Enabling Fast Bootstrap of Reputation in P2P Mobile Networks

Roberto G. Cascella
University of Trento
Dipartimento di Ingegneria e Scienza dell'Informazione
via Sommarive 14, I-38100 Povo (TN), Italy
cascella@disi.unitn.it

Abstract

The easy deployment of P2P self-organized systems has contributed to their wide diffusion and to the definition of a new communication paradigm. Mobile communities can now spontaneously emerge to enable users to become both consumer and service providers. However, the presence of selfish and malicious nodes can thwart the sustainability of these systems as nodes try to exploit services without contributing resources. In these P2P systems reputation management schemes can promote collaboration, but they are mostly ineffective in communities that last for short time.

In this paper we propose a token based mechanism that extends existing reputation management schemes to support mobility. It reduces the problem of bootstrapping the reputation values and provides incentives for nodes to properly behave. Simulation results show that the token based extension enables the correlation of transactions in different contexts efficiently.

1. Introduction

The integration of peer-to-peer (P2P) technology with mobile applications brings new interesting opportunities both for mobile consumers and wireless providers. Thus, the attention of researchers focuses on the design of new service platforms for the integration of the two technologies [1]. This new communication paradigm leverages autonomous systems, such as P2P self-organizing networks, to change the role of the user, who is at the same time content consumer and producer. In a mobile and autonomous system multiple community of interests can thus spontaneously mushroom based on the common interests of users or their current positions. These virtual groups consist of nodes who dynamically can leave/join by simply changing location.

In such a scenario, the survivability of these self-organizing systems relies on the willingness of mobile entities to contribute in terms of bandwidth, storage, battery and services. But, the human nature is not prone to follow instructions toward the social welfare and nodes tend to be selfish, i.e., they do not share resources or, in the worst case, to be malicious, i.e., they misbehave just for the sake of disrupting the network functionality. The impact on the system performance has different effects, but the need to reduce the risks of possible *attacks* is the same.

An effective countermeasure is reputation management schemes which give incentives for collaboration [2] and reduce the risk of transacting with malicious users [3], [4], [5]. However, the applicability of these reputation management schemes can be inappropriate for communities of interest that live for short time as nodes cannot account for past transactions. In particular, the nodes' reputation value is uncertain and the cost of the initialization of the reputation management scheme can be too high compared to the benefits [6].

In this paper, we define a new mechanism, suitable for ad hoc P2P virtual communities, which is based on the use of a personal token to reduce the bootstrap problem of reputation management schemes. This token stores reports which reflect the past behaviour of the node and each report is digitally signed by the clusterheads of previous communities, who act on behalf of the community for providing a consistent view of the activities of the nodes.

Nodes are not anymore considered as new entrants in a community. Therefore, nodes can leverage the reputation value gathered in other communities to start benefiting from their past cooperative transactions and, at the same time, the new community has a preliminary estimation of how the node will behave.

The rest of this paper is organized as follows. Sec 2 discusses the related works. Sec. 3 presents the system objective and the adversarial model. Sec 4 and Sec. 5 detail the proposed solution and its implementation respectively. Sec. 6 evaluates the approach and Sec. 7 discusses the security of our solution. Finally, Sec. 8 concludes the paper.

2. Related works

The use of reputation management schemes is conditional to three properties [7]: 1) nodes must stay a long time in the system in order to account for future interactions otherwise they only look for the immediate outcome of the transaction if the time that nodes remain online is short; 2) nodes should

report transactions and distribute feedbacks; 3) the reputation value should be useful for the community.

In distributed and self-organized systems mobility is an issue for the correct establishment of reputation management schemes as user relocation results in a high churn rate. Indeed, nodes join the communities for a short period and approaches similar to tit-for-tat, as in BitTorrent, are ineffective. If we do not count past transactions, nodes are considered *strangers* when they join a new community and they can hardly start to benefit from their participation. In fact, other nodes might not initiate transaction with them because they are unknown and might be reputed to be malicious.

In general, reputation management schemes for P2P systems rely on designated agents to aggregate and store reputation values [3], but they still require an initial *training* period to predict correctly the nodes' behavior. [8] propose to organize nodes in a hierarchical structure to make an effective use of the designated agents for storing reputation values. For instance in a mobile setting, nodes that move often can join the lower layers while nodes in higher layers can keep track of reputation values. The feasibility of this solution is limited by the delay in transferring information between nodes, that increases if ad hoc networks are temporally disconnected.

