Reputation-based Service Selection in a Peer-to-Peer Mobile Environment

Michele Amoretti, Maria Chiara Laghi, Alberto Carubelli, Francesco Zanichelli, Gianni Conte
Distributed Systems Group
Information Technology Department
University of Parma
43100 Parma, Italy
amoretti, laghi, carubelli, zanichelli, conte@ce.unipr.it

Abstract

Mobile computing environments may be very critical for traditional QoS management techniques often relying on centralized resource coordinating services, due to the unavoidable high unstability of network and service layers. On the other hand, a peer-to-peer approach for continuous service provisioning to mobile users may maintain locally an updated list of service providers, and allow clients to switch sources depending on the experienced service quality.

In this paper we present a QoS-aware reputation management system for service-oriented P2P networks which aims at improving the quality of service provider selection within purely decentralized mobile environments. The selection process is based on the evaluations provided by other service consumers in terms of a set of application specific QoS parameters.

1. Introduction

Recently, research interest in quality of service (QoS) management architectures has been considerable. Allowing users to specify their requirements in the form of a QoS contract and controlling accordingly system resources, these architecture are becoming of paramount importance in several demanding application domains, and particularly in multimedia content delivery systems. QoS management is usually obtained by combining many resource-oriented functions, namely monitoring and adaptation, admission control as well as resource reservation. Most existing QoS management techniques assume however that the network and service layers offer a relatively stable environment in terms of service quality parameters, *e.g.* connectivity, available throughput and delay jitter. These assumptions cannot generally be made with reference to mobile computing environments, where users may nomadically roam from network to network experiencing frequent changes in service and bandwidth availability. A peer-to-peer (P2P) infrastructure, *i.e.* with no centralized coordinators, may represent a very challenging yet promising approach to the problem of continuous service provisioning to mobile users. As a matter of fact, P2P discovery mechanisms may allow to maintain an up-to-date list of service providers, which can be exploited to switch from one source to another depending on the context as well as on the experienced service quality.

Our research activity in the field of service-oriented P2P architectures has been recently focusing these issues. In this paper we illustrate a relevant outcome of our work, *i.e.* a consistent QoS-aware reputation management system for service-oriented P2P networks which aims at improving the quality of service provider selection within purely decentralized mobile environments. The system, termed Service Advisors For E-business (SAFE), can be considered as a component of our SP2A framework [2]. SAFE relies on a decentralized voting scheme with the novelty of a DHT-based approach to globally 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 QoS parameters, *e.g.* timeliness and accurateness, 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. To avoid that malicious advisor peers, namely those providing deceptive advices, might collect a very high number of transactions, SAFE's DHT stops storing the updates if an excessively fast growth of the number of interactions with the same consumers is observed. Data integrity of information exchanged among peers is guaranteed by means of digital signatures.

The paper is organized as follows. In section 2 we briefly recall the general features of the SP2A framework. In section 3 we introduce the SAFE model by describing how the aggregated reputation of a provider, related to a specific ser-

978-1-4244-2100-8/08/$25.00 ©2008 IEEE

vice, is computed by a peer from its previous experience and from advices provided by other peers, as well as how the trust value in each advisor is maintained. In section 4 we illustrate the results of the simulation of SAFE applied to different scenarios. Section 5 presents a real SP2A deployment where mobile devices equipped with a GUI-based application enable users to search the network for services and to choose among them according to their reputation values and QoS parameters. In section 6 we illustrate the state of the art in distributed reputation management. Finally, in section 7 we provide a discussion of the presented work and illustrate some future work.

2. The SP2A Framework

The Service-oriented Peer-to-Peer Architecture (SP2A) is a framework based on the Peer pattern [1], which defines the basic modules for building service-oriented peers (SOPs).

Figure 1 shows a common SP2A-based system, emphasizing intra- and inter-SOP module interactions.

