,j}$ about node j. If k is classified as "trustworthy" by i, or if $F_{k,j}$ is close to $R_{i,j}$ (in a sense that is made precise in Section IV-C) then $F_{k,j}$ is accepted by i and is used to slightly modify the rating $R_{i,j}$. Else, the reputation rating is not updated. In all cases, the trust rating $T_{i,k}$ is updated; if $F_{k,j}$ is close to $R_{i,j}$, the trust rating $T_{i,k}$ slightly improves, else it slightly worsens. The updates are based on a modified Bayesian approach we designed and present in this paper, and on a linear model merging heuristic.

Note that, with our method, only first hand information $F_{i,j}$ is published; the reputation and trust ratings $R_{i,j}$ and $T_{i,j}$ are never disseminated.

The ratings are used to make decisions about other nodes, which is the ultimate goal of the entire reputation system. For example, in a mobile ad-hoc network, decisions are 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. In our framework, this is done as follows. Every node uses its rating to periodically classify other nodes, according to two criteria: (1) regular/misbehaved (2) trustworthy/not trustworthy. Both classifications are performed using the Bayesian approach, based on reputation ratings for the former, trust ratings for the latter.

C. Issues in Reputation Systems for Mobile Ad-hoc Networks

Intentional vs. accidental misbehavior. Categorizations of misbehavior have been proposed, such as selfishness vs. malice. Although these types of misbehavior stem from a different motivation, they can be generalized as intentional misbehavior. However, we also deem the consideration of accidental misbehavior of high importance, and we think it is vital to protect the network against misbehaved nodes regardless the nature of their intentions. Accidental misbehavior can result in a node being unable to perform correctly due to a lack of resources or due to its particular location in the network. The enhanced version of CONFIDANT, as proposed in this paper, is indifferent to the actual cause of the misbehavior, be it intentional or accidental. When a node is classified as misbehaved it simply means that the node performs badly at routing or forwarding. No moral judgment is implied.

Should liars be punished? If we punish nodes for their seemingly inaccurate testimonials, we might end up punishing the messenger and thus discourage honest reporting of observed misbehavior. Note that we evaluate testimonial accuracy according to affinity to the belief of the requesting node along with the overall belief of the network as gathered over time. The accuracy is not measured as compared to the actual true behavior of a node, since the latter is unknown and can not be proved beyond doubt. Even if it were possible to test a node and obtain a truthful verdict on its nature, a contradicting previous testimonial could still be accurate. Thus, instead of punishing deviating views we restrict our system to merely reduce their impact on public opinion. Some node is bound to be the first witness of a node misbehaving, thus starting to deviate from public opinion. Punishing this discovery would be counterproductive, as the goal is precisely to learn about misbehaved nodes even before having had to make a bad experience in direct encounter. Therefore, in our design, we do not punish a node when it is classified as not trustworthy.

Identity. The question of identity is central to reputation systems. We require three properties of identity which we call persistent, unique, and distinct. The requirement to be persistent means that a node cannot easily change its identity. One way of achieving this is by expensive pseudonyms. This property is desirable for reputation systems to enable them to gather the behavior history of a node. An identity is unique if no other node can use it and thus impersonate another node. One way to ensure this is the use of cryptographically generated unique identifiers, as proposed by Montenegro and Castelluccia [15]. This property is needed to ensure that behavior observed was indeed that of the node observed. The requirement of distinct identities is the target of the so-called Sybil attack analyzed by Douceur [9], where nodes generate several identities for themselves to be used at the same time. This property does not so much concern the reputation system itself, since those identities that exhibit misbehavior will be excluded, while other identities stemming from the same node will remain in the network as long as they behave well. The Sybil attack can, however, influence public opinion by having its rating considered more than once. In the scenario where the mobile ad-hoc network is not completely cut off the Internet, we can make use of certification authorities. An example for such a scenario are publicly accessible wireless LANs with Internet connection. The detection and isolation of misbehaved nodes as achieved by a distributed reputation system for mobile ad-

hoc networks are still necessary, even in the presence of network operators. For the case of a pure ad-hoc network without Internet connectivity, solutions based on public keys are under investigation, see for example [5].

Redemption. Our solution enforces redemption of nodes over time, by the combination of two mechanisms: periodic re-evaluation and reputation fading. Periodic re-evaluation is implemented by the fact that node classification is performed periodically. It is thus possible for a node to redeem itself, given that nodes have each their own reputation belief which is not necessarily shared by all the others. Since their opinions can differ, a node is most probably not excluded by all other nodes and can thus partially participate in the network with the potential of showing its good behavior. Even if this is not the case and the suspect is excluded by everyone it can redeem itself by means of the second mechanism. Reputation fading is implemented by our modification to the Bayesian update of the posterior, which decays exponentially. Contrary to standard Bayesian estimation, this gives more weight to recent observations. We also periodically discount the rating in the absence of testimonials and observations.