[9] proposes to solve the bootstrap problem of the reputation value by leveraging the presence of an *ambassador*, trusted by the community clusterhead, in other regions. This ambassador verifies the *visa* issued by the clusterhead to a node traveling from the home community to this new region. This approach requires the presence of ambassadors of every region in every other region, which cannot be guaranteed in an autonomous self-organized system. Moreover, it relies on the willingness of the ambassador to guarantee for the nodes and on the trustworthiness of the ambassadors themselves. Our solution differs because we do not rely on *home* nodes in the *visitor* community but we leverage distributed signatures schemes for clusterheads who act on behalf of the community, as we discuss in Sec. 7.

[10] proposes another solution which consists of newcomers who query the system to ask its members to lend part of their reputation. This mechanism enables new nodes to participate actively in the system after joining the community. Although this solution is appealing, it falls short to address mobile nodes that move across communities. Nodes are mainly *strangers* and the interactions can be too short in time to establish a *trust* relationship among the nodes.

3. System objective and adversarial model

We target a self-organized system, without any central authority, composed by entities that dynamically change positions in an open environment. The movement can be driven by a task or can be random in the area. We envision the spontaneous formation of ad hoc communities in different locations to enable content and service exchange. We suppose that users, while on the move, can join the virtual community established in a specific cluster.

Our purpose is to facilitate the joining process of the mobile nodes in new areas. The objective is to define a mechanism that enables the application of reputation management schemes in ephemeral communities so that nodes can be rewarded for their good behaviour in the past. This results in incentives for cooperation in all communities.

We consider a system populated by malicious nodes that can inject false content in the system or they can misreport information with the intent to subvert the system. The goal of this scheme is to thwart malicious behaviour and to reduce the risk of impersonation, whitewashing, bad mouthing and repudiation attacks [11]. Herein we specifically do not deal with collusion attacks and DoS attacks, in the sense that nodes can send multiple requests or multiple reports to overload the serving capabilities.

4. Our scheme: the token based approach

In this paper, we propose a token-based mechanism that extends reputation management systems in such a way that nodes can *maintain* the history of their past transactions. This token correlates reputation values earned in different communities and it gives a first view of the node's willingness to cooperate when joining a new one. Each entry of the token is digitally signed by a clusterhead on behalf of the community to bind the report with the behaviour of the node within the community itself. This solution moves the burden of storing personal information to the nodes as it is their interest to trace their reputation.

The use of the token has a twofold meaning: 1) it eliminates the problem of the bootstrap of reputation in a new community and 2) it allows nodes to exploit their reputation values to benefit services immediately. From the community perspective, it is also important to admit new nodes, which might bring new content and resources, and have a preliminary estimation of how these nodes will behave. As a consequence, nodes will be rewarded if they prove to be trustworthy while those malicious will be punished because they limit the scope of the community.

4.1. The reputation management scheme

We refer to ROCQ [3] as reputation management scheme for describing the token based approach and for evaluation purposes of our mechanism, which is general and can work with other distributed reputation schemes.

In ROCQ, each node is associated with an identifier which is globally unique and it is used to identify the node for reputation management purposes. Multiple designated agents are selected for each node and they collect and store

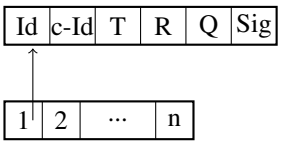


Figure 1. The Trust-Token

the reputation value of the node itself. Before interacting with a node j, a peer x retrieves the node reputation values and, then, it decides if interacting with j by computing the trustworthiness based on the reputation values and the local opinion, if they have transacted in the past.

$$R_{xj}^{avg} = \frac{\sum_d R_{dj} \cdot C_{xd} \cdot Q_{dj}}{\sum_d C_{xd} \cdot Qdj} \quad (1)$$

Eq. (1) shows the reputation value, which is computed by weighing the values R_{dj} retrieved from the designated agents d with the credibility C_{xd} that this nodes has on the reporting capabilities of these agents and the confidence Q_{dj} the agents have in their reporting values. The credibility is a measure used to detect agents that report incorrect values while the quality is used by agents to indicate how much their opinions should count.

After each transaction, nodes report their opinions on the transaction to the designated score agents, which aggregate reports to compute the reputation value. The details of the ROCQ scheme can be found in [3].

4.2. Mobility support: the token

We assume the presence of clusterheads which are the nodes that initiated the community and are mainly stable inside the cluster area. This assumption is consistent with human mobility because nodes tend to persist in a location due to a specific interest or task, such as work or leisure activities [12]. In the remaining of this paper we use the terms cluster and community interchangeably.