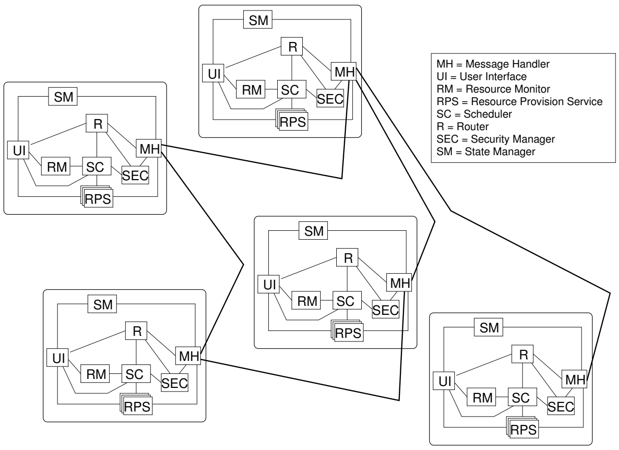


Figure 1. Example of SP2A-based system. For each peer, interactions among internal modules are illustrated.

The User Interface is aware of one or more Resource Provision Services, either local or remote. Moreover, it must be able to query the Resource Monitor and the State Manager, to collect information about the system and to provide it to the user. Finally, the User Interface uses the Router when it publishes its Resource Provision Services or searches for remote ones. The State Manager can modify the state of the peer as a consequence of a user command received from the User Interface, or after the delivery of an event by the Message Handler. For this reason the State Manager is referred by both the User Interface and the Message Handler. The Router computes the destination of messages constructed by the User Interface. Messages can be distributed using the supported transport protocols accessed through one Message Handler. When a remote request for service provision is received, the Scheduler queries the Resource Monitor for available resources. On positive answer, the Scheduler activates the appropriate Resource Provision Service based on the call handler. The Security Manager interacts with the Scheduler, Router, and Message Handler, to modify the behaviour of the peer in the current peergroup.

SP2A mechanisms for service description, sharing and discovery are out of the scope of this paper, which focuses on service selection and delivery. These two aspects involve shared Resource Provision Services and five SOP modules (figure 2).

- The Message Handler delivers a message to the Security Manager, communicating resource request and credentials of the requester (another Peer).

- If the requester's credentials are valid, the Security Manager delivers the resource request to the Scheduler.

- The Scheduler interacts with the Resource Monitor to check the availability of the requested resource.

- If the requested resource is available, the Scheduler invokes its allocation on the corresponding Resource Provision Service.

- If the resource is not available, or by its nature can be provided concurrently by many Peers, the Scheduler starts searching for remote resources, transparently to the user. The Router computes the destination(s) and sends request message(s).

- In the meantime, the Resource Provision Service provides the available local resource to the requester, through the Message Handler.

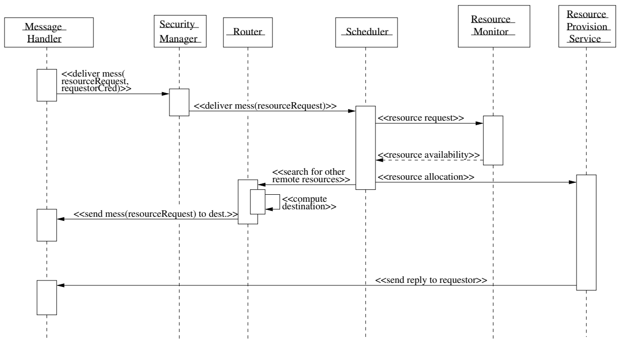

Message Handler
Security Manager
Router
Scheduler
Resource Monitor
Resource Provision Service
<<deliver mess(resourceRequest, requestorCred)>>
<<deliver mess(resourceRequest)>>
<<resource request>>
<<resource availability>>
<<search for other remote resources>>
<<resource allocation>>
<<compute destination>>
<<send mess(resourceRequest) to dest.>>
<<send reply to requestor>>

Figure 2. How a SOP provides its resources.

To complete the scenario, the following section illustrates a component of the SP2A framework which addresses two important steps, service selection and service evaluation, that are executed respectively before and after the service-oriented resource provision procedure.

3. Reputation-based Service Selection

Service Advisors For E-business (SAFE) is a QoS-aware reputation management system, based on distributed voting, with the novelty of a DHT-based approach to share information about advisors in the peer-to-peer network. In the following we describe the overall process in details.

