Distributed Reputation Management for Service-Oriented Peer-to-Peer Enterprise Communities

Michele Amoretti* and Francesco Zanichelli

Department of Information Engineering, Università degli Studi di Parma, 43124 Parma, Italy
E-mail: michele.amoretti@unipr.it
E-mail: francesco.zanichelli@unipr.it

*Corresponding author

Abstract: Peer-to-peer service sharing might allow for unmediated, almost instantaneous inter-firm collaboration, yet robust reputation management techniques are strongly required to withstand fraudulent actions of malicious peers. In this paper we illustrate a distributed reputation management system for service-oriented peer-to-peer networks, called SAFE, which exploits voting and effectively copes with trust misrepresentation attempts. Simulation results show that SAFE allows peers to select high-quality services, even when a large extent of malicious providers and advisors are active in the community.

Keywords: Service-Oriented Architectures; Peer-to-Peer; Reputation Management.


Biographical notes: Michele Amoretti received the Dr. Ing. (Master) degree in Electronic Engineering in 2002, and the Ph.D. degree in Information Technologies in 2006 from Universit degli Studi di Parma (Italy). Currently, he is Research Associate at the same university. His research focuses on modeling and simulation of large-scale distributed systems, in particular those based on the peer-to-peer paradigm, complex adaptive systems and autonomic computing, design and development of service-oriented architectures, as well as wireless sensor networks. He has published over 80 technical papers in refereed international journals and conference proceedings.

Francesco Zanichelli received the Dr. Eng degree (Master) in Electronic Engineering at the University of Bologna and the Ph.D. degree in Information Technologies at the University of Parma. He is currently an Aggregate Professor with the Department of Information Engineering of the University of Parma, where he is teaching the Operating Systems course. After several years devoted to the study of intelligent robotics and robot programming, current research interests of Francesco Zanichelli include service-oriented peer-to-peer systems and middleware, as well as distributed multimedia architectures and protocols, including security and Quality of Service.

1 Introduction

One of the most effective solutions that electronic commerce companies have explored as a response to the problem of presenting pertinent products to each customer is personalization. One of the major technologies for personalization is collaborative filtering, a systematic method which provides each user with personalized recommendations, based on his/her preferences and on other users’ evaluations. A similar approach can be applied to virtual organizations in which companies provide services not only to end users but also to each other, in a peer-to-peer (P2P) fashion. Within a highly dynamic and very rich service market, the role of Reputation Management Systems (RMSs) is essential to ensure that peers obtain from experienced service consumers reliable information on the quality of services they are going to select (Watanabe et al., 2009; Gao et al., 2014). However, most state-of-art RMSs for P2P systems address content sharing (Mondal et al., 2010), while limited research considers service sharing.

In this work, we illustrate a Distributed RMS (DRMS) for service-oriented P2P networks, called Service Advisors For E-business (SAFE), that effectively
copes with trust misrepresentation attempts. SAFE relies on a decentralized voting scheme, with a Distributed Hash Table (DHT) to store and share information about advisors, i.e., peers with direct experience on specific service providers. Each service transaction is followed by an evaluation expressed in terms of a set of application-specific Quality of Service (QoS) parameters, e.g., timeliness and accuracy, in order to improve the objectivity of the assessment. Service provider selection, albeit not deterministic, is based on available reputation, which is ultimately dependent on advisors’ evaluations.

The paper is organized as follows. Section 2 illustrates with more details the problem of distributed reputation management, starting from basic definitions. Section 3 analyzes state-of-art solutions. Section 4 illustrates the SAFE architecture. Section 5 presents a simulation-based performance evaluation, where SAFE is compared both to a P2P network where no reputation system is available and to the EigenTrust scheme (Kamvar et al., 2003). Finally, Section 6 concludes the paper with a summary of obtained results and future work.