D. Paper Contributions

In this paper we propose a reputation system that makes neighborhood watch systems in mobile ad-hoc networks, such as CONFIDANT, robust against false accusations or false praise while retaining the benefit of using second-hand information.

We introduce a mechanism based on Bayesian estimation to keep track of both positive and negative reputation and trust information, as well as the confidence in the respective ratings themselves. We achieve this by employing the Beta function for reputation and trust representation.

We modify a Bayesian model merging method to exclusively consider compatible second-hand information and even then to only slightly influence a node.

We introduce another two mechanisms, namely re-evaluation and reputation fading. The former consists of repeated Bayesian classification, and the latter modifies the standard Bayesian update by an exponential decay of the posterior. These two mechanisms allow for node redemption and at the same time prevent a node from misbehaving without hindrance by capitalizing on a good reputation built in the past.

We evaluate the performance of our proposed reputation system in its application to CONFIDANT by means of simulation using GloMoSim. We show by simulation that our method is effective at maintaining at a low level both the risk of false positives, i.e. deeming another node misbehaved although it is not, and the risk of false negatives, i.e. not recognizing a node as misbehaved although it actually misbehaves.

E. Paper Roadmap

The remainder of the paper is organized as follows. In the next section we describe what we need to know about the CONFIDANT protocol. Related work is discussed in Section III. Our Bayesian solution proposal is detailed in Section IV and its application to CONFIDANT is described in Section V. A performance evaluation follows in Section VI, and Section VII concludes the paper.

II. CONFIDANT IN A NUTSHELL

Since we apply our reputation system approach to the CONFIDANT [3] protocol, we briefly describe its main features here. The approach we use in CONFIDANT is to find the selfish and/or misbehaved nodes and to isolate them, so that misbehavior will not pay off but result in isolation and thus cannot continue. CONFIDANT is short for 'Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks' and detects misbehaved nodes by means of observation or reports about several types of attacks, thus allowing nodes to route around misbehaved nodes and to isolate them. Figure 1 shows the CONFIDANT components as extension to a routing protocol such as Dynamic Source Routing (DSR)[11].

Nodes have a *monitor* for observations, *reputation records* for first-hand and trusted second-hand observations about routing and forwarding behavior of other nodes, *trust records* to control trust given to received warnings, and a *path manager* to adapt their behavior according to reputation and to take action against misbehaved nodes. The term *reputation* is used to evaluate routing and forwarding behavior according to the network protocol, whereas the term *trust* is used to evaluate participation in the CONFIDANT meta-protocol.

The dynamic behavior of CONFIDANT is as follows. Nodes monitor their neighbors and change the reputation accordingly. If they have reason to believe that a node misbehaves, i.e. when the reputation rating is bad, they take action in terms of their own routing and forwarding. They thus route around suspected misbehaved nodes. Depending on the rating and the availability of paths to the destination, the routes containing the misbehaved node are either reranked or deleted from the path cache. Future requests by the badly rated node are ignored. In addition, once a node has detected a misbehaved node, it informs other nodes by sending an ALARM message.

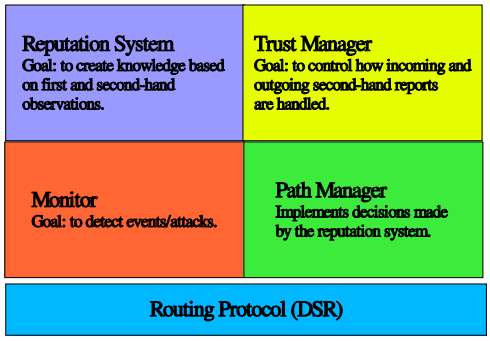


Fig. 1. CONFIDANT Components in a Node.

When a node receives such an ALARM either directly or by promiscuously listening to the network, it evaluates how trustworthy the ALARM is based on the source of the ALARM and the accumulated ALARM messages about the node in question. It can then decide whether to take action against the misbehaving node.

Simulations for "no forwarding" have shown that CONFIDANT can cope well, even if half of the network population misbehaves. Note that simply not forwarding is just one of the possible types of misbehavior in mobile ad-hoc networks. Several others, mostly concerned with routing rather that forwarding have been suggested, such as black hole routing, gray hole routing, worm hole routing. Other kinds of misbehavior aim at draining energy, such as the sleep deprivation attack. CONFIDANT is not restricted to handling any particular kind of misbehavior but can handle any attack that is observable. Even if the observation cannot precisely be attributed to an attack but is the result of another circumstance in the network such as a collision, CONFIDANT can make use of it. If it is a rare accident, it will anyhow not influence the reputation rating significantly, and if it happens more often, it means the observed node has difficulties performing its tasks.