Each time a service provider completes a transaction with another peer, it publishes the updated number of interactions n with that peer, replicated in c different nodes of the DHT. Thus, every service consumer can ask opinions on a particular provider to a set of **advisors**, selected on the basis of their number of transactions with that provider. 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 placed 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.

Supposing a consumer wants to know the reputation of the provider whose identifier is ID_x, the steps it has to follow are summarized in Algorithm 1.

```
search the DHT for $[ID_x \leftrightarrow ID_i, n_{xi}]$ with $i \neq x$
compute the total number of transactions performed by the provider
**if** the total number of transactions is over a threshold defined by the peergroup owner **then**
    **if** if the consumer with the largest number of interactions has performed more than 50% (or another threshold) of the total number of transactions **then**
        exclude it from the list of advisors
    **end if**
    choose at least $N_A$ most experienced advisors
    ask advisors for votes about peer $ID_x$
    weigh received votes
    aggregate votes in a global reputation value $R(x)$
**end if**
```

Algorithm 1: SAFE: basic cycle.

If there are many providers for the same service, for each of them the consumer performs the above steps, and finally chooses the one with highest reputation. After the transaction, the consumer updates its opinion about the service provider, since the consumer itself can be advisor for other peers, and trust parameters associated to the advisors.

If the number of available advisors for a given provider is less than N_A, the provider is considered to be *unknown*. Thus there are three possible situations, for a given service whose provider must be chosen:

- *Worst case* - All providers are unknown, *e.g.* because the dynamicity of the network is too high. In this case the provider is randomly chosen, with uniform probability.
- *Intermediate case* - Some providers are unknown. It is the most frequent case, and it is addressed with the following procedure. The percentage of unknown providers is computed

$$u_{\%} = \frac{N_P^u}{N_P} \tag{1}$$

then the sum of unknown peers' reputation values is estimated as

$$\sum_{N_P^u} R = \frac{\sum_{N_P^k} R}{1 - u_{\%}} u_{\%} \tag{2}$$

and the total reputation value of available providers, both known and unknown is computed

$$\sum_{N_P} R = \sum_{N_P^k} R + \sum_{N_P^u} R \tag{3}$$

A general rule in SAFE is that 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. Moreover, this strategy guarantees a fair distribution of the message traffic in the overlay network.

Thus, to each available provider is assigned a probability P_c of being chosen. If the provider is known, P_c depends on the reputation value which is computed from existing votes. On the other side, to each unknown provider is assigned the same probability

$$P_c^u = \frac{\sum_{N_P^u} R}{N_P^u \sum_{N_P} R} \tag{4}$$

- *Best case* - All providers are known. A probability of being chosen is assigned to each provider, depending on the reputation value which is computed from existing votes:

$$P_c = \frac{R(x)}{\sum_{N_P} R} \tag{5}$$

3.1 Aggregated reputation value

We now illustrate the voting mechanism with more details. Suppose peer ID_i finds k advisors which are able to provide their advices about provider peer ID_j, according to different QoS parameters. For a generic QoS parameter p, the value of trust that peer ID_l (one of the k advisors) assigns to ID_j is

$$t_{lj}^{p} = \frac{{T^{lj}}_{sat}}{{T^{lj}}_{tot}} \tag{6}$$

where T_{sat}^{lj} is the number of satisfactory transactions between peer ID_l (acting as a consumer, in the past) and peer ID_j (provider), and T_{tot}^{lj} is the total number of transactions between the same peers.

The partial reputation value that peer ID_i assigns to peer ID_j, associated to parameter p, as weighted average of the trust values provided by the k advisors is

$$r_{ij}^{p} = \frac{\sum_{l=1}^{k} t_{il}^{A} t_{lj}^{p}}{\sum_{l=1}^{k} t_{il}^{A}} \tag{7}$$

where t_{il}^{A} is peer ID_i's trust value in peer ID_l as an advisor.