Clusterheads are considered trustworthy in providing the view of the community with respect to the past behaviour of a node moving to other communities since they do not have interest in loosing the cluster credibility. A similar assumption has been used in [4] to reduce the risk of collusion attacks and to speed up the convergence of the algorithm to compute reputation values. However, we consider that when clusterheads act as peers or designated agents they behave as others since they compete for resources.

On behalf of the community clusterheads periodically sign and disseminate reports on how nodes have behaved in their cluster; then, nodes add their report to the *trust token*. This token stores the reputation values associated to the activity of a node in different periods of time. It is personal and consists of *n*-entries, as shown in Fig. 1, to judge over more samples the behaviour of the node and possibly detect when nodes behave inconsistently.

The node identifier (Id) is used to bind a report to the node. This binding ensures that the node does not lend the token to others or that the node does not use the same token to enter a new community with multiple identities.
The cluster Identifier(c-Id) specifies the issuer of the token. This has a twofold meaning: 1) nodes in other communities use the correct cryptographic material to verify the signature of a report and 2) these nodes compute the capability of other clusters to *recommend* nodes, i.e., the cluster credibility.
The timestamp (T) specifies when the report has been issued. Clusterheads release reports at regular intervals and if a report is not present, null reputation value is associated to each missed one. The timestamp is also required to age the reports when the reputation value of a new entrant is calculated to account more recent reports.
The reputation value (R) gives an estimation of the behaviour of the node inside a cluster. The reported score is the node's global trust value when the token-entry is issued.
A quality value (Q) is associated to reputation. It represents the confidence that the clusterheads have in their reports as giving incorrect reports can decrease the credibility of a cluster, as defined later in Sec. 4.3. The quality value is computed based on the accordance of the reputation values received by the designated agents of a specific node. Clusterheads can lower the quality value, therefore risking less loss of credibility, in case there are only few samples to estimate the reputation value or the node behaves inconsistently in the community.
The digital signature is done on the hash of the report. On behalf of the cluster, a clusterhead signs the report to provide integrity and a proof of participation of the node to the system. The signature is verified by the members of other clusters to validate a report. This procedure avoids fake reports from the nodes and possible modification of the message. The report is not encrypted for two reasons: the report is disclosed to different clusters and nodes can track their reputation value.

4.3. The cluster credibility

In order to support mobility in virtual communities, we introduce the concept of *cluster credibility*, i.e., the confidence that a node has in the capability of a cluster to judge a node. We extend the computation of the initial reputation value by using this credibility factor to weigh the reputation values of an entrant node in a new cluster.

The update of the cluster credibility accounts for the behaviour of nodes entering a new community. If a cluster gives wrong reports about peers, its credibility rating is decreased and its subsequent reports count less on the reputation of another entrant peer coming from the same community. Similarly, if a cluster's report is consistently good, i.e., in agreement with the behaviour of the nodes in the new cluster, its credibility rating goes up.

The cluster credibility has an initial value of 0.5 and it is computed locally by nodes in other communities upon the agreement of the old reported reputation values and the behaviour of any new entrant node coming from the same cluster. The computation is similar to the confidence of a node for ROCQ and it is shown in eq. 2.

$$C_{mc}^{k+1} = \begin{cases} C_{mc}^{k} + \frac{(1-C_{mc}^{k}Q_{cj})}{2}(1 - \frac{|R_{cj}-O_j^{avg}|}{s_{mj}}), \\ \quad \text{if } |R_{cj} - O_j^{avg}| < s_{mj} \\ C_{mc}^{k} - \frac{C_{mc}^{k}Q_{cj}}{2}(1 - \frac{s_{mj}}{|R_{cj}-O_j^{avg}|}), \\ \quad \text{if } |R_{cj} - O_j^{avg}| \geq s_{mj} \end{cases} \quad (2)$$

C_{mc}^{k} is the local credibility of cluster c computed by peer m after k reports, O_j^{avg} is the opinion being currently reported on the new entrant j, Q_{cj} is the quality value of the cluster on the previous reported reputation value R_{cj} for j and s_{mj} is the standard deviation of all the reported opinions about peer j. The cluster credibility ratings are based on first-hand experience only and they are not shared with other peers to avoid the recursive problem of trusting nodes.