## 2 Problem Statement

In a P2P service sharing scenario, a peer may provide one or more services, characterized by non-negotiable QoS parameters. No specific technique for service description is considered here. We assume that services can be compared to assess their equivalence or similarity, so that multiple providers may offer the same service within a competitive market. Moreover, we assume that service consumers, which are themselves peers, are enabled to search for services, to discover multiple providers for the service of interest, and to select the best one according to an evaluation process that takes into account QoS requirements. The service selection strategy is the core problem we tackle in this paper.

The lack of centralized control for resource access and provision, and the high dynamism of the P2P network, call for distributed mechanisms for peer behaviour control (Rasmussen and Jansson, 1996), based on two main concepts: trust and reputation. Trust is the subjective probability with which a peer will perform a particular action, both before we can monitor such an action, and in a context in which it affects our own action (Gambetta, 1990). Orthogonally, reputation is the evaluation of a peer based on experience or recommendations. In general, trust and reputation are non-transitive and subjective.

Although decentralized authentication mechanisms have been proposed, to cope with trust and reputation in P2P networks (Pathak and Iftode, 2006; Wierzbicki et al., 2005), still unresolved issues are caused by incomplete a-priori knowledge of all possible service providers. We adopt the following categorization of threat models (Kamvar et al., 2003):

- **Individual malicious peers** always provide a poor service, when selected as providers.
- **Malicious collectives** are sets of individual malicious peers that cooperate to artificially increase their reputation values.
- **Malicious collectives with camouflage** usually (but not always) provide corrupted services, when selected as providers.

To minimize the risk of corrupted service selection, the (D)RMS must dictate how to aggregate a peer’s own experience with other peers’ recommendations, in order to build trust for each discovered service provider and to select the most suitable one. Moreover, the (D)RMS must point the way to build and update trust in recommendation providers (thereafter termed service advisors).

## 3 Related Work

We propose a classification with three categories of RMS models, and we discuss their benefits and drawbacks in order to clearly motivate the underpinnings of the architecture we propose.

### 3.1 Local Evaluation

The simplest approach is the Local Evaluation model, for which after each transaction the consumer evaluates the quality of the retrieved resource/service and updates the local reputation value of the provider. In this model, reputation and trust are coinciding concepts. For example, Marti et al. (2004) define the reputation/trust of peer \( p_j \) computed by peer \( p_i \) as:

\[
R_{ij} = \frac{\text{sat}(i,j)}{\text{sat}(i,j) + \text{unsat}(i,j)}
\]

i.e., the number of satisfactory transactions versus the total number of transactions. Another, more recent, example of local reputation system is iTrust (Dai et al., 2013), which is designed in compliance with stringent requirements of mobile ad-hoc networks. Indeed, local evaluation does not flood the network with messages, since reputation information is not exchanged among peers. This is also the main drawback of the Local Evaluation approach which is almost inapplicable when the network becomes very large and dynamic, as, in this case, it is highly probable that discovered services come from unknown (i.e., untrusted) peers.

### 3.2 Voting

In a system based on the Voting model (Aringhieri et al., 2006; Chang and Wong, 2011; Kamvar et al., 2003; Song et al., 2005), the consumer chooses the best provider according to its own previous experiences and those provided by other peers. Chosen information
providers can be neighbors, i.e., peers which are directly connected to the consumer, or remote peers which have been discovered in the overlay network. The main advantage of this approach is that each peer can rely on a distributed knowledge base.

We looked at EigenTrust (Kamvar et al., 2003) as a touchstone for the solution we propose in this paper, which is based on the Voting model, too. The main objective of EigenTrust is the definition of a global trust value for each peer, i.e., an absolute value of trust that the whole system assigns to each peer. Local trust values are aggregated based on transitive trust, i.e., considering that if a peer provides a good service, then probably it also provides good advices. In our opinion this assumption appears too strong; transitive trust should be conditional, rather than assumed a priori.