The aggregated reputation value that peer ID_i assigns to peer ID_j is the weighted sum of the partial reputation values, considering all the h QoS paramaters:

$$R_i(j) = \sum_{p=1}^{h} w_p r_{ij}^{p} \tag{8}$$

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

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^p is 1 if the transaction is considered satisfactory in relation to parameter p, otherwise it is 0. The aggregated satisfaction value, *i.e.* weighted sum of partial satisfaction values, is

$$S = \sum_{p=1}^{h} w_p S^p \tag{9}$$

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

Parameter α specifies the importance of the last transaction with respect to past history, in order to fix the trust value in peer j as an advisor ($\alpha = 0$ means that only the last transaction is considered, while $\alpha = 1$ means that only past history is considered; all other values of α between 0 and 1 mean that all transactions are considered).

We also define parameter e_α, whose value is $+1$ if the transaction was satisfactory and peer j provided a positive advice, is -1 if the transaction was not satisfactory and peer j provided a positive advice, is 0 in the other cases.

Being $t_{ij}^{A}(n)$ peer i's previous trust value in peer j as an advisor, the updated trust value that peer ID_i assigns to peer ID_j as an advisor is

$$t_{ij}^{A}(n+1) = \alpha t_{ij}^{A}(n) + (1-\alpha)(e_\alpha + t_{ij}^{A}(n)) \tag{10}$$

4. Simulation of Different Scenarios

Using simulations we assessed the performance of SAFE and compared it to a P2P scenario where no reputation management is available. As we considered a P2P network in which services cannot migrate from one node to another, the discovery process has not been simulated, since it does not affect the distribution of resources among peers, unlike file sharing networks.

Each peer belongs to one of the following categories:

- **Honest**: a peer which provides high-quality services with probability between 0.9 and 1, consumes services, and always provides sincere advices.
- **Malicious Provider**: a peer which only provides low-quality services, neither consuming services nor providing advices.
- **Malicious Advisor**: a peer which provides good services, consumes services, and provides deceptive advices.

Asuming N_H honest peers, N_{MP} malicious providers and N_{MA} malicious advisors, the total number of peers in the network is

$$N = N_H + N_{MP} + N_{MA} \tag{11}$$

We simulated a network of $N = 5000$ peers, with many different combinations of (N_H, N_{MP}, N_{MA}). In the following we refer to most significant ones.

The threat model we considered may be called *malicious collective*, since malicious advisors are aware of which peers are malicious providers. The latter may host also good services, in order to deceive honest peers, and do not provide advices, since they assume that malicious advisors' support is enough.

Base settings that apply for most of performed experiments are summarized in Table 1.

4.1 Simple malicious collective

We measured the fraction of high-quality services selected by honest peers, 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. Figure 3 compares three cases of SAFE reputation management (respectively with N_{MA} $=$ $10\%, 40\%, 70\%$ *of the service consumers*)

Network	number of nodes	1000, 5000
	% of malicious provider peers	$10\% - 90\%$
	% of malicious advisors	$10\%, 40\%, 70\%$
Services	number of evaluation paramters	1
Peer behaviour	personal experience weight	$0.15\ [0-1]$
	last transaction weight (in advisor trust computation)	$0.2\ ]0-1]$
	service invocation probability	$0.5\ ]0-1]$
	evaluated service providers	$10\ [1-\infty[$
	minimum number of queried advisors	$5\ [1-\infty[$
	maximum number of queried advisors	$100\ [1-\infty[$
	minimum number of transactions per advisor	$1\ [1-\infty[$
	transaction percentage threshold	$90\%\ ]0-100[$
	suspicious individual transaction percentage	$40\%\ ]0-100[$
	maximum transaction statistics update derivative	$10.0\ [0-\infty[$
	service evaluation satisfaction threshold	$0.6\ ]0-1[$
Simulation	mumber of simulation cycles per experiment	50
	number of experiments over which results are averaged	10

Table 1. Parameter values which have been set in order to evaluate the algorithm by means of simulations.

and the case without reputation management, in which the provider is randomly chosen with uniform distribution of probability.

Comparing the first column with the fourth one, we observe an higher fraction of obtained high-quality services when SAFE is running and the percentage of malicious advisors is 10%, with respect to the random choice solution. In both cases, when N_{MP} increases, the fraction of high-quality service selections decreases, but if the trend is linearly dependent on N_{MP} when providers are randomly selected, it is 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 second and third columns 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 is better than performing random choices.