4.4. Operations to join a cluster

When a node joins the system, its initial reputation value is 0, which rates it as a uncooperative node, thus, it needs to provide services to increase its reputation. In this period it collects reports that are stored in the trust token. When a node moves from its current cluster, it enters a new community and submits the trust token, which is the ticket required to join an already formed cluster. The token is routed following the communication protocol specifications used inside the community and sent to the new node's designated agents. We do not consider the nomination of designated agents as, even if interesting, it is outside the scope of this paper; the reader can refer to [13].

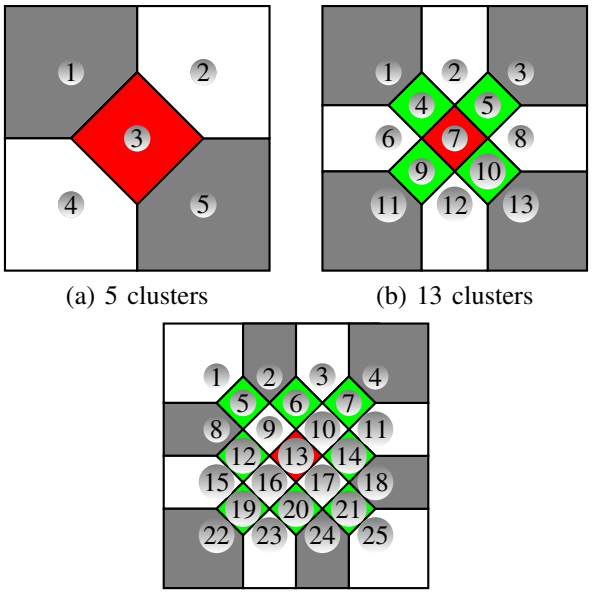
When a new node joins the community, the designated agents verify the integrity and the signatures of the token and compute the new reputation value which is stored locally for future use inside the cluster. These agents form the *first* reputation value by aging and weighing the information the token contains with the credibility of the clusters that have issued the entries. It is important noticing that a node can erase the entries contained inside the token, but these deleted entries count as negative transactions in the computation of the reputation value, as discussed earlier for the usage of the timestamp.

5. Network scenario

We simulate a mobility scenario in an area of $1,000m \times 1,000m$, which is divided in clusters representing the *virtual communities*, as shown in Fig. 2. We use the Canu mobility simulator [14] to create the trace files by using the *random waypoint model* [15] to simulate the mobility of the nodes. We assume that the area is open without obstacles, thus, the movement of the nodes is assumed to be free and it follows a straight trajectory to reach the destination. When a node reaches the border area the trajectory is reflected.

The simulation runs for $90,000s$ divided in time slots of $90s$ after which we update the position of the nodes to define the composition of the clusters. We simulate a total of 6 settings for the same area to test the token-based mechanism in different contexts. The scenarios are defined by the speed of the nodes and the number of clusters 5 (a), 13 (b) and 25 (c), as shown in Fig. 2.

We simulate two types of nodes: clusterheads, or more stable users, and nomadic users randomly placed in the area. The movement of nodes can be slow or fast and it depends



(a) 5 clusters (b) 13 clusters

(c) 25 clusters

Figure 2. Clusters in area of $1,000m \times 1,000m$

Table 1. Parameters' setting

Type of Nodes			
Total # Nodes	1,000	Initial # CHs per community	5
# Nodes Class I	315 − # CHs	# Nodes Class II	685

Slow Mobility		
	Speed	Stay Time
Class I	[0.18; 0.46] m/s	[30; 60] s
Class II	[0.18; 0.46] m/s	[36; 120] s
Clusterheads	[0.018; 0.046] m/s	[3,000; 6,000] s
Fast Mobility		
	Speed	Stay Time
Class I	[0.74; 1.85] m/s	[7.5; 15] s
Class II	[0.74; 1.85] m/s	[9; 30] s
Clusterheads	[0.074; 0.185] m/s	[750; 1,500] s

Table 2. Avg # of nodes changing cluster

	5 clusters	13 clusters	25 clusters
Slow Mobility	57.9	107.1	148.2
Fast Mobility	218.7	385.8	509

on the speed and the time a node remains in a position before moving again. For all settings, the nomadic users are divided in two classes to simulate a more heterogeneous population: the difference consists in the time nodes spend in a place and are identified as class I and class II. Table 1 shows the parameters used to derive the traces of the nodes' positions.