On the other hand, the EigenTrust algorithm has many interesting features, namely self-policing, anonymity preservation, reputation improvement as a consequence of lasting fair behaviour, overhead minimization, and robustness against malicious collectives — for these reasons, it is still often used as a starting point for other protocols (Fedotova and Veltri, 2009).

3.3 Transaction Certificates

The third strategy is to use Transaction Certificates (Gupta and Somani, 2004; Han et al., 2014), generated by involved parties and including the score assigned to just completed transactions. At the end of a successful service completion, the client signs a satisfaction certificate (a form of digital certificate) and gives it to the server. The satisfaction certificate is a proof that the server provided a good or bad service to the client. The server, upon receiving the satisfaction certificate, sends it to known nodes.

Such a strategy requires the presence of a certification authority (CA), i.e., an entity which issues certificates attesting that, in a PKI scheme, the public key contained in the certificate belongs to the person, organization, server or other entity noted in the certificate. Moreover, the CA could maintain a certificate revocation list (CRL), i.e., a list of certificates which have been revoked, are no longer valid, and should not be relied on by any system user.

3.4 Discussion

We already pointed that the Local Evaluation model (Marti2004; Dai2013) is easy to implement but of little use if the set of providers shows high dynamism. Consequently, we focus here on the Voting and Transaction Certificates models, which both rely on information exchange among peers.

In the Voting model (Arighieri et al., 2006; Chang and Wong, 2011; Kamvar et al., 2003; Song et al., 2005), advices are generated on demand and their motivation cannot be verified. Conversely, in the Transaction Certificates scheme (Gupta and Somani, 2004; Han et al., 2014), each advice is clearly and objectively tracked, for which it is possible to check for the trustworthiness of the issuer and reduce the influence of malicious collectives.

With respect to overhead, in the Voting scheme there is only one kind of message exchange, the one in which consumers collect advices about providers offering resources they are interested in. Interactions among peers are more complex in the Transaction Certificates model, and certificates may flood the network if propagation constraints are not set.

Concerning vulnerability, both schemes are resistant to one-shot attacks, in which a malicious peer acts unfairly and immediately leaves the network. The Voting model is vulnerable to malicious collectives. The Transaction Certificates model can be subject to many kinds of attack, ranging from DoS attacks against CAs, to false certificate emission, to attacks on certificate storage sites. Certificate replication is a possible solution, although it introduces further overhead to maintain data consistence.

With respect to storage requirements, in the Voting model — where each node is responsible for its own advices — they increase with the number of active providers involved in transactions. The Transaction Certificates model requires that a set of copies of each issued certificate be spread in the network, with all peers (or a subset) responsible for data maintenance, and storage requirements grow proportionally to the total number of performed transactions.

Considering all these aspects, we chose to adopt the Voting approach and we defined the SAFE framework, whose detailed description is given in next section.

4 SAFE Architecture

Service Advisors For E-business (SAFE) is a DRMS based on the Voting model, where each consumer’s assessment on a service provider is related to specific QoS interests. This approach may be defined probabilistic, since an assessment represents the conditional probability that the offered service is high-valued, assuming a number of QoS constraints defined by the requester, before consuming the service. Moreover, SAFE adopts a DHT-based approach to share information about advisors in the peer-to-peer network.

4.1 DHT-based information storage

In DHT-based systems, the peer which is responsible for storing a specific information is determined with a hash function within O(1) time, and its location is found within O(log N) time. Examples are Chord (Stoica et al., 2001) and Pastry Rowstron and Druschel, 2001). In SAFE, the DHT is used to store information about past peer-to-peer interactions. In detail, the DHT stores |ID(p_i), ID(p_l), ID(s_m), n_{jl}| tuples, where ID(p_l) identifies a peer (the consumer) that has concluded its n_{jl}-th transaction for service ID(s_m) with
the peer identified by $ID(p_j)$ (the provider). For each $jl$ couple, there is a set of peers that store the related tuple and its updates. In this set, one peer is the main tuple storing responsible, while the others store consistent replicas, for the sake of robustness.