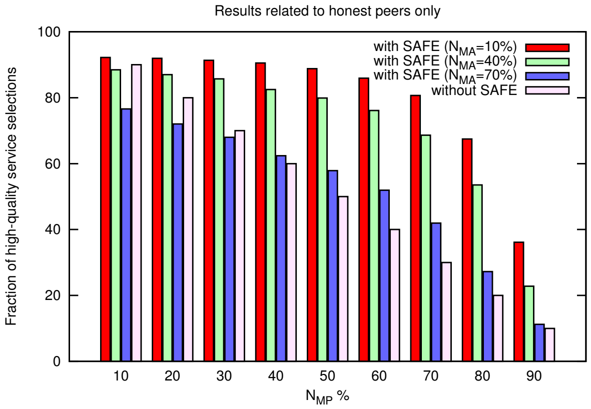


Figure 3. The fraction of high-quality services selected by honest peers in a network where some peers form a malicious collective of service providers and advisors.

4.2 Malicious collective with camouflage

The threat model can be difficult to contrast if some malicious providers host high-quality services together with bad ones (*i.e.* malicious collective *with camouflage*). We simulated three different situations, respectively with 15%, 30%, and 45% high-quality services hosted by malicious providers. Results are shown in Figure 4; each window shows a histogram like that of Figure 3.

Compared to the scenario illustrated in section 4.1, 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 this apparently contradictory behavior is very simple: the total number of available good services is higher in this scenario than in the simpler one illustrated in section 4.1. On the other side, it can be noticed that when the fraction of high-quality services provided by

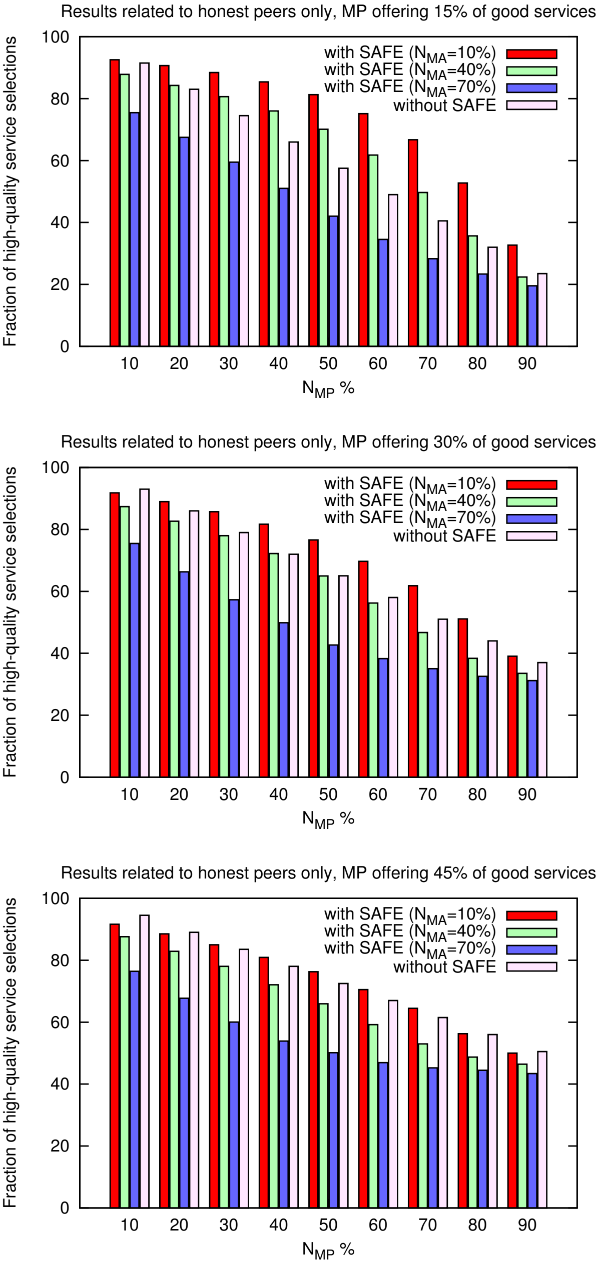


Figure 4. Performance of SAFE when malicious providers also offer high-quality services in order to deceive honest advisors. Considered percentages of malicious advisors are 10%, 40%, and 70%.

malicious peers is near to 45% or greater, SAFE is useless and random service selection is more effective. The latter case is quite unrealistic, indeed, because providing high-quality services is expensive and the aim of malicious peers should be to save their resources while consuming those of honest peers.