CONFIDANT is vulnerable to the attack that a trusted node makes wrong accusations, or that a sufficient number of nodes collude to make wrong accusations. Also, CONFIDANT relies exclusively on negative ratings. This changes if we add to CONFIDANT the reputation system described in this paper. Indeed, we then have both positive and negative observations influence the rating, by managing trust dynamically or not relying on trust at all, and by limiting the weight of second-hand information altogether.

III. Related Work

A. Reputation Systems in General

False accusations are not an issue in positive reputation systems, since no negative information is kept [12], [8], however, the disseminated information could still be false praise and result in a good reputation for misbehaved nodes. Moreover, even if the disseminated information is correct, one cannot distinguish between a misbehaved node and a new node that just joined the network. Many reputation systems build on positive reputation only [20], some couple privileges to accumulated good reputation, e.g. for exchange of gaming items or auctioning [19]. Positive reputation systems are thus used for where one has a choice of transaction partners and wishes to find the best one. In mobile ad-hoc networks, the requirements are different, the focus is on the isolation of misbehaved nodes.

B. Applied to Mobile Ad-hoc or Peer-to-Peer Networks

In the following we describe and discuss several misbehavior detection and reputation systems that are fully distributed and hence potential solutions for mobile ad-hoc networks or peer-to-peer networks. For each of these we describe the strategy used to detect misbehaved nodes and to cope with false accusations and relate it to ours.

The following protocols either rely only on first-hand information or on positive second-hand information. Since in this paper we evaluate the use of disseminated information, we provide a quantitative reason, namely the speed-up of detection time, why they could potentially benefit from our Bayesian approach while still being robust against false accusations.

Watchdog and path rater components to mitigate routing misbehavior have been proposed by Marti, Giuli, Lai and Baker [13]. They observed increased throughput in mobile ad-hoc networks by complementing DSR with a *watchdog* for detection of denied packet forwarding and a *path rater* 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. They thus chose the approach of not using information dissemination, trading off the robustness against longer detection delay.

CORE, a collaborative reputation mechanism proposed by Michiardi and Molva [14], 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. Nodes only exchange positive reputation information, thus making the same trade-off between robustness against lies and detection speed as the watchdog and path rater scheme, but in addition, false praise can make malicious nodes harder to detect. A performance analysis by simulation is stated for future work.

The protocols discussed next already use negative second-hand information and cope with false accusations by requiring the disseminated information to come from several sources. Our approach could be beneficial for them in the case of collusion of several liars. As opposed to the protocols previously discussed in this section, the benefit is not straightforward to quantify and thus outside of the scope of this paper.

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. The disseminated information is exclusively negative, in the form of complaints that are then redundantly stored at different agents. When agents want to assess the trustworthiness of other agents, they query several agents for complaints about the agent in question. To assess the trustworthiness of the agents responding to the query and thus to avoid relying on lies, a complaint query about that agent can be made. To avoid the exploration of the whole network, the trustworthiness of the responders is said to be given when a sufficient number of replicas returns the same result. An assumption is that the underlying communication network is sound in that the complaints do not have to be routed through malicious nodes, so the approach is not readily applicable to mobile ad-hoc networks.

A context-aware inference mechanism has been proposed by Paul and Westhoff [17], 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 false accusations, it potentially also discourages correct accusations for fear of being the only denouncer, resulting in reduced information dissemination.

The protocols discussed next use second-hand information.

A formal model for trust in dynamic networks based on intervals and a policy language has been proposed by Carbone, Nielsen, and Sassone [6]. They express both trust and the uncertainty of it as trust ordering and information ordering, respectively. They consider the delegation of trust to other principals. In their model, only positive information influences trust, such that the information ordering and the trust ordering can differ. In our system, both positive and negative information influence the trust and the certainty, since we prefer p positive observations that come out of n total observations to p out of N when $n < N$. Evaluation of the trust model and the design of an operational model are stated for future work.

Collaboration enforcement for peer-to-peer networks have been proposed by Moreton and Twigg [16]. They allow for selective trust transitivity and distinguish between trust as participator and trust as recommender. They define three operators, namely discounting, consensus, and difference, to compute trust values. Since they use recommenders, trust in participators, trust in recommenders, and meta-recommenders, the trust becomes recursive and they thus look for fixed-point solutions to the resulting trust equations. The performance has not been evaluated.

As opposed to the **Byzantine Generals problem**, the nodes in a misbehavior detection and reputation system for mobile ad-hoc networks do not have to reach a consensus on which nodes misbehave. Each node can keep its own rating of the network denoted by the reputation system entries and it can choose to consider the ratings 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 [18] 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 malicious nodes by means of the reputation systems has to be followed by a response in order to render these nodes harmless.