To avoid that malicious peers providing deceptive advices collect a large number of transactions, the DHT stops storing the updates if a suspicious fast growth of the number of interactions with the same consumers is noticed. Data integrity is guaranteed by digital signatures, i.e., the number of transactions of a service provider is hashed and encrypted with the provider’s private key. This digest is stored in the DHT, attached to the information, whose integrity can be checked by comparing its hash with the digest decrypted with the provider’s public key.

4.2 Provider selection

To assess the reputation of provider $p_j$, with reference to service $s_m$, a peer has to apply Algorithm 1. Throughout this section we omit the $m$ index in order to alleviate the burden on the notation, although it is constantly implied.

Algorithm 1:

1: for each provider $p_j$, search the DHT for $[ID(p_j), ID(p_l), ID(s_m), n_{jl}]$, with $l \neq j$
2: for each provider $p_j$, compute the total number of performed transactions
3: if the total number of transactions is over a certain threshold then
4: exclude from the list of advisors the consumers that have performed too many interactions
5: choose at least $N_A$ most experienced advisors and ask them for votes about provider $p_j$
6: weigh the $n_A$ received votes, using eq. (10)
7: aggregate votes in a global reputation value $R_j$, using eq. (11)
8: end if

For each of the $N_P$ providers that offer the service of interest $ID(s_m)$, the consumer applies Algorithm 1 and finally chooses the one with highest reputation:

$$j = \text{argmax}_j R_j. \quad (2)$$

After the transaction is completed, the consumer updates its opinion about the service provider, as well as trust parameters associated to the contacted advisors.

If the number of available advisors for a given provider is $n_A < N_A$, the provider is considered to be lacking adequate evaluation by other consumers (unrated). In general, $N_P = N_p^u + N_p^k$, where $N_p^u$ is the number of unrated peers, while $N_p^k$ is the number of rated peers.

Within the service provider selection process, three situations may arise:

- **Worst case** - All providers are unrated ($N_p^k = 0$). In this case, the provider is chosen on the basis of personal experience, if available, otherwise it is randomly chosen with uniform probability.

- **Intermediate case** - Some providers are unrated ($0 < N_p^u < N_p$). It is the most frequent case, to be addressed in the following way. The fraction of unrated providers $u$ is computed as

$$u = \frac{N_p^u}{N_p}. \quad (3)$$

The sum of unrated peers’ reputation values is given by the following estimate:

$$\sum_{N_p^u} R = \frac{\sum_{N_p^u} R}{1 - u u}. \quad (4)$$

The total reputation value of the available $N_P$ providers is

$$\sum_{N_P} R = \sum_{N_p^u} R + \sum_{N_p^k} R. \quad (5)$$

The more the reputation of the provider, the more the probability of being chosen. The choice is not deterministic, in order to avoid the concentration of all service requests on the same highly reputed providers, with the risk of unlimited reputation increase for some peers, and perpetual avoidance for the others. Thus, a probability $P_j$ of being chosen is assigned to each $j$-th available provider.

If the provider is known, $P_j$ depends on the reputation value $R_j$, which is computed from existing votes, using the following formula:

$$P_j = \frac{R_j}{\sum_{N_P} R}. \quad (6)$$

To each unknown provider is assigned the same probability

$$P_j = P^u = \frac{\sum_{N_p^u} R}{N_p^u \sum_{N_P} R}. \quad (7)$$

- **Best case** - All providers are known ($N_p^u = 0$). A probability of being chosen is assigned to the $j$-th provider, depending on the reputation value $R_j$, which is computed from advisor’s votes and from personal experience, using equation (6).

The reputation value $R_j$ is computed as a weighted sum of external advices (the aggregated reputation value $R'_j$) and personal experience $E_j$ gained from the consumer peer on the $j$-th service provider:

$$R_j = \beta E_j + (1 - \beta) R'_j \quad (8)$$

where $\beta \in [0,1]$ is the personal experience weight.
4.3 Computing the aggregated reputation value $R_j^p$

Suppose $n_A$ advisors are able to provide their advices about provider peer $p_j$ to peer $p_i$, according to the $h$ QoS parameters that have been agreed for service $s_m$.

