An Aggregation Scheme for the Optimisation of Service Search in Peer-to-Peer Overlays

Adriano Fiorese*†
*UDESC, University of the State of Santa Catarina
CCT, Department of Computer Science - DCC
89223-100, Joinville-SC, Brazil
e-mail:fiorese@joinville.udesc.br

Paulo Simões†, and Fernando Boavida†
†CISUC, University of Coimbra
Pólo II, Department of Informatics Engineering - DEI
3030-290 Coimbra, Portugal
e-mail: {fiorese,psimoes,boavida}@dei.uc.pt

***Abstract*—This paper presents a mechanism to improve the efficiency of service searching in large-scale multi-domain environments composed of service providers organized in a P2P Service Overlay Network (SON). This mechanism relies on a P2P overlay-tier whose purpose is to aggregate the services and service components maintained by service providers in the P2P SON. We name this mechanism Aggregation Service (AgS). It comprises the service and service components publishing and allows the separation between the service providers and the search schema. The average search path length was used as metric for the AgS assessment through simulation. The simulation results clearly show an improvement in search operations when the proposed Aggregation Service is used, when compared to searches performed without it.**

***Index Terms*—Services management, P2P, Service aggregation**

I. Introduction

Services and service components are becoming the basic elements in interactions among service providers in the Future Internet. These services and service components might include content (e.g. finding vendors offering a specific movie; finding dubbing or subtitling services for that same movie), connectivity (e.g. interconnection links which satisfy QoS and security requirements between the consumer and the origin of the contents) and complementary services (electronic billing and/or payment systems; multi-session controllers for video-conferencing sessions, etc.). These services might be dispersed among providers on the big cloud the Internet currently represents. Such services and service components need to be searched, grouped, composed, provisioned, etc., in order to offer the final users a consolidated new product (service).

A Service Overlay Network (SON) [1], [2] enhances the service provider ability to make their services or component services available. SON acts as an infrastructure where services are published/offered and to which users access in order to select and use these services. This infrastructure should be formed and supported by the service providers committed in creating a collaborative/competitive environment.

The use of P2P technology to create and manage a SON is beneficial in order to achieve a self-organizing overlay level as well as to share the maintenance costs among the providers that form the SON. Another advantage is that at the same time the service providers organize themselves into a P2P SON they also can keep their sensitive business information hidden from the way the services are dispersed and offered.

Despite the P2P SON formation, services still need to be searched. The intrinsic P2P searching mechanisms are well suited for this task. However, in a foreseen scenario for Future Internet (taking into account services negotiated in a huge market and even more virtualized environments such as Clouds with IaaS and SaaS [3], [4]), where the number of services and service components can easily scale to considerable numbers, more specialized and efficient search mechanisms are required.

In order to face the problem of optimizing service searching in a P2P SON, we propose an Aggregation Service (AgS). AgS is a second P2P overlay-tier that executes on top of the P2P SON, aggregating the published services and thus making the search faster. In this respect, some questions arise:

- Can the use of an Aggregation Service improve current search P2P mechanisms?
- Can the replication/caching of the search results boost the searching process? What is the effect of replication in environments with and without the Aggregation Service?
- What is the influence of services distribution in environments with and without Aggregation Service?

In order to answer the stated questions, the following approach was considered: 1) Implementation of the P2P SON Aggregation Service in a simulated environment; 2) Analysis of the simulation results taking into consideration the average search path length (APL) as the measurement metric.

Having in mind the stated goal and approach, this paper is organized as follows. Section II discusses related work. Section III describes the proposed AgS. Subsequently, Section IV presents and discusses the simulation results after describing in detail the simulated scenarios. Finally, Section V summarizes the proposal and discusses further work.

II. Related Work

This work crosses three particular areas in the field of network and distributed systems: 1) Network and Services Management; 2) Peer-to-Peer Networking; and 3) Service Overlay Networks. The inter-relations between these subareas shape the efforts of developing an efficient, scalable, and robust infrastructure to cope with the services market. Fig. 1 shows these subareas and their intersection, where this work is logically located. In the following subsections we are going to discuss previous work on each one of the subareas, which are related with the problem of search optimization in distributed large-scale environments.