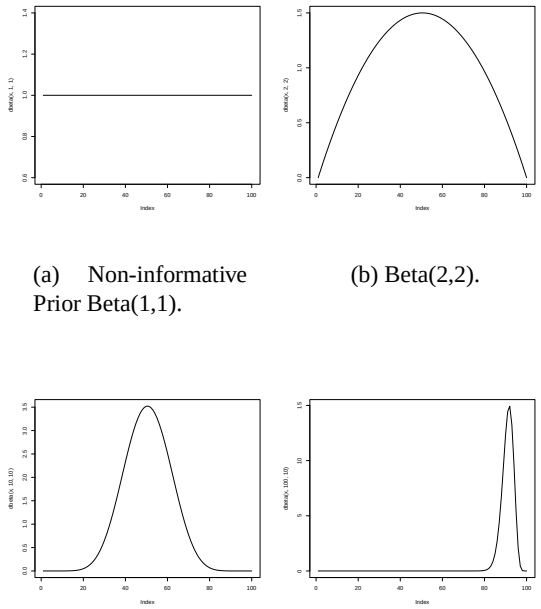
IV. Solution Proposal: A Bayesian Approach to Reputation Systems

A. A Bayesian Framework

Node i models the behavior of node j as an actor in the base system as follows. Node i thinks that there is a parameter θ such that node j misbehaves with probability θ, and that the outcome is drawn independently from observation to observation (Node i thinks that there is a

different parameter θ for every different node j, and every node i may believe in different parameters θ; thus θ should be indexed by i and j, but for brevity, we omit the indices here). The parameters θ are unknown, and node i models this uncertainty by assuming that θ itself is drawn according to a distribution (the "prior") that is updated as new observations become available. This is the standard Bayesian framework. We use for the prior the distribution Beta(α, β), as is commonly done [2], [7].

The standard Bayesian procedure is as follows. Initially, the prior is Beta$(1, 1)$, the uniform distribution on $[0, 1]$; this represents absence of information about which θ will be drawn. Then, when a new observation is made, say with s observed misbehaviors and f observed correct behaviors, the prior is updated according to $\alpha := \alpha + s$ and $\beta := \beta + f$. If θ, the true unknown value, is constant, then after a large number n of observations, $\alpha \sim n\theta$ (in expectation), $\beta \sim n(1 - \theta)$ and Beta(α, β) becomes close to a Dirac at θ, as expected. The advantage of using the Beta function is that it only needs two parameters that are continuously updated as observations are made or reported. See Figure 2 (the actual calculation of the density has been carried out here for illustrative purpose only).



(a) Non-informative Prior Beta(1,1).

(b) Beta(2,2).

(c) Beta(10,10).

(d) Beta(100,10).

Fig. 2. Density of the Beta Function.

We use a modification of the standard Bayesian method, one for reputation, and one for trust, as described next.

B. Modified Bayesian Approach for First-Hand Information

The first-hand information record $F_{i,j}$ mentioned in the introduction has the form (α, β). It represents the parameters of the Beta distribution assumed by node i in its Bayesian view of node j's behavior as an actor in the base system. Initially, it is set to $(1, 1)$.

The standard Bayesian method gives the same weight to each observation, regardless of its time of occurrence. We want to give less weight to evidence received in the past to allow for reputation fading. We therefore developed a modified Bayesian update approach by introducing a moving weighted average as follows. Assume i makes one individual observation about j; let $s = 1$ if this observation is qualified as misbehavior (by a system such as CONFIDANT), and $s = 0$ otherwise. The update is

$$\alpha := u\alpha + s \quad (1)$$

$$\beta := u\beta + (1 - s) \quad (2)$$

The weight u is a discount factor for past experiences, which serves as the fading mechanism.

We now analyze how to find a good value of u. Call $s_1, ..., s_n$ the sequence of observations. We can easily derive from Equation (1) that the value of α after n first hand observations is

$$\alpha_n = s_n + us_{n-1} + ... + u^{n-1}s_1 + u^n \quad (3)$$

Assume (temporarily) that θ would be constant. For large n we would have

$$\mathbb{E}(\alpha_n) \approx \frac{\theta}{1 - u} \quad (4)$$

$$\mathbb{E}(\beta_n) \approx \frac{1 - \theta}{1 - u} \quad (5)$$

Assume in addition that $m = \frac{1}{1-u}$ is an integer. Thus the standard Bayesian approach after m observations would result in the same posterior as ours after infinitely many observations. Thus, as a rule of thumb, we should select u as

$$u = 1 - \frac{1}{m} \quad (6)$$

where m is the order of magnitude of the number of observations over which we believe it makes sense to assume stationary behavior.

In addition, during inactivity periods, we periodically decay the values of α, β as follows. Whenever the inactivity time expires, we let $\alpha := u\alpha$ and $\beta := u\beta$. This is to allow for redemption even in the absence of observations.