Due to the mobility, the cluster community changes. Table 2 shows the average number of nodes that move after each time slot: this number is function of the speed and the number of clusters. For instance, about half of the system population changes cluster when there are 25 clusters and the speed is high. On the contrary, about 6% of the nodes change community if there are only 5 clusters and the nodes move slowly. As for clusterheads, many exist in a community and they are assumed to have low mobility compared to other nodes, but in case they change cluster, new ones must be nominated. In this paper we do not deal specifically with the election process and we require that at least one clusterhead is present in the community.

We implement the token-based mechanism in Java as an extension to the ROCQ reputation management scheme. We use the mobility traces to construct the network topology defined by the position of the nodes and the number of clusters. Each cluster is organized in a Distributed Hash Table (DHT) to simplify the construction of the overlay topology and the assignment of designated agents [3]. Multiple agents are used to maintain a consistent view of the reputation values and nodes are labeled as designated agents only for peers in the same cluster. The use of a DHT is not required for small networks like the one we simulate, but the overlay topology of the community does not impact on the performance of the token-based mechanism extension compared to the normal functionality of the initial reputation management scheme.

6. Performance evaluation

We validate the performance of the token-based approach by comparing the results of this mechanism with the ROCQ reputation management scheme, also named hereafter basic reputation scheme. In both cases, when changing cluster nodes remove all the stored information, the reputation and credibility values of other peers and the quality values of the nodes with whom they have interacted, to simulate nodes always joining different communities.

In our experiments we use the parameters listed in Table 3. There are 5 initial clusterheads in each cluster and 6 designated agents to aggregate and store the reputation value of a node in a cluster. To compute the trust value, nodes can follow two strategies, indicated by *ro* and *or* in the plots respectively: 1) the reputation value is used and 2) the average opinion is used, if there exists past direct transactions between the same two nodes, otherwise the reputation value. Then, nodes use a deterministic threshold 0.5 to decide if peers are trustworthy. At each iteration, a node is randomly selected within the entire population and the interacting peer is chosen randomly within the same cluster. The result of the interactions is used to evaluate the performance of the reputation management system, such as the success rate of transactions defined as follows:

$$Success\ rate = \frac{\#Tr_{good} + \#Av_{malicious}}{Total\ \#\ of\ transactions} \qquad (3)$$

where $\#Tr_{good}$ is the number of interactions with good peers that go ahead and $\#Av_{malicious}$ the number of avoided interactions with malicious peers.

We run an initial number of transactions to bootstrap the reputation management system and the token-based mechanism, as shown in Table 3. This is required to have an initial reputation value for the nodes, an initial estimation of the clusters' credibility and initial reports inside the token. The cluster and node credibility are initially set to the uncertain value of 0.5: 0 means no confidence and 1 the node is fully confident in the reporting cluster/agent.

At regular intervals 5 reports are collected in a time slot of $90s$. The size of the token is limited to 100 records, thus, it stores the history of the node for the last 30 minutes; a smaller token could not account for the relevant history of the node, as few samples could not be sufficient to estimate the behaviour of the node. For each time slot, we simulate a different number of transactions in the system, indicated by 500 and $2,500$ iterations in the plots. Finally, the membership of the cluster is updated after each time slot.

We simulate malicious behaviour for the transaction and report. In the latter case, malicious nodes report the inverse of the amount of satisfaction they receive from an interaction or the inverse of the reputation values if they act as designated agents to subvert nodes' feedbacks. Hence, if $O \in [0,1]$ ($R \in [0,1]$) is the actual opinion (reputation) value, the value that is sent is $(1-O)$ $((1-R))$,

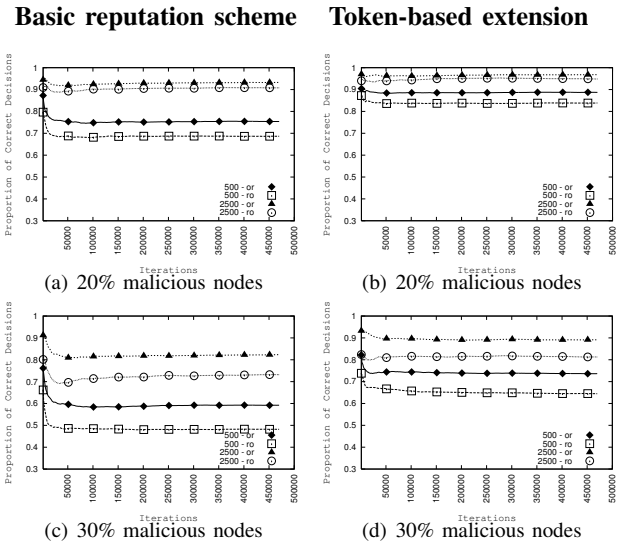
In the following sections we analyze the impact of the node speed, i.e., how often the cluster membership changes, the frequency of report collection and the size of the cluster on the capability of the reputation management schemes in identifying malicious nodes. Extensive simulations of the token-based approach for other cases can be found in [13].