In summary, to damage a SAFE-based system, malicious providers should be numerous and well supported by malicious advisors. The camouflage technique is useless, since for little camouflage SAFE is still effective, and high camouflage is too much resource-expensive for malicious peers. Most dangerous collectives are those with malicious providers forming the $40\% - 60\%$ of the peergroup and providing both high-quality and low-quality services (with a $30\% - 70\%$ distribution), and 70% of the remaining peers acting as malicious advisors.

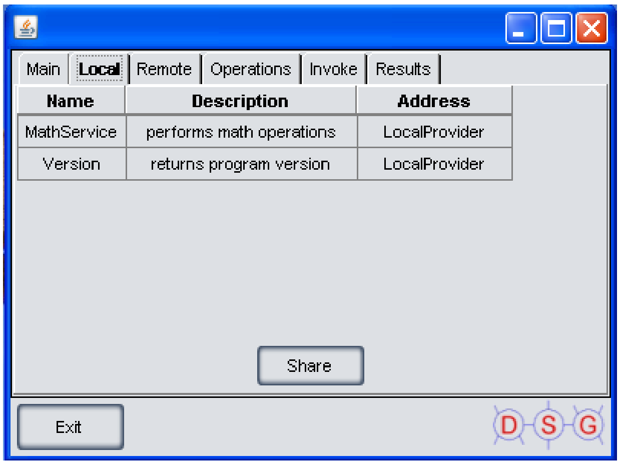
5. System Deployment

SP2A has been implemented as a set of Java interfaces with both J2SE and J2ME class implementations. The API includes four packages: group, rps (*i.e.* resource provision service), security and state. The SP2A middleware currently supports three state-of-the-art technologies: Web Services [11], OWL-S [8] and JXTA [10]. These technologies complement each others: Web Services provide a framework for service description and invocation; OWL-S supplies a service ontology which can be used to improve service descriptions and to enable their orchestration; JXTA operates at the lower level providing P2P protocols. All these technologies are XML-based, and independent from the programming language used to implement them. The SP2A middleware could have been written in C++ rather than Java, but we chose the latter because of its portability (in particular on mobile devices).

Using SP2A, we developed a GUI-based application that allows to join a JXTA-based P2P network in which Web Services can be published, discovered and invoked. The application uses SAFE to compute, for each discovered service, the aggregated reputation of the provider, and the values of the QoS parameters.

The Local panel (in figure 5) shows locally deployed services. A table lists all services and a Share Service button allows to publish their advertisements. Currently, only the J2SE version of the application allows to host services, using the Axis-based JXTA-SOAP component (we plan to port it to J2ME, using kSoap in place of Axis).

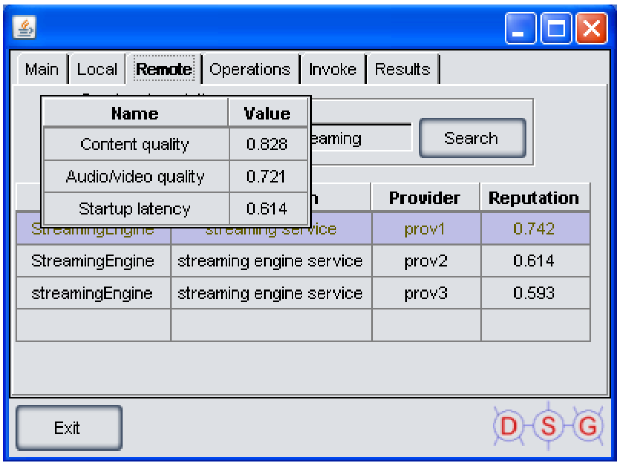
The Remote panel (in figure 6) shows discovered remote services. It is possible to search for services in the P2P network, and to select one of them from the resulting list, in order to see all the operations it offers, which are shown in the Operation tab. The user puts a description of the desired service in the search field, *e.g.* ”streaming”, and all the



Name	Description	Address
MathService	performs math operations	LocalProvider
Version	returns program version	LocalProvider

Figure 5. Local services panel.

matching services are ranked in the table according to the reputation value of the provider. By selecting a row in the table, a window appears with some additional QoS parameters information, used by SAFE to obtain the aggregated reputation value of the provider.