C. Reputation Rating and Model Merging

The reputation rating $R_{i,j}$ is also defined by two numbers, say (α', β'). Initially, it is set to $(1, 1)$. It is updated on two types of events: (1) when first-hand observation is updated (2) when a reputation rating published by some other node is copied.

In the former case, the update is the same as for the first-hand information. More precisely: let $s \in \{0, 1\}$ be the observation:

$$\alpha' := u\alpha' + s \quad (7)$$

$$\beta' := u\beta' + (1 - s) \quad (8)$$

If the update to the first-hand information is due to inactivity, the formula is $\alpha' := u\alpha'$, $\beta' := u\beta'$.

In the latter case, we use linear pool model merging [2], as follows. Assume node i receives the reported first-hand information $F_{k,j}$ from node k. The question is how to detect and avoid false reports. Our approach is for a node i to take into account trust and compatibility. If $T_{i,k}$ is such that i considers k trustworthy according to Equation (14) (defined later), $F_{k,j}$ is considered by node i who modifies $R_{i,j}$ according to

$$R_{i,j} := R_{i,j} + wF_{k,j} \quad (9)$$

Here, w is a small positive constant [2]. This is performed for all j contained in the report.

Otherwise, i considers k not trustworthy, and, for every node j in the report, uses the results of the *deviation test*, as follows. We denote with $\mathbb{E}(\text{Beta}(\alpha, \beta))$ the expectation of the distribution $\text{Beta}(\alpha, \beta)$. Let $F_{k,j} = (\alpha_F, \beta_F)$ and $R_{i,j} = (\alpha, \beta)$ The deviation test is

$$|\mathbb{E}(\text{Beta}(\alpha_F, \beta_F)) - \mathbb{E}(\text{Beta}(\alpha, \beta))| \geq d \quad (10)$$

where d is a positive constant (deviation threshold). If the deviation test is positive, the first hand information $F_{k,j}$ is considered incompatible and is not used. Else $F_{k,j}$ is incorporated using Equation (9) as previously.

D. Trust Ratings

Trust rating uses a similar Bayesian approach. Node i thinks that there is a parameter ϕ such that node j gives false reports with probability ϕ, so it uses for ϕ the prior $\text{Beta}(\gamma, \delta)$. The trust rating $T_{i,j}$ is equal to (γ, δ).

Initially, $(\gamma, \delta) = (1, 1)$. Then an update is performed whenever node i receives a reported by some node k on first-hand information about node j. Let $s = 1$ if the deviation test in Equation (10) succeeds, and $s = 0$ otherwise. The trust rating $T_{i,k} = (\gamma, \delta)$ is updated by

$$\gamma := v\gamma + s \quad (11)$$

$$\delta := v\delta + (1 - s) \quad (12)$$

Here v is the discount factor for trust, similar to u. There is a similar update in periods of inactivity as for first hand information.

Note that the deviation test is always performed, whether k is considered trustworthy by i or not. In the former case, it is used only to update $T_{i,k}$; in the latter case, it is used to update $T_{i,k}$ and decide whether to update $R_{i,j}$.

E. Classification

The decision-making process works as follows. First, the posterior according to all the given data is calculated. This is done by node i by updating $R_{i,j} = (\alpha', \beta')$ and $T_{i,j} = (\gamma, \delta)$ as explained above. Then node i chooses the decision with minimal loss.

We use squared-error loss for the deviation from the true θ and ϕ; this amounts to considering $\mathbb{E}(\text{Beta}(\alpha', \beta'))$ for θ and $\mathbb{E}(\text{Beta}(\gamma, \delta))$ for ϕ. More precisely:

Node i classifies the behavior of node j as

$$\begin{cases} \text{regular} & \text{if } \mathbb{E}(\text{Beta}(\alpha', \beta')) < r \\ \text{misbehaved} & \text{if } \mathbb{E}(\text{Beta}(\alpha', \beta')) \geq r \end{cases} \quad (13)$$

and the trustworthiness of node j as

$$\begin{cases} \text{trustworthy} & \text{if } \mathbb{E}(\text{Beta}(\gamma, \delta)) < t \\ \text{not trustworthy} & \text{if } \mathbb{E}(\text{Beta}(\gamma, \delta)) \geq t \end{cases} \quad (14)$$

The thresholds r and t are an expression of tolerance. If node i tolerates a node j that misbehaves not more than half of the time, it should set r to 0.5. In analogy, if i trusts a node if its ratings deviate no more than in 25% of the cases, it sets its t to 0.75.

V. Application to CONFIDANT

We plugged our system into CONFIDANT as the Reputation System and Trust Manager components. We discuss the protocol behavior in detail in the following description.