6.1. Impact of the speed of the nodes

In Fig. 3 we compare the performance of the token-based mechanism, in terms of fraction of correct decisions, when the nodes move slowly in the area, as defined in Table 1. Fig. 3 shows that the token-based mechanism improves the performance of the reputation management

Table 3. Parameters' setting for the simulation of the reputation management scheme

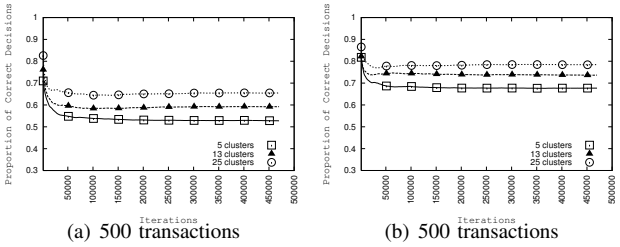
Transactions			
# Transactions	470,000	Before nodes' movement	500; 2,500
To bootstrap the reputation	1,000	To bootstrap the token approach	29,000
Simulation settings			
Topology	Random	Mobility model	Random waypoint
Experiments run	6	# Designated agents	6
Type of node maliciousness	Report and service	Type of decision	Deterministic
Trust threshold	0.5	Size of the Token	100



(a) 20% malicious nodes (b) 20% malicious nodes

(c) 30% malicious nodes (d) 30% malicious nodes

Figure 3. Proportion of correct decisions for 13 clusters and slow mobility of the nodes.

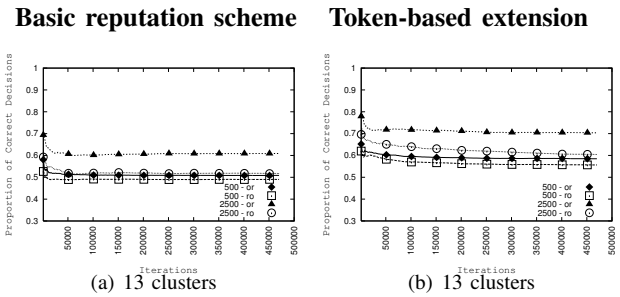


(a) 500 transactions (b) 500 transactions

Figure 4. Success rate for slow mobility of the nodes and 30% of malicious nodes when opinion is used first.

system, specifically, when the fraction of malicious nodes increases in the system (plots (c) and (d)). The improvement is greater when there are few interactions available to form an opinion or to estimate the reputation of the nodes. This is shown by the curves plotted for 500 transactions in each time slot with an increase of 15% of correct decisions compared to 10% for 2,500 transactions.

As expected, a decision based on direct experience (indicated by *or* in the plots) increases the success rate in



(a) 13 clusters (b) 13 clusters

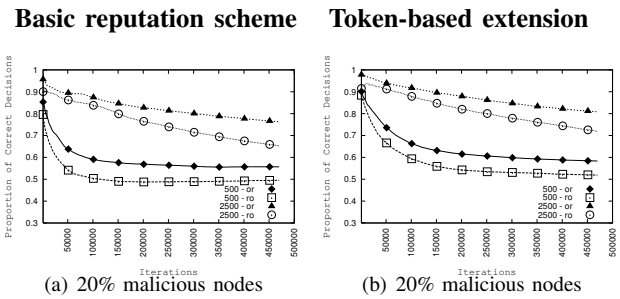
Figure 5. Success rate with fast mobility of the nodes and 30% of malicious nodes.

all cases and reduces the improvement of the token based solution over the basic scheme. In fact, our approach aims at improving the evaluation of the reputation, but in the simulated setting decisions are local and thus not biased by the maliciousness of the reporting agents.

In Fig. 4, we simulate the presence of a different number of communities in the area. We plot only the case when nodes decide to transact based on direct experience if available otherwise they use reputation, since it is closer to a real scenario. When there are few interactions in a time slot, the presence of more clusters gives a higher success rate, plots (a) and (b) in Fig. 4. This is true for both the traditional reputation scheme and the extended version as nodes interacts more frequently with the same nodes of the same cluster, and, as such, they can form a more accurate estimation of the nodes' trustworthiness. We expect that when the number of subsequent transactions increases the impact of the number of clusters is smoothed by a higher number of samples to evaluate reputation values.