Name	Value
Content quality	0.828
Audio/video quality	0.721
Startup latency	0.614

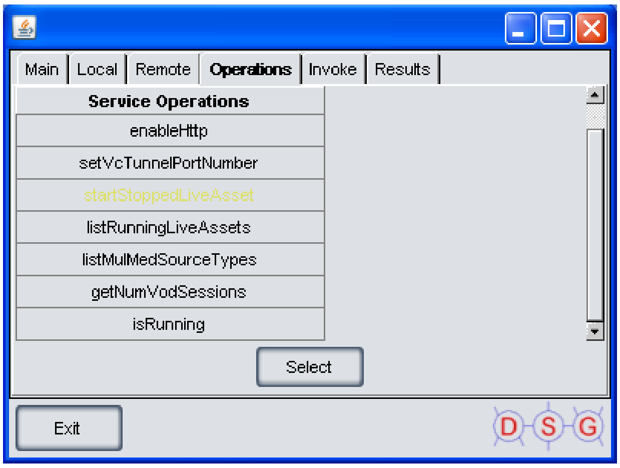
eaming

	n	Provider	Reputation
StreamingEngine	streaming service	prov1	0.742
StreamingEngine	streaming engine service	prov2	0.614
streamingEngine	streaming engine service	prov3	0.593

Figure 6. Panel for reputation-based service selection.

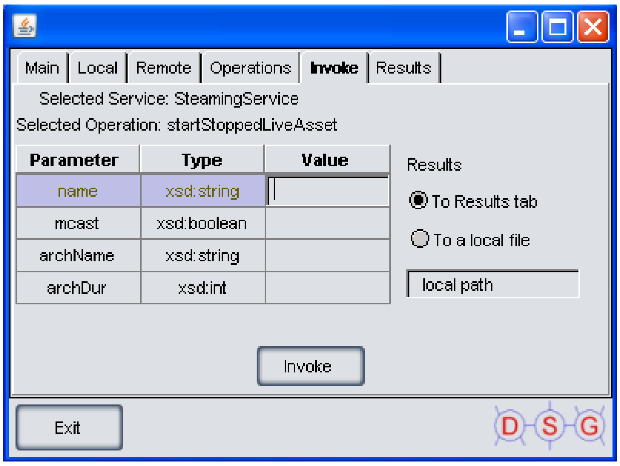
Figure 7 shows all the functionalities provided by the selected Streaming Engine service; the user can choose a particular operation and fill the input parameters table in the Invocation panel.

The invocation panel (in figure 8) is where the user introduces the required parameters for service invocation. If the service returns a result, the user can select where to save it, whether in a file stored locally or in the Result tab (in the last case the corresponding panel is updated with service response).



Service Operations
enableHttp
setVcTunnelPortNumber
startStoppedLiveAsset
listRunningLiveAssets
listMulMedSourceTypes
getNumVodSessions
isRunning

Figure 7. Operation management panel.



Selected Service: SteamingService
Selected Operation: startStoppedLiveAsset

Parameter	Type	Value
name	xsd:string	
mcast	xsd:boolean	
archName	xsd:string	
archDur	xsd:int	

Results
● To Results tab
○ To a local file
local path

Figure 8. Service invocation panel.

6. Related Work

SP2A current prototype can be compared to WSPeer [4], a framework for deploying and invoking Web Services in a peer-to-peer environment, although SP2A is designed to be independent from a particular service description and implementation technology. Moreover, SP2A is concerned with stateful services for Grid environments, *i.e.* services for resource management (deployment, sharing, reservation and use). Another interesting system is OurGrid [3], which is released as a set of components enabling peer-to-peer sharing of computational power. The main difference between OurGrid and SP2A is that the former associates each peer to an istitution, while the latter can run on single user machines. Moreover, SP2A is service-oriented, which allows to address several security issues using standard mechanisms.