A. Monitoring

Nodes constantly monitor their neighborhood by promiscuously listening to transmissions in order to detect misbehavior. Specifically to detect non forwarding, they make use of passive acknowledgement to verify if the next-hop node forwards a packet. Every observation made by node i about node j constitutes an s as defined in Section IV-B. The observation made is taken as indication either for regular behavior of a node, i.e. for forwarding this is the overheard attempt of forwarding a packet,

or as indication for misbehavior, the timeout of the passive acknowledgement timer without overheard forwarding. Each observation thus made is given to the reputation system.

The monitor component not only overhears the routing and forwarding behavior of the neighborhood but also listens to published reports. These are handed over to the trust manager for evaluation and subsequent model integration by the reputation system.

B. Managing Reputation

Maintaining reputation records. Node i keeps reputation records about every other node j that it cares about. The records $F_{i,j}$ contain the first-hand observations, $R_{i,j}$ records contain the reputation rating about j including accepted reports.

Updating reputation. Every time a node i receives evidence about node j from the monitor, it updates $F_{i,j}$ and $R_{i,j}$ according to Equation (1).

Classifying nodes. When receiving new evidence or when making a decision about any node j, node i checks its rating of j, $R_{i,j}$, to classify a node as misbehaved or regular as in Equation (13).

Re-evaluating for redemption. Every time new evidence about node j is considered at node i, it re-evaluates its classification according to Equation (13). In addition, for the case that a node has been excluded or simply not active to enable observation, $R_{i,j}$ is slightly faded as in Equation (1).

C. Managing Trust

In the trust part of the protocol, we determine how second-hand information is considered. In Section IV we explained how models are selected and merged and how trust is updated. Here we describe where the models come from.

Exchanging information. From time to time, nodes publish their first-hand observations of other nodes as reports. This is done by each node, say i, by sending its set of $F_{i,j}$ for all nodes j that it has observed to its next-hop neighbors by broadcasting with a TTL set to 1.

Integrating second-hand information. Whenever the monitor overhears a publication of reputation by another node, it is evaluated for trust according to Equation (14) and compatibility using the deviation test of Equation (10).Only the second-hand information that passed this evaluation is integrated according to Equation (9). The result of the deviation test is also used to update the trust of i in j, as captured in $T_{i,j}$, according to Equation (11). .

D. Managing Paths

The reputation ratings provided by the reputation system are used for classifying nodes as regular or misbehaved using Equation (14). The latter triggers the following actions.

Checking the path cache. Paths that have been obtained by previous Route Requests are stored in the path cache. Once a node has been classified as misbehaved, the path cache is searched whether it contains paths including that node.

Deleting paths. Paths that contain a misbehaved node are considered contaminated and are deleted.

Rerouting. If after the deletion of contaminated paths there is no path left to a destination for which a node has packets to send, it triggers a Route Request according to DSR.

Re-ranking paths. If the result of a Route Request is only a set of contaminated paths, the node performs a path re-ranking and prefers the least contaminated path. This is an optimistic attempt to get its packets through, since there is a non-zero probability of the path being salvaged. And the alternative of not sending at all provides certain zero throughput.

Ignoring requests. A Route Request received by the monitor does not trigger a Route Reply by the path manager if the source of the Route Request is classified as misbehaved. As opposed to the previous actions which aim at maximizing its own throughput, this action aims at retribution and isolation of the misbehaved node.

VI. Performance Evaluation

We evaluate the performance of our reputation system, as applied to the CONFIDANT protocol by simulation using the GloMoSim simulator.

A. Goals and Metrics of the Simulation

In previous work, we have shown the performance of CONFIDANT in increased throughput, decreased number of packets dropped intentionally, and its overhead in terms of control messages. In this paper we focus on the robustness performance of our proposed modified Bayesian approach as applied to CONFIDANT. Specifically we are interested in its performance according to the following metrics.

1) Detection time of misbehaved nodes. We measure this as the time taken for all misbehaved nodes to be classified as detected by all regular nodes.
2) Robustness against false accusations (false positives). We consider a false positive to be the classification of a regular node as misbehaved by one regular node.

3) Robustness against false praise (false negatives). Here we have to distinguish between two cases. The first case is a misbehaved node that has not been classified as such by a regular node due to lack of encounter or second-hand information. The second case is that a misbehaved node has been classified as regular despite the information available, hence a misclassification in the steady state of the protocol. We call the latter case a false negative.
4) Overhead in terms of control messages, computation, and storage.