In Fig. 5 we plot the success rate when the nodes move with fast mobility. The higher mobility does not allow nodes to stay in a cluster for the time sufficient to have an accurate estimation of the reputation value. In fact, the success rate decreases by 10% compared to slow mobility. We also expect that higher mobility slightly reduces the impact of the number of clusters on the system's performance as the same pair of nodes interacts less frequently. In particular this is true for big clusters because nodes change cluster less often (see Table 2).

Thus, we can conclude that small clusters enables the node to rely more on their direct experience, as they can



(a) 20% malicious nodes (b) 20% malicious nodes

Figure 6. Proportion of correct decisions for 13 clusters and slow mobility when nodes have inconsistent behaviour.

exploit the amount of satisfaction in previous transactions to predict the behaviour of the nodes. Moreover, the token helps nodes in taking decisions when they must evaluate the trustworthiness of newcomers.

6.2. Milking attack: inconsistent behaviour

When peers act maliciously in a consistent fashion while moving across clusters, they can be identified relatively quickly. However, if peers choose to act maliciously only in some clusters and for some transactions, a malicious behavoiur is harder to be detected. This type of attack, known as *milking*, causes the credibility of the clusters to be lowered as they do not provide accurate information due to the inconsistent behaviour of the nodes.

We simulate that a fraction of random nodes, equal to the percentage set initially for malicious nodes, changes behaviour with probability $p = 0.5$ two times in a slot. In Fig. 6 we plot the success rate when nodes move slowly in the system. A small percentage of nodes, that are initially malicious, decreases the performance of the system, in particular when nodes use reputation to decide their behaviour in a transaction. The milking attack is more effective when the number of transactions in a time slot is high, i.e., $2,500$. In this case, nodes form a consistent opinion of the members' trustworthiness and they put more confidence in their reports, sent to designated agents. Thus, when a node changes behaviour, the reputation value cannot predict how the node will behave in the future accurately. If it turns to be malicious, its attack has higher impact on the performance of the system as the node might have acquired *privileges* for being cooperative before.

We expect that when the percentage of malicious nodes increases and there are few transactions in a time slot, the gain of the token-mechanism is limited since the node has a lower confidence value on the reporting clusters. Indeed, clusters, which have scored these nodes as cooperative (or malicious in the opposite case), are less credible for their following reports if nodes change behaviour.

7. Discussion

The token-based mechanism improves the proportion of correct decisions taken by nodes in all scenarios we have analyzed. Specifically, it enables to account for transactions in other communities to detect misbehaving nodes timely. The fact that nodes are rewarded for their good behaviour promotes cooperation which is a basic property that self-organizing system must have to function properly.

A potential criticism to the token-based approach consists of the possibility for the nodes to fake their old reports to hide their malicious behaviour or to simply sell/lend their tokens to other nodes. We tackle these problems by imposing that the score for a node is bound to the identity of the nomadic node and it is digitally signed by the community, as discussed in Sec. 4.2.

As regards privacy, the token-based extension requires nodes to reveal which communities the node has joined in the past, thus, the location of the node is fully traceable. However, as we have discussed in Section 4.2, a node has completely control over its token and it can decide which information wants to reveal and to which community.

We now analyze the security solutions to mitigate the impact of impersonation and bad mouthing attacks. The token-based scheme uses digital signatures to ensure integrity and correctness of the reports. The signature must be recognized to belong to the cluster otherwise the report is not valid. We assume the existence of an off-line certification authority that issues certificates to the nodes, associated with a pair of keys. This is required by any reputation management scheme to avoid non-repuditiation of an opinion.

Clusterheads can digitally sign the reports simply, but this solution has two main drawbacks: 1) the leaders in a cluster are many and they do not share the same key pair; 2) the verifiers must know the public key of the signer, i.e., each mobile node should store the public keys of all possible signers present in the system. The storage is not a big issues for mobile devices; in fact a typical public key has a size that ranges from 512 to 2048 bits if no elliptic cryptography is used, thus, in the worst case (2048 bits) 1 MB is sufficient to store 4096 keys.

The main issue of this solution consists of the form of the signature: if clusterheads use their own private key to sign the report, this report is associated to the signer and not to the cluster, i.e., what we want to achieve. In this setting, malicious nodes can collude and generate valid fake reports unless the verifier has the complete list of authorized clusterheads, which might not be feasible.