In the context of reputation management in P2P networks, researchers have proposed several strategies. Here we briefly describe the most important approaches. Most models refer to content sharing networks, while few works consider service-oriented architectures. With *local evaluation*, after each transaction, the consumer evaluates the quality of the retrieved resource and updates the local reputation value of the provider. In [7] this value is the fraction of satisfactory transactions. This model is very easy to implement, and it does not flood the network with messages, since reputation information are not exchanged among peers. This is also the main drawback of the Local approach, introducing scalability issues. In a *voting* system, the consumer chooses the best provider according to its previous experience and those provided by other peers. Examples of protocols based on this approach are EigenTrust [5] and FuzzyTrust [9]. Choosed 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, thus caching only useful information about resource providers and information providers. The third possible solution is to use *transaction certificates*, generated by involved parties and including the score assigned to the just completed transactions. The resource provider can use the certificate produced by the consumer to demonstrate the quality of its service. When searching for the best provider, networked peers can retrieve and compare transaction certificates in order to assign reputation values to possible providers. Examples of this model are PeerTrust [12] and PET [6]. The advantage of this approach is that nodes are encouraged in providing good resources since they obtain an immediate reward (in fact, reputation can be considered as ”money”). Moreover, malicious collectives are ineffective against this model. The main drawback is that when a peer leaves a group, spontaneously or forcedly, a new affiliation certificate must be generated and shared among other members.

7. Conclusions and Future Work

In this paper we illustrated SAFE, a new component of our SP2A framework which supports service selection considering distributed advices related to QoS values. The aggregated reputation of a provider, related to a specific service, is computed by a peer from its previous experience and from advices provided by other peers. We illustrated the results of the simulation of SAFE applied to different scenarios, involving collectives of malicious service providers and advisors.

For future work, we are interested in comparing SAFE with other voting stategies, by means of the same evaluation parameters we considered in section 4 (we already have some preliminary results related to EigenTrust [5]). From the practical point of view, we plan to use our GUI-based peer application to enable peer-to-peer e-learning communities, with context-aware service provision.

References

[1] M. Amoretti, M. Reggiani, F. Zanichelli, and G. Conte. Peer: an Architectural Pattern Enabling Resource Sharing in Virtual Organizations. In *The 12th Pattern Languages of Programs (PLoP) 2005, Monticello, Illinois, USA*, September 2005.

[2] M. Amoretti, F. Zanichelli, and G. Conte. SP2A: a Service-oriented Framework for P2P-based Grids. In *3rd International Workshop on Middleware for Grid Computing, Co-located with Middleware 2005.*, November 2005.

[3] N. Andrade, L. Costa, G. Germóglio, and W. Cirne. Peer-to-peer grid computing with the OurGrid Community. In *Proceedings of the 23rd Brazilian Symposium on Computer Networks*, 2005.

[4] A. Harrison and I. Taylor. Dynamic Web Service Deployment Using WSPeer. In *The 13th Mardi Gras Conference, Baton Rouge, Louisiana.*, February 2005.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *12th International Conference on World Wide Web*, May 2003.

[6] L. Liang and L. Shi. Enforcing Cooperative Resource Sharing in Untrusted P2P Computing Environments. *Mobile Networks and Applications*, 10(6), December 2005.

[7] S. Marti and H. Garcia-Molina. Limited Reputation Sharing in P2P Systems. In *5th ACM Conference on Electronic Commerce*, May 2004.

[8] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. Semantic Matching of Web Services Capabilities. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, pages 333–347, 2002.

[9] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted P2P Transactions with Fuzzy Reputation Aggregation. *IEEE Internet Computing*, November 2005.

[10] B. Traversat, M. Abdelaziz, and E. Pouyoul. A Loosely-Consistent DHT Rendezvous Walker. *Project JXTA*, March 2003.

[11] W3C. W3C Web Services Activity home page. http://www.w3.org/2002/ws/, 2002.

[12] L. Xiong and L. Liu. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7), July 2004.