B. Simulation Parameters and Factors

Parameter	Level
Area	1000 m × 1000 m
Speed	uniformly distributed between 10 and 20 m/s
Radio Range	250 m
Placement	uniform
Movement	random waypoint model
MAC	802.11
Sending capacity	2 Mbps
Application	CBR
Packet size	64 B
Passive ack period	100 ms
Simulation time	900 s
Fading u, v	0.999
Threshold r	0.75
Threshold d	0.5
Publication timer	10 s
Re-evaluation timer	10 s
Fading timer	10 s

TABLE I
FIXED PARAMETERS

The fixed parameters for the simulation are listed in Table I. The radio range, sending capacity and MAC have been chosen to represent an off-the-shelf device, the chosen area approximately represents the center of a town. The simulation time is chosen to be long enough to potentially roam the whole area. The mobility model chosen is the *Random Waypoint Model*, but with a range of 10 to 20 m/s to avoid the low-mobility non-stationary effects of the model [21]. The placement has been chosen to start with a good network connectivity over the whole area. Finally, CBR has been chosen for traffic (we refer to it as applications) to avoid protocol particularities of more complicated protocols such as TCP.

In order to find out which factors actually have an effect on the performance metrics and to reduce the number of experiments, a $2^k r$ factorial design according to Jain [10] is being performed, with k (the number of factors) being set to 6 , r (the number of repetitions of the experiment) set to 10, resulting in experiments or 640 simulation runs, respectively. Table II shows the factors and the two extreme levels that were chosen for the experiments.

Factor	Level 1	Level 2
Number of nodes	10	50
Pause time	0 s	600 s
Percentage of misbehaved nodes	10%	50%
Weight w	0	0.1
Percentage of untrustworthy nodes	10%	50%
Threshold t	0.25	1

TABLE II
LEVELS FOR FACTORIAL DESIGN

The choice for the number of nodes was made with the intention to show both a very small network that still allows for multiple paths and reasonable network connectivity given the area and a larger network to get insights on scalability. The pause times were chosen to reflect a very mobile network as well as a very moderately mobile one given that the duration of the simulation is 900 s.

To simulate the effect of the trust component and its absence, we set the threshold t to 0.25 and to 1, the latter meaning a node trusts anyone.

Similarly, to show the effect of presence or absence of the reputation system, we set w, the weight for second-hand observations, to 0.1 and 0, the latter meaning that nodes do not consider second-hand information at all.

C. Results

Figure 3 shows the mean detection time, i.e., the time in the simulation when the last node detected a particular misbehaved node, vs. which fraction of the malicious nodes were detected by all at that time, Figure 4 shows the maximum detection time for all nodes. We compare the use of second-hand reports to relying exclusively on first-hand observation by means of taking extreme values for w, the weight given to second-hand reports at the model-merging stage. Although the percentage of untrustworthy nodes that reversed their reputation ratings when publishing is as high as 50% in this particular set of experiment runs, it nevertheless pays off to consider compati-

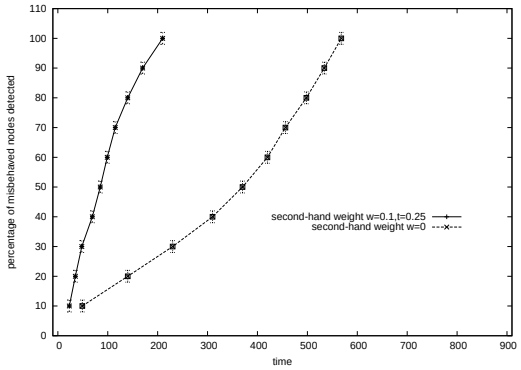


Fig. 3. Mean Detection Time of All Misbehaved Nodes.

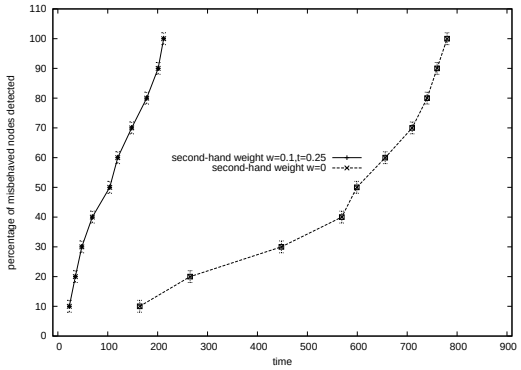


Fig. 4. Max Detection Time of All Misbehaved Nodes.

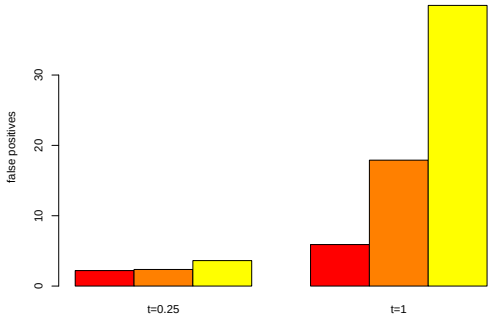


Fig. 5. False Positives with Increased Untrustworthy Population.