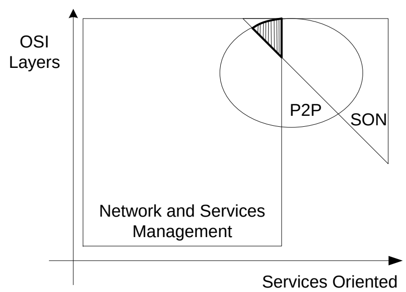

Fig. 1. Location of this work taking into account three subareas of network and distributed systems

A. Network and Services Management

This area practically was born with the advent of computer networks. Therefore, it is not a new research theme nor is it completely focused on the search optimization issue. However, many pieces of work provide indirect contributions addressing this problem.

A contribution mixing network management and search improvement was made in 1997 by Soukouti and Holberg [5]. They suggested integrating management protocols with the distributed objects searching functionalities provided by the CORBA technology. However, CORBA's naming service requires a high overhead on its configuration in order to adequately support multi-domain environments, which means this solution was not very dynamic or scalable.

Management by Policies is also used to enhance services management [6], [7]. Work in this area addresses managing performance service level agreements between internal service providers in a network through the enforcement of policy levels. However, these approaches depend on a series of agreements, adaptation and trust to be realized in cross-domain environments that suggests the use of an appropriate Service Overlay Network to take care of this.

Management by Delegation (MbD) [8] is another classical technique for network management. Besides that, the rational can also be used for services management. Since services in the foreseen scenario can be provided using components, in a service composition chain some specific tasks can be delegated by a heavily loaded component to another less loaded one. An example could be the billing service, where a video service component can delegate charging tasks to the billing service belonging to the same service provider.

Searching is also a challenge faced by Mobile Agents (MA) [9], [10] approaches. The challenge exists even when the searching is for other MA platforms [11] and alternative paths for the agent's itinerary; or when the searching is for services [12] in a large scale environment. The latter addresses the use of mobile agents to discover semantic web services. However, the itinerary taken by mobile agents does not take into account the possible fail of the web services hosts. Indeed, this is a drawback of many pieces of work in the Web Services searching area.

Currently, web services are the most developed approach to network and services management [13]. They seem to be the most popular solution for offering service interfaces and service composition [14]–[17], on which the Service Oriented Architecture (SOA) lays on [18]. Taking this into account, the huge interest in the development of mechanisms that overpass the single centralized Universal Description, Discovery and Integration (UDDI) searching mechanisms is understandable. Our paper fits in this category of work, using a decentralized approach based on P2P, which is supported by a particular SON composed amongst the service and service component providers.

B. Peer-to-Peer

This is a more recent research area and it is currently well explored as first-tier application as well as support application.

As first-tier application the development of P2P networks and their optimization is the most common research subject. P2P networks can be classified according to different criteria. In one of them they are generally classified into two categories: 1) Unstructured and 2) Structured. These terms relate to the topology of the P2P overlay network. When the topology is tightly controlled and content is placed at specific peers instead of randomly choosen, then the P2P network is structured. Generally, this is accomplished using a Distributed Hashing Table (DHT) as the core of the P2P network. Some examples of structured P2P overlay networks are: CHORD [19], Content Addressable Network (CAN) [20], Chamaleon [21]. If the topology is not tightly coupled, which means the peers join the network with some loose rules, then this network is classified as unstructured. In this kind of P2P networks there is not a coupling between topology and data items' location. Instead, peers organize a random graph in a flat or hierarchical manner. Generally, in this kind of P2P network, peers use some kind of flooding to send queries (searching) with a limited scope and through their neighbors. Some examples of seminal unstructured P2P overlay networks are: Gnutella [22], FreeNet [23]. Reference [24] presents an extended discussion and comparison about structured and unstructured P2P overlay networks.

On the strand of support-tier, P2P overlay networks are significantly used. The relation with the goal of this paper is its use as tool for network and services management, especially regarding searching. In this case, works [25]–[27] propose cooperation among network managers by mutual P2P-based exchange of messages, and also by allowing the management tasks to be carried out by groups of peers.

Nowadays, web services seem to be the most popular solution for offering service interfaces. Several authors have addressed the use of P2P for searching web services. Schmidt and Parashar [28] for instance, propose a structured P2P overlay, based on CHORD, to improve the search of web services by using multiple keywords. The introduction of semantics on the search is well studied [29]–[31]. All these proposals claim that the P2P approach has some advantages for the service discovery process, when compared with centralized approaches such as UDDI. These advantages include scalability; enhanced failure tolerance (by eliminating the single point of failure of the centralized approaches); and efficiency (by reducing the overhead of updates and replication operations that can be found in centralized approaches).