For the $q$-th QoS parameter, the value of trust that peer $p_i$ (one of the $n_A$ advisors) assigns to $p_j$ is

$$t_{ij}^q = \frac{T_{sat}^q i_j}{T_{tot}^q i_j}$$

(9)

where $T_{sat}^q i_j$ is the number of satisfactory transactions between peer $p_i$ (in the past acting as a consumer) and peer $p_j$ (provider), while $T_{tot}^q i_j$ is the total number of transactions between the same peers.

The partial reputation value that peer $p_i$ assigns to peer $p_j$, associated to the $q$-th parameter, is the weighted average of the trust values provided by the $n_A$ advisors:

$$r_j^q = \frac{\sum_{l=1}^{n_A} w_l t_{ij}^q}{\sum_{l=1}^{n_A} w_l}$$

(10)

where $t_{ij}^q$ is the trust value that peer $p_i$ maintains in peer $p_j$ as an advisor.

The aggregated reputation value $R_j^p$ that peer $p_i$ assigns to peer $p_j$ is the weighted sum of these partial reputation values, considering all the $h$ QoS parameters:

$$R_j^p = \sum_{q=1}^{h} w_q r_j^q$$

(11)

with $w_1 + w_2 + .. + w_h = 1$.

4.4 Updating trust values in advisors

Once the provider has been chosen and the transaction performed, the consumer quantifies its satisfaction according to the $h$ QoS parameters we already mentioned. The partial satisfaction value $S_j$ is 1 if the transaction is considered satisfactory in relation to parameter $q$, otherwise it is 0. The aggregated satisfaction value for provider $p_j$, i.e., the weighted sum of partial satisfaction values, is

$$S_j = \sum_{q=1}^{h} w_q S_j^q$$

(12)

with $w_1 + w_2 + .. + w_h = 1$.

Parameter $\alpha$ specifies the importance of the last transaction with respect to past history, in order to establish the trust value in advisor $p_l$ ($\alpha = 0$ means that only the last transaction is considered, while $\alpha = 1$ means that only past history is considered; all other values of $\alpha$ between 0 and 1 mean that all transactions are considered). We also define parameter $\epsilon$, whose value is $+1$ if the transaction was satisfactory, i.e., $S_j^q \geq S^q$, and peer $p_l$ provided a positive advice; it is $-1$ if the transaction was not satisfactory and peer $p_l$ provided a positive advice; it is 0 in the other cases.

The updated trust value that peer $p_i$ assigns to advisor $p_l$ is

$$t_l^i(n+1) = \alpha t_l^i(n) + (1-\alpha)[\epsilon + t_l^i(n)]$$

(13)

The purpose of the $\alpha$ parameter is to reflect the importance of a single interaction, the last, compared to the previous ones. It allows for faster or slower modification of the trust put into the queried advisors, and it can be used as a further defense against the action of malicious advisors.

5 Simulation of different scenarios

Using simulations we have assessed the performance of SAFE and compared it with that of no-RMS (where service providers are randomly selected) and EigenTrust (Kamvar et al., 2003). According to the problem statement discussion (Section 2), the service search and provider discovery processes have not been simulated.

In the simulated scenario, each peer belongs to one of the following sets:

- **Honest Peers (HP):** peers that provide high-quality services, consume services, and always provide truthful advices.
- **Malicious Providers (MP):** peers that only provide low-quality services, neither consuming services nor providing advices.
- **Malicious Advisors (MA):** peers that provides good services, consume services, and provide untruthful, deceptive advices.