To implement a signature on behalf of the group and to guarantee anonymity of the signer two schemes are *group signature* [16] and *ring signature* [17]. In the former, an authority generates the private signing keys and distributes them to the members of the group which uses its private signing key to generate the signature; a verification key,

common for the group, is used to validate the signature. In *ring signature* a clusterhead, responsible for the report, creates an ad-hoc ring signature composed by other cluster entities *without* their approval or their aid. This mechanism preserves anonymity of the signer, but there is no control on ad hoc formation of groups and the verifier can hardly know the members of a group authorized to sign a message.

Thus, we propose to use *Id-based cryptography* [18] to create a strong relationship between the signer and the cluster. In Id-based cryptography the public key is an identifier and the private key can only be generated from the public key by a trusted authority. In our setting, the identifier is a tuple that contains the node identifier and the cluster id, e.g. *Id.clusterId*, to ensure that a signature has been issued by a clusterhead. In a dynamic environment when clusterheads might leave the community, we might want to give the opportunity to outsource other nodes the responsibility to sign reports. To serve our goal, we exploit Id-based signatures schemes organized in a hierarchical structure as proposed in [19].

8. Conclusions

In this paper we propose a mechanism to solve the bootstrap problem and enable the use of reputation management schemes in mobile P2P networks when a node lasts not for long in a community. We present a token-based solution, that allows nodes to carry information on their reputation defined in other virtual communities.

This token-based extension moves the burden of storing personal information to the nodes and enables the correlation of reputation values earned in different communities. Simulation results show that our solution is effective to eliminate the problem of bootstrapping reputation values in new communities and to timely detect malicious nodes. We show that the token-based solution increases the success rate upto 15% in the presence of 30% misbehaving nodes.

As future work, we will consider different mobility models and scenarios to enhance our reputation scheme and other type of attacks, such as collusion.

Acknowledgment

Work partially supported by projects DAMASCO funded by Italian Ministry of Research and BIONETS (FP6-027748) funded by the FET program of European Commission.

References

[1] W. Kellerer, Z. Despotovic, M. Michel, Q. Hofstatter, and S. Zols, "Towards a Mobile Peer-to-Peer Service Platform," in *SAINT-W*, Jan. 2007.

[2] R. G. Cascella, "The "Value" of Reputation in Peer-to-Peer Networks," in *IEEE CCNC*, Jan. 2008, pp. 516–520.

[3] A. Garg, R. Battiti, and R. Cascella, "Reputation management: Experiments on the Robustness of ROCQ," in *ISADS - First International Workshop on Autonomic Communication for Evolvable Next Generation Networks*, Apr. 2005.

[4] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *WWW*, May 2003.

[5] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security*, Sep. 2002.

[6] R. G. Cascella, "Costs and Benefits of Reputation Management Systems," in *IEEE WoWMoM*, Jun. 2008.

[7] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems: Facilitating trust in internet interactions," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[8] X. Liu and L. Xiao, "hiREP: Hierarchical reputation management for peer-to-peer systems," in *ICPP*, Aug 2006.

[9] F. Li and J. Wu, "Authentication via ambassadors: A novel authentication mechanism in manets," in *MilCom*, Oct. 2007.

[10] A. Garg, A. Montresor, and R. Battiti, "Reputation Lending for Virtual Communities," in *ICDEW*, Apr. 2006.

[11] S. Marti and H. Garcia-Molina, "Taxonomy of trust: categorizing p2p reputation systems," *Computer Networks*, vol. 50, no. 4, pp. 472–484, 2006.

[12] C.-A. La and P. Michiardi, "Characterizing user mobility in second life," in *WOSN*, Aug. 2008.

[13] R. G. Cascella, "Application of reputation management systems in autonomic communication networks," Ph.D. dissertation, DISI - University of Trento, Nov. 2007.

[14] "Canu mobility simulation environment (canumobisim)," http://canu.informatik.uni-stuttgart.de/mobisim/.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[16] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: formal definition, simplified requirements and a construction based on trapdoor permutations," in *Advances in cryptology - EUROCRYPT*, ser. LNCS, vol. 2656, May 2003, pp. 614–629.

[17] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret: Theory and applications of ring signatures," in *ASIACRYPT*, Dec. 2001.

[18] D. Boneh and M. Franklin, "Identity based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[19] T. H. Yuen and V. K. Wei, "Constant-size hierarchical identity-based signature/signcryption without random oracles," Cryptology ePrint Archive, Report 2005/412, 2005.