C. Service Overlay Networks

According to [1], a Service Overlay Network (SON) is a network composed of interconnected nodes, whose generic purpose is to provide the required Quality of Service (QoS) to applications that execute on those nodes. The same authors establish a difference between SON and P2P overlay network claiming that the purpose of the latter is related with providing efficient searching and retrieval.

The problem of bandwidth provisioning in a SON composed of nodes that lease links from different link providers is studied in [32].

We advocate a P2P overlay network can also provide QoS services when the participants are in a consortium of service providers that establish well-defined SLAs to regulate the contribution of each participant to the network. In this sense, a platform called ALASA is presented in [2]. It uses a structured P2P overlay network over the Internet to describe, discover, compose, and reputation of services.

Lavinal et. al. [33] also uses P2P as support for the SON architecture. In that piece of work the authors also address the discovery of services, although they consider QoS aspects in their approach whilst we take into account performance aspects.

III. AGGREGATION SERVICE

The Aggregation Service (AgS) we propose is an unstructured P2P overlay consisting of peers that represent potential providers offering their services. The AgS executes on top the P2P SON composed by these potential providers.

Each service offer may be represented by more than one peer, which allows for some redundancy. The peers that form the AgS P2P overlay are called *aggregation peers*. The purpose of the aggregation peers is to aggregate the offerings of services and service components in an overlay-tier to facilitate the searching.

The AgS also includes another class of peers: the *SON peers*. These peers belong to the service providers that offer service/components. Unlike the aggregation peers, SON peers participate in the AgS by making the services administration and management interfaces indirectly available to external entities (such as service aggregators), likely located in different domains. Such management interfaces are used in the assembly and lifetime management of the composed services. SON peers make these interfaces available by publishing a service offer at several aggregation peers in the P2P AgS overlay. These aggregation peers may be located in the same domain as the SON peers or, in some cases, in different domains. Searching for a service, using the AgS framework, therefore results in a set of references for SON peers that offer an interface to services matching the search criteria. This preserves the internal details of the service, since the external entity is only granted with a mediated access (by means of the SON peer), which may hide sensitive information and filter undesired operations.

The AgS can be used as a non-structured solution since it can be deployed within the current infrastructure of the service providers without increasing costs. In addition, the two-tiered AgS architecture allows the publishing and searching functionalities (provided by the AgS) to be separated from the management functions to be carried out by the management peers. Thus, the service managers of the service provider can restrict the sensitive configuration of its services (e.g. the existing management services, topologies, etc...) by only making available a previously selected set of interfaces for services and service components to a third party. The design of the AgS depicted in Fig. 2 consists of a P2P overlay using a ring topology, although other P2P topologies can be easily applied. Fig. 2 also shows the SON peers belonging to different administrative domains, which announce their services and service components to the aggregation peers.

Taking into account the way of working of the AgS, the P2P SON underlay can be composed of an unstructured or structured overlay. Since the underlay SON is a way to organize the participant service providers, the AgS takes up the role of booster for the search and also the role of contact interface for the users (e.g. providers of new composed services and final users). Even if the SON underlay was not composed through P2P bindings, the AgS can still be used. However, one of the goals of this paper is to demonstrate the utilization of the AgS is advantageous when compared with the native searching mechanisms. For this reason, a P2P SON was adopted in our assessment.

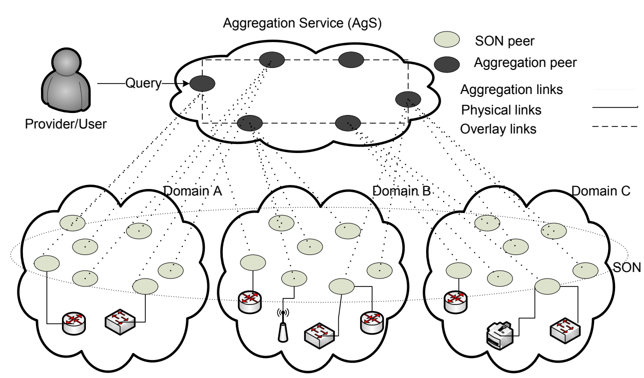

Fig. 2. The Aggregation Service Architecture

A. AgS Operations

The activities of the AgS are based on a number of operations. Table I presents its key operations and the corresponding messages exchanged among peers.