We simulated a network of $N = 5000$ peers. In the following, when we refer to service consumers we mean honest peers and malicious advisors, all together: $N_{SC} = |HP| + |MA| = N - |MP|$. All service consumers are providers of good services. $N_{MP}$ is the fraction of malicious providers with respect to $N$, while $N_{MA}$ is the fraction of malicious advisors with respect to $N_{SC}$.

Table 1 summarizes the main simulation settings of the performed experiments.

5.1 Simple malicious collective

We measured the fraction of high-quality service selections versus different distributions of malicious providers, in order to show the steady state performance of SAFE. For this study, we considered malicious providers with low-quality services only. An example of network configuration and service distribution used our simulations is shown in figure 1.

In Figure 2, the curve that refers to SAFE with $N_{MA} = 10\%$ is compared with the one of the no-RMS scenario. With SAFE, the percentage of obtained high-quality services is always larger. With both strategies, when $N_{MP}$ increases, the fraction of high-quality service selections decreases. The trend is linearly dependent on
Table 1 Parameter values used in the evaluation of SAFE by means of simulations.

<table>
<thead>
<tr>
<th>Network</th>
<th>( N )</th>
<th>number of nodes</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{MP} )</td>
<td>% of malicious provider peers</td>
<td>10%, 20%, ..., 90%</td>
<td></td>
</tr>
<tr>
<td>( N_{MA} )</td>
<td>% of malicious advisors</td>
<td>10%, 40%, 70%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Services</th>
<th>( h )</th>
<th>number of evaluation parameters</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta )</td>
<td>personal experience weight</td>
<td>0.15 ([0, 1])</td>
<td></td>
</tr>
<tr>
<td>( \alpha )</td>
<td>last transaction weight (in advisor trust computation)</td>
<td>0.2 ([0, 1])</td>
<td></td>
</tr>
<tr>
<td>( N_P )</td>
<td>service invocation probability</td>
<td>0.5 ([0, 1])</td>
<td></td>
</tr>
<tr>
<td>( N_A )</td>
<td>evaluated service providers</td>
<td>10 ([1, \infty])</td>
<td></td>
</tr>
<tr>
<td>( P_T )</td>
<td>minimum number of queried advisors</td>
<td>5 ([1, \infty])</td>
<td></td>
</tr>
<tr>
<td>( N_{MP} )</td>
<td>maximum number of queried advisors</td>
<td>100 ([1, \infty])</td>
<td></td>
</tr>
<tr>
<td>( N_{MA} )</td>
<td>minimum number of transactions per advisor</td>
<td>1 ([1, \infty])</td>
<td></td>
</tr>
<tr>
<td>( N_{SC} )</td>
<td>transaction percentage threshold</td>
<td>90% ([0, 100])</td>
<td></td>
</tr>
<tr>
<td>( N_{MP} )</td>
<td>suspicious individual transaction percentage</td>
<td>40% ([0, 100])</td>
<td></td>
</tr>
<tr>
<td>( N_{MA} )</td>
<td>maximum transaction statistics update derivative</td>
<td>10.0 ([0, \infty])</td>
<td></td>
</tr>
<tr>
<td>( S_T )</td>
<td>service evaluation satisfaction threshold</td>
<td>0.6 ([0, 1])</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation</th>
<th>number of simulation cycles per experiment</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>number of experiments over which results are averaged</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 1 Network configuration with 30\% honest peers (HP), 50\% malicious providers (MP), and 20\% malicious advisors (MA). The small pie represents the quality distribution of services. Since only MPs provide low-quality services, the distribution is 50\% low-quality services and 50\% high-quality services.

Figure 2 The fraction of high-quality service selections in a network where some peers form a malicious collective of service providers and advisors. In this plot we consider all service consumers.

\( N_{MP} \) when providers are randomly selected, while it becomes much more smooth when SAFE is used. In fact, only with \( N_{MP} > 80\% \) it is possible to negatively affect SAFE’s performance.