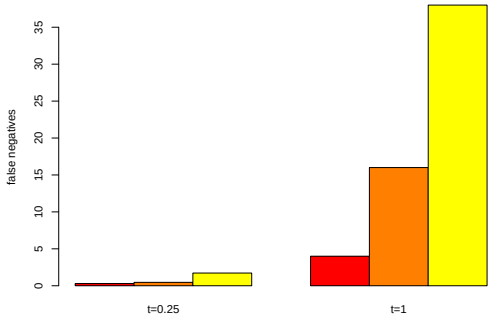


Fig. 6. False Negatives with Increased Untrustworthy Population.

ble second-hand reports. The time for detection of misbehaved nodes is significantly shorter.

The potential drawback of using a $w > 0$ in terms of false positives or false negatives is shown in Figures 5 and 6, respectively. Here, the results depend strongly on how trust is handled. We varied the trust threshold t to show the effect of the presence or absence of trust management. Both the false positives and negatives are limited by the having the effect of the deviation test come into play as the trust threshold is set to a small value that expresses trust only when the source of the report has been evaluated as trustworthy in the past. The smaller the trust threshold, the smaller the probability of a record to be accepted for model merging, yet even then it improves the decision making of a node.

The numbers of false positives and negatives do not vary much with the increase of the proportion of untrustworthy nodes, here from 0.1 to 0.5 and 0.9, if the trust threshold t is significantly smaller than 1.

The ratio of false positives and negatives to correct positives and negatives, respectively, depend on the simulation time and the frequency of re-evaluation in our simulation, because the misbehavior is constant over time. We are currently investigating several adversary types that have irregular misbehavior.

Over the course of the simulation, it has emerged that using second-hand ratings significantly improves on the performance of the mean detection time when compared to the using only first-hand observations, 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 misbehaved node to be classified as such by all the nodes of the network.

The overhead in terms of additional messages is one local message per node to its one-hop neighbors per publication interval, in our simulation, once per 10 seconds. These publications do not get forwarded. Storage overhead are the three ratings, $R_{i,j}$, $T_{i,j}$, and $F_{i,j}$, that each node i stores about each node j that it cares about. The ratings consist of two parameters each.

VII. Conclusions

In this paper, we proposed a robust reputation system for misbehavior detection in mobile ad-hoc networks. Our solution is based on a modified Bayesian estimation approach which we designed. In our approach, everyone

maintains a reputation rating and a trust rating about everyone else who is of interest. The approach is fully distributed and no agreement is necessary. However, to speed up the detection of misbehaved nodes, it is advantageous to, cautiously, make use also of reputation records from others in addition to first-hand observations. These records are only considered in the case when they come from a source that has consistently been trustworthy or when they pass the deviation test which evaluates compatibility with one's own reputation ratings. Even after passing the test, they only slightly modify the reputation rating of a node. The results of the deviation test are additionally used to update the trust rating. We allow for redemption and prevent capitalizing excessively on past behavior by two mechanisms, namely re-evaluation and fading. We presented a concrete application of our proposed reputation system to a neighborhood watch system for mobile ad-hoc networks, specifically the CONFIDANT protocol. We evaluated the performance by simulation and showed that our method is coping well with false second-hand reports, as it keeps the number of false positives and false negatives low. The simulation also showed that the detection of misbehaved nodes accelerates significantly with the use of selected second-hand information. Further performance evaluation by simulation and the investigation of additional elaborate adversary models, both for misbehavior and for trustworthiness, are under way.

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. 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.

[4] Sonja Buchegger and Jean-Yves Le Boudec. The effect of rumor spreading in reputation systems in mobile ad-hoc networks. Wiopt'03, Sofi a-Antipolis, March, 2003.

[5] S. Capkun, L. Buttyan, and J. P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, page 17, 2003.

[6] Marco Carbone, Mogens Nielsen, and Vladimiro Sassone. A formal model for trust in dynamic networks. BRICS Report RS-03-4, 2003.

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

[8] 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.

[9] John R. Douceur. The sybil attack. In Proc. of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002.

[10] Raj Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, New York, 1989 edition, 1991. All you need to know about performance analysis.

[11] Dave B. Johnson and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, Mobile Ad Hoc Network (MANET) Working Group, IETF, October 1999.

[12] 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.

[13] 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.

[14] Pietro Michiardi and Refi k 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.

[15] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifi able(sucv) identifi ers and addresses. NDSS'02, February 2002., 2002.

[16] Tim Moreton and Andrew Twigg. Enforcing collaboration in peer-to-peer routing services, 2003.

[17] Krishna Paul and Dirk Westhoff. Context aware inferencing to rate a selfi sh node in dsr based ad-hoc networks. In *Proceedings of the IEEE Globecom Conference*, Taipeh, Taiwan, 2002. IEEE.

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

[19] 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.

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

[21] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. Infocom 2003, 2003.