IV. Validation Framework

In order to assess the AgS behavior and answer the aforementioned research questions, we conducted a simulation study to measure the average path length (APL) necessary to locate the SON peer that makes a particular service available.

The simulation involved 1000 aggregation peers, which kept the services that were made available and published by 10,000 SON peers spread over 10 different domains. For the sake of simplicity, a particular SON peer can only offer, at most, seven services or service components randomly chosen (using a uniform distribution) from the service set S={S1,S2,S3,S4,S5,S6,S7}.

Each SON peer can only publish its service subset on, at most, 10 distinct, randomly chosen, aggregation peers (also following a uniform distribution). Nevertheless, it is possible that more than one SON peer can offer the same services subset at the same domain and publish it on the same aggregation peer. In the interests of simplification, the search concludes with the first match (although AgS also supports the return of all matches).

Each execution simulates 50 hours of work. Each simulation used the same scenario and a different number of search operations were simulated, which ranged from 100 to 1000 operations. Every operation (e.g. joining, leaving, publishing, and searching) is specified in time. Fig. 3 displays an excerpt of the operations file that feeds the simulation. There we can see line by line, the name (ID) of the peer, followed by the time in which the operation should execute (in minutes but it can be configured in seconds as well) and finally the operation to be executed. The exception is the Search service operation that needs the service name as its actual parameter.

In order to simulate scenarios as real as possible, the simulations run on "well-behaved" scenarios that are also described as consistent. This means the execution of the operations follow a temporal sequence. As depicted in Fig. 3, the Publish operation only executes when every aggregation peer on the SON peer's publication list has already carried out the Join operation. The same happens in the case of the Leave operations. The Query operations however do not follow this pattern, since for a Query operation to be started it is enough its peer is alive. Thus, according to Fig. 3 for example, the search operation conducted by A628 has as result M86, if any of the aggregation peers (A593, A67, A222, A797, A134, A433) had not yet left the AgS P2P overlay. In this case, there would be only a QueryReply message to A628 if some other SON peer belonging to Domain 6 had made its publication of the S4 service to another aggregation peer that had not yet left the overlay.

A433 497m join
A134 721m join
A593 777m join
A67 1402m join
A797 2734m join
A222 2813m join
M86 2921m SONNodesApplication:publish
A134 1342m leave
A433 1361m leave
A67 2422m leave
A797 2770m leave
A593 2818m leave
A222 2861m leave
A628 2937m searchService S7@Domain6

Fig. 3. Fragment of the simulation's operations file

In order to answer the second question stated in Sec. I, the replication of the search results was also taken into account. Thus, in the simulations where replication was active, when the queryMessage reaches the aggregation peer that keeps the desired service or service component, then a queryReplyMessage containing that information is sent to the origin aggregation peer that stores the information in its local cache. In a future query for the same service or service component started by aggregation peers located before the mentioned one in the ring, then the search will get fewer hops.

Two sets of simulations were executed comprising two scenarios involving the P2P SON: 1) with the AgS and 2) without the AgS. For each scenario two environments had simulations executed: 1) results replicated, and 2) results not replicated. Fig. 4 summarizes the validation framework.

TABLE I
AgS Table of Operations and Messages

Operation	***Goal***	***Executor***	***Message sent***
Join	Form the Aggregation Service.	aggregation peer	JoinMessage sent by the requesting peer to its successor and predecessor in the overlay.
Leave	Leave the Aggregation Service (in a normal way).	aggregation peer	LeaveMessage sent by the requesting peer to its successor and predecessor in the overlay.
Query	Look for a peer that provides a particular service/service component in a specific domain.	aggregation peer	QueryMessage sent by the requesting peer to its successor in the overlay ring in a clockwise manner. The message is forwarded clockwise until it arrives at its goal or until the message reaches the requesting peer. When the desired information is found, a QueryReply message containing it, is then created. This latter message is directly transmitted to the requesting peer of the Query´s operation.
Publish	Make the services to be searched available.	SON peer	PublishMessage sent by the SON peer to its aggregation peer(s), which makes the service(s) public. A PublishMessage is sent to each aggregation peer.