Augmenting the fraction of malicious advisors is a means to reinforce malicious providers against honest consumers. The dashed and dotted curves in Figure 2 show the percentage of obtained high-quality services when SAFE is exploited, and the fraction of malicious advisors is respectively 40\% and 70\% of the set of service consumers \( (N_{SC}) \). We can observe an apparently contradictory behaviour emerging when the fraction of malicious providers is over 70\%, which can be justified by the fact that malicious advisors know which peers are malicious providers and do not choose them, when searching for high-quality services. Preposterously, if the network were made only of malicious providers and malicious advisors, malicious providers would never be invoked.

Figure 3 The fraction of high-quality services selected by honest peers, in a network where some peers form a malicious collective of service providers and advisors.
Figure 3 shows the same kind of results as Figure 2, although only honest peers are considered. The malicious collective is effective only when \( N_{MP} > 50\% \) and \( N_{MA} \geq 70\% \). When SAFE is running and \( N_{MA} = 10\% \), the fraction of obtained high-quality services is larger, with respect to the no-RMS scenario. In both cases, when \( N_{MP} \) increases, the fraction of high-quality service selections decreases, but while the trend is linearly dependent on \( N_{MP} \) when providers are randomly selected, it is much more smooth when SAFE is used. In the worst considered case, i.e., \( N_{MA} = 70\% \), the fraction of selected good services falls under 50\% only if \( N_{MP} > 60\% \). Without SAFE, in the same conditions the fraction of selected good services is less than 40\%. Remarkably, with an high fraction of malicious advisors (\( N_{MA} = 70\% \)), SAFE is expedient only when \( N_{MP} > 35\% \).

The dashed and dotted curves in Figure 3 show the percentage of obtained high-quality services when SAFE is running, and the fraction of malicious advisors is respectively 40\% and 70\%. Once again, choosing services...
using SAFE improves greatly over performing random choices.

Figure 4 shows that SAFE is more efficient than EigenTrust, although their performances tend to converge for high values of the percentage of malicious advisors. For this particular experiment we considered the whole set of service consumers, including both honest and malicious advisors.

In Figure 5 we show the same kind of results of figure 4, but considering only honest peers. These graphs clearly show that SAFE, with respect to EigenTrust, ensures higher preservation to honest peers against malicious collectives. With 70% malicious advisors, the fraction of malicious providers must be over 50% in order to significantly damage honest peers.

In Figure 6, we refer the fraction of malicious advisors to the whole network size N. The curve related to the $N_{MA} = 10\%$ case, in the first window, starts from its minimum value and increases towards the steady-state value. The trend is monotonically increasing with decreasing slope. The other curves in the same window have different trends, in particular the one related to the $N_{MA} = 70\%$ case. The performance has a minimum in correspondence of the tenth simulation cycle. The initial good value of the fraction of selected high-quality services is due to the fact that malicious advisors do not choose malicious providers, when searching for services.

5.2 Convergence measures

Another important metrics is the number of iterations required by SAFE to reach the convergence, i.e., the condition in which no services provided by malicious providers are considered anymore for selection, and no more advices provided by malicious advisors are taken into account.

We have simulated the scenario of all peers turning on SAFE at the same time. Malicious advisors start searching for services, avoiding malicious providers (together they form a malicious collective and know each other) and providing deceptive advices. Honest peers, on the other hand, search for services, being unaware of malicious providers and malicious advisors, and providing truthful advices. During the transient state, in which SAFE has not provided all its benefits yet, there is the major consumption of low-quality services. After some interaction cycles, acquired knowledge helps honest peers in protecting themselves against the malicious collective.