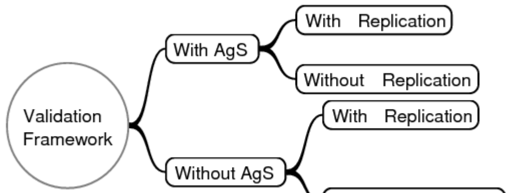

Fig. 4. Validation framework schema

The validation framework reckons on particular parameters regarding the network and transport layers that support the AgS overlay. These parameters are the bandwidth for upload and download of 100 and 200 Mbps, respectively. Also, a static latency model for packet transmissions was used with value of 10 ms. At the transport layer the AgS uses UDP for datagrams.

A. The Simulator

The PeerFactSim.KOM [34] simulator was used in the simulations. This simulator is based on discrete events, which explains why the operations file is used. It is particularly designed to simulate large-scale P2P networks.

B. Results

Some considerations must be done before presenting and discussing the results. As aforementioned (see Table I), the Query operation generates an initial message that is successively forwarded until it finds the desired information or the original requesting aggregation peer. Each transmission/forwarding represents a hop on the message path. Thus, the average path length will be the ratio between the number of hops and the number of search messages or search operations successfully or unsuccessfully accomplished, depending on the intended analysis to be done. Eq. 1 depicts the formula for the average path length.

It is worth mentioning that Query operations and Query Messages are different things. Operations send messages and receive replies. However, for the Query operations this assumption is not 100% true. When the aggregation peer that starts the Query operation finds the desired service or service component in its local cache, a new Query message is not necessary and by consequence a reply is never received. In our analysis, as Eq. 1 shows, we use the number of received replies to the sent query messages, in order to actually represent the APL as the average path travelled by a Query Message until it reaches its destination.

$$APL = \frac{\sum_{hops=0}^{n_{infoFound}} hops}{QueryReplyMessages} \quad (1)$$

In this sense, the number of executed operations is different between the experiments with and without AgS. Fig. 5 depicts this difference. This discrepancy is explained by the absence of necessity of Publish operations in the environment without AgS.

Fig. 5 shows the relation between the total number of simulated operations and the number of query operations. In inset (a) of Fig. 5, lines representing values with replication (1,3,5) are identical to the ones that represent values without replication (2,4,6). The identical pairs are (1,2), (3,4), and (5,6). The same happens for inset (b) in Fig. 5. This means the replication has not effect in the effectiveness of the experiments, despite its existent influence on the performance (in terms of APL) of the searches.

An important point that should be highlighted about this figure, is that the query hit ratio is almost 100% (the fails are searches started for inexistent data at the time of execution), i.e., almost every searched service or service component is found independently of the existence of AgS. In spite of this, it does not mean the AgS does not make any difference. The AgS advantage is the improvement of the performance of the searches, as we will see in the APL analysis, below.

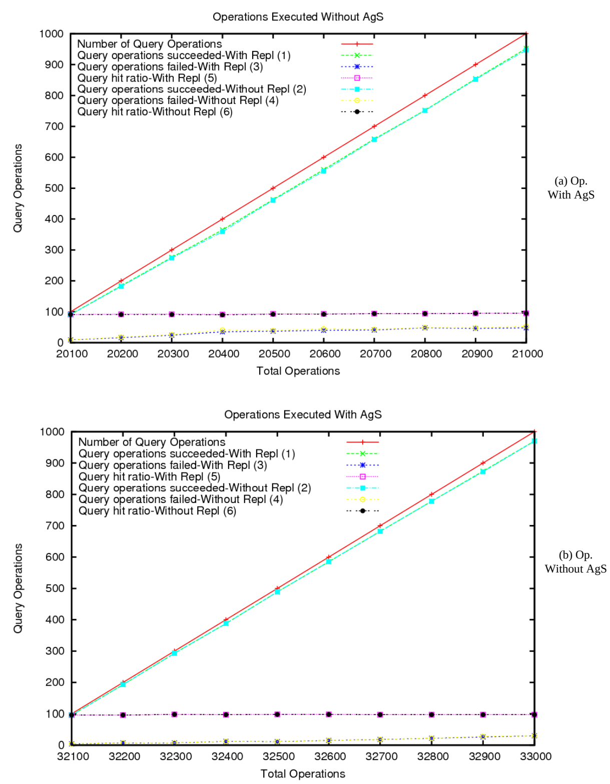

Fig. 5. Compared operations executed in both scenarios

The APL analysis takes into account the number of hops in successful Query Messages, i.e., the number of aggregation peers that are counted in between the query requesting peer and the providing peer by a traveler Query Message. Our results rely on a confidence interval (C.I.) of 95% for the mean APL value. This C.I. is calculated taking into account the APL population standard deviation is unknown, which means the use of the Student table with $(n-1)$-degrees of freedom. The value of n is 10 and 7 depending on the experiments, which are the number of experiments varying the number of Query operations, and the number of services, respectively.

Table II provides the values of APL for the experiments with and without AgS. The columns C.I. represent values for the margin of error calculated for the confidence interval of 95%. The first column named Op Query represent in descending order the number of query operations that were executed for each scenario and environment. The columns named Repl represent the values of APL in the environment with replication of the search results. On the other hand, the columns named Repl represent the values for the environment where replication was not used.

Fig. 6 depicts the APL for the aforementioned scenarios and environments. In this figure, it is possible to notice two representative sets of curves. On top are the ones that represent the scenario without AgS, in which the higher curve depicts the environment without the replication of the search results. On the bottom there are the two curves for the scenario with AgS, and as expected, the lower curve represents the environment with replication. All curves are drawn with the margin of error considering a 95% confidence interval.

Fig. 6 is quite revealing in several ways. First, the replication of search results is worthwhile for repeated queries for the same service/service components. Taking the arithmetic mean of the APL from Table II as basis, it is possible to figure out that the APL is approximately 14,39% lower when the replication of search results is used in the scenario without aggregation service. The APL is also low when the replication is used in the scenario with AgS. Although, the difference is only approximately 3,73%.

These differences are explained by the services and service components concentration. In the scenario without the AgS the replication is more relevant since the services are not concentrated at the peers' local cache. On the other hand, in the scenario with AgS the Publish operation feeds the peers local cache concentrating the services on a lower number of peers and consequently making the queries less dependent on replication.

Furthermore, this finding helps the assessment of our second research question. In spite of the positive answer, it is worth to mention that the replication of the search results and even the AgS can face trusting problems. The use of a pair (service key, reference of the SON peer offering that service key) in the base of the searches can minimize this problem. However, when a malicious provider wants to deny services from competitors it can modify the SON peers' references in its local cache to its own advantage. An entire solution for this problem depends on trust and repudiation solutions and they are out of the scope of this work.

Fig. 6 is also revealing with regard to the use of the proposed AgS. Indeed, the AgS dramatically reduces the search APL. A comparison between the arithmetic mean values for both scenarios, taking into account the environment where the replication was not used, reveals the APL is approximately 64,67% lower when using the AgS. When the same comparison is made for both scenarios taking into account the environment where replication was used, then the difference is approximately 60,26%. These drastic reductions in the APL values show the effect of using AgS to improve the searching of services and service components in a P2P SON. This positively answers our first research question.

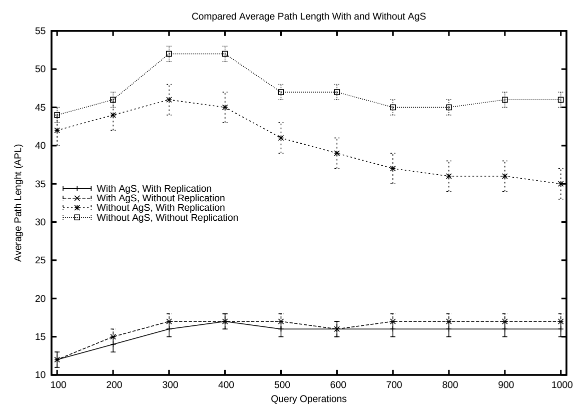

Fig. 6. Compared Average Path Length

The behavior of the APL can be seen in Fig. 7. The curve adjustment by polynomial regression (the standard error and correlation coefficient are shown in the graph) reveals 4 different equations that describe the behavior of the APL where x is the number of query operations and y is the

TABLE II
AVERAGE PATH LENGTH VALUES

	With AgS				**Without AgS**			
Op. Query	***Repl***	***C.I.(±)***	***¬Repl***	***C.I.(±)***	***Repl***	***C.I.(±)***	***¬Repl***	***C.I.(±)***
1000	16,9	1,083836	17,85	1,135426	42,08	2,715183	44,49	1,993238
900	16,48	1,083836	17,3	1,135426	44,16	2,715183	46,55	1,993238
800	16,93	1,083836	17,7	1,135426	46,09	2,715183	52,39	1,993238
700	16,84	1,083836	17,14	1,135426	45,12	2,715183	52,22	1,993238
600	16,13	1,083836	16,63	1,135426	41,01	2,715183	47,07	1,993238
500	16,68	1,083836	17,18	1,135426	39,08	2,715183	47,19	1,993238
400	17,12	1,083836	17,5	1,135426	37,92	2,715183	45,19	1,993238
300	16,92	1,083836	17,3	1,135426	36,83	2,715183	45,44	1,993238
200	14,25	1,083836	15,88	1,135426	36,59	2,715183	46	1,993238
100	12,47	1,083836	12,47	1,135426	35,65	2,715183	46,01	1,993238