In Figure 6, we refer the fraction of malicious advisors to the whole network size N. The curve related to the $N_{MA} = 10\%$ case, in the first window, starts from its minimum value and increases towards the steady-state value. The trend is monotonically increasing with decreasing slope. The other curves in the same window have different trends, in particular the one related to the $N_{MA} = 70\%$ case. The performance has a minimum in correspondence of the tenth simulation cycle. The initial good value of the fraction of selected high-quality services is due to the fact that malicious advisors do not choose malicious providers, when searching for services.
high-quality services. We observe that the more $N_{MA}$ increases, the slower is the convergence towards the steady-state value. This is reasonable, as the increment of deceptive information spread by malicious advisors further prevents honest peers from consuming high-quality services, in the transition state. The $N_{MP} = 50\%$ case, illustrated by the second window in figure 6, is similar, but with some quantitative differences. Initial values of the three graphs are much more distant than in the previous case, and steady-state values are quite lower.

It is interesting to compare these results with those of figure 7, which shows the transient behaviour of the system when EigenTrust is applied by all nodes. It appears that the system reaches the steady state more quickly than in the case of SAFE, and independently from the fraction of malicious providers. This aspects minimally compensate the results observed in section 5.1, i.e., the minor fraction of selected good services, with respect to the case in which SAFE is adopted.

### 5.3 Malicious collective with camouflage

The considered threat model can be difficult to contrast if some malicious providers host high-quality services, together with bad ones. An example of such malicious collective with camouflage is illustrated, in terms of network configuration and service distribution, in Figure 8.

![Figure 8](image)

**Figure 8** Network configuration with 30% honest peers (HP), 50% malicious providers (MP), and 20% malicious advisors (MA). The small pie represents the quality distribution of services. In this case (malicious collective with camouflage) MPs provide low-quality services but also high-quality ones, resulting in a distribution with 35% low-quality services and 65% high-quality services.

We simulated three different situations, respectively with 15%, 30%, and 45% high-quality services hosted by malicious providers.

Figure 9 considers only honest consumers. Compared to the scenario illustrated in Section 5.1, we observe that SAFE’s performance is slightly worsened by malicious peer camouflages. Moreover, we observe that if the fraction of malicious providers and the fraction of good services offered by malicious providers increase, SAFE’s performance is less and less affected by the number of malicious advisors. The reason of such an apparently contradictory behaviour is that the total number of available good services is higher in this scenario than in the simpler one illustrated in Section 5.1. On the other hand, it can be noticed that when the fraction of high-quality services provided by malicious peers is near to 45% or greater, SAFE becomes useless and random service selection is more effective. The latter case is quite unrealistic, because providing high-quality services may be expensive and the typical aim of malicious peers is to spare their resources while consuming those of honest peers.
Figure 10 Total number of consumed low-quality services, versus total number of high-quality services (in different percentages: respectively 15%, 30%, and 45%) hosted by malicious providers.

An alternative point of view is illustrated by figure 10, in which the total number of consumed low-quality services is considered, in relation with the total number of high-quality services hosted by malicious providers. Each window shows three graphs, as many as the fractions of malicious advisors (10%, 40%, and 70%) we considered. Points connected by the piecewise curve correspond to a different percentage of malicious providers (from 10% to 90%). Looking at the first window, we notice that, when the fraction of malicious providers exceeds 50%, the trend of the curves changes from increasing to decreasing, depending once again on the immunity of malicious advisors against malicious providers. From the point of view of the attackers, the major damage is inflicted with $N_{MA} = 40\%$ and $N_{MA} = 70\%$ of service consumers.

6 Conclusion

In this paper we have illustrated SAFE, a distributed reputation management system for P2P service architectures. To minimize the risk of corrupted service selection, SAFE allows to relate a peer’s trust in a service provider with reputation values requested to, or published by other service consumers, which play the role of service advisors in that juncture. The service consumer is thus enabled to compute the aggregated reputation of a provider, related to a specific service and to specific QoS constraints.

The effectiveness of SAFE, with respect to different attack models, has been demonstrated by means of simulations. Results have shown that malicious service providers should be numerous and well supported by malicious advisors, to damage a SAFE-based network.

Future work will pursue two main objectives, i.e., the simulative study of SAFE within more dynamic scenarios, as well as the implementation of SAFE as a lightweight Java library.

References