APL. Table III shows the approximate (due to lack of space) coefficients for the curves presented in Fig. 7. Considering the high correlation coefficient and relatively low adjustment error (at least for the scenario with AgS), we assume the remaining behavior can be predicted using these polynomials, which allows considering Fig. 7 as the probability density function for the APL behavior in our simulations.

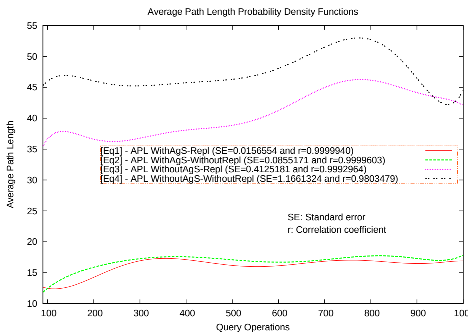

Fig. 7. PDF Average Path Length distributions in both scenarios in both environments

These findings were revealed using scenarios where the services a peer can publish in the AgS or keep in its local cache were randomly chosen from a set of 7 services taken from the minimum of 1 to the maximum of 7. Nevertheless, what happens when the distribution of services changes in the AgS and in the P2P SON? This indeed is our third research question. Thus, considering the set of 1000 search operations as the most significant set of search operations, we have simulated the changing distribution of the services. Fig. 8 shows the results. There, the x axe represents the maximum number of services a peer can offer (published in AgS or kept locally). All simulations used 1000 search operations. The y axe represents the APL. As expected, the higher the number of services a peer can offer, the lower will be the APL. It is worth to mention that peers still are constrained to publish their offer to at most 10 aggregation peers.

This preliminary result means that the more services a peer can publish the better in terms of APL; therefore, more easily these services can be found. This helps to explain the low APL with AgS in Fig. 6. Since peers publish at most 7 services in a reduced set of aggregation peers, the random spread makes them available in few hops and that means low average path length.

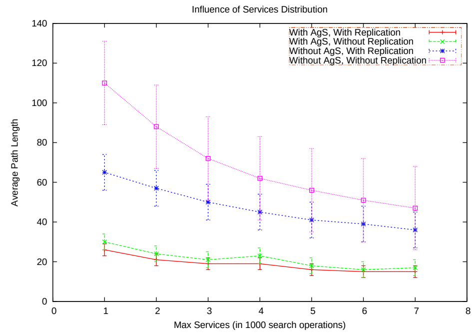

Fig. 8. Influence of services distribution on the APL

V. Conclusion

This paper addressed the optimization of service and service components searching in a P2P SON. To accomplish that, an Aggregation Service (AgS) was proposed and evaluated through simulation using the average search path length (APL) as the assessment metric. The AgS is based on the publication of the service offerings in a second-tier P2P overlay, which optimizes the searches. The APL is the ratio between the number of hops a Query Message experiences from the search requesting peer until the peer that keeps the searched service reference, and the number of positive returned Query Reply messages.

The results show that the AgS reduces the APL in 64,67%, and in 60,26% when both scenarios (with and without AgS) are compared, taking into account when replication of the search results is not used and when it is used, respectively. The replication of the search results also helps to diminish APL, though it is more relevant when used for searching on the P2P SON without AgS. Results also show the inverse relation between the peers' service offering capacity and APL. In this case, the higher the number of offered (published) services the lower the APL.

However, further studies are necessary. The maintenance of the AgS data consistency is an open issue. This improvement is important to prevent the occurrence of query results that do not actually represent active services. A possible approach is to send a message (e.g., ACK message) to the P2P SON peer found by the query operation in order to verify the component service availability. This approach may prevent the overhead of the P2P SON peers' pooling monitoring messages.

TABLE III
Approximate Coefficients for Polynomial Functions on Fig. 7

	≈**Coefficients**								
	a	***b***	***c***	***d***	***e***	***f***	***g***	***h***	***i***
Eq1	1.909E+001	-1.215E-001	4.535E-004	2.481E-006	-1.923E-008	4.904E-011	-6.109E-014	3.770E-017	-9.234E-021
Eq2	1.800E-001	2.240E-001	-1.480E-003	5.911E-006	-1.399E-008	1.874E-011	-1.300E-014	3.618E-018	
Eq3	-2.053E+001	1.445E+000	-1.407E-002	7.039E-005	-2.009E-007	3.400E-010	-3.361E-013	1.790E-016	-3.964E-020
Eq4	1.701E+001	6.773E-001	-5.892E-003	2.551E-005	-6.086E-008	8.112E-011	-5.626E-014	1.573E-017	

Planned future work also encompasses the fine-tuning of the parameters that control the AgS (e.g. the max number of aggregation nodes a SON peer can publish on; the number of network domains; and the maximum number of services a peer can publish) in order to achieve the best proportion between APL and minimal number of aggregation peers in the AgS. These results can help the planning of the P2P SON by the interested service providers in order to keep costs under control.

ACKNOWLEDGMENT

This work was partially funded by FCT under scholarship contract SFRH/BD/45683/2008.

REFERENCES

[1] C. Tran and Z. Dziong, "Service overlay network capacity adaptation for profit maximization," *IEEE Transactions on Network and Service Management*, vol. 7, no. 2, pp. 72–82, Jun. 2010.

[2] S. Zhou, M. Hogan, S. Ardon, M. Portman, T. Hu, K. Wongrujira, and A. Seneviratne, "ALASA: when service overlay networks meet peer-to-peer networks," in *Communications, 2005 Asia-Pacific Conference on*, Perth, WA, USA, 2005, pp. 1053–1057.

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, Austin, TX, USA, Dec. 2008, pp. 1–10.

[4] L. Youseff, M. Butrico, and D. D. Silva, "Toward a unified ontology of cloud computing," in *Grid Computing Environments Workshop, 2008. GCE '08*, 2008, pp. 1–10.

[5] N. Soukouti and U. Hollberg, "Joint inter domain management: CORBA, CMIP and SNMP," in *Proceedings of the fifth IFIP/IEEE international symposium on Integrated network management: Integrated management in a virtual world*. London, UK: Aurel A. Lazar, Roberto Saracco, Rolf Stadler, 1997, pp. 153–164.

[6] L. Lymberopoulos, E. Lupu, and M. Sloman, "An adaptive police-based framework for network services management," *Journal of Network and Systems Management*, vol. 11, no. 3, pp. 277–303, Sep. 2003.

[7] D. C. Verma, "Simplifying network administration using policy-based management," *IEEE Network*, vol. 16, no. 2, pp. 20–16, Apr. 2002, IEEE Communications Society.

[8] G. Goldszmidt and Y. Yemini, "Distributed management by delegation," in *Proceedings of the 15th International Conference on Distributed Computing Systems*. IEEE Computer Society, 1995, pp. 326–333.

[9] H. H. To, S. Krishnaswamy, and B. Srinivasan, "Mobile agents for network management: when and when not!" in *Proceedings of the 2005 ACM symposium on Applied computing (SAC '05)*. Santa Fe, New Mexico, USA: ACM New York, NY, USA, 2005, pp. 47–53, sponsor: SIGAPP: ACM Special Interest Group on Applied Computing.

[10] M. Al-Kasassbeh and M. Adda, "Analysis of mobile agents in network fault management," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 699–711, 2008, publisher: Elsevier.

[11] D. Gavalas, G. E. Tsekouras, and C. Anagnostopoulos, "A mobile agent platform for distributed network and systems management," *Journal of Systems and Software*, vol. 82, no. 2, pp. 355–371, Feb. 2009.

[12] X. quiang Wang and S. lin Song, "Mobile agent-based semantic web service discovery model," *Computer Engineering and Design*, vol. 30, no. 6, pp. 1328–1331, 2009.

[13] R. L. Vianna, M. J. B. Almeida, L. M. R. Tarouco, and L. Z. Granville, "Investigating web services composition applied to network management," in *Proceedings of IEEE International Conference on Web Services*. Chicago, USA: IEEE Computer Society, 2006, pp. 531–540.

[14] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson, *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging, and more*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2005.

[15] C. Peltz, "Web services orchestration and choreography," *Computer*, vol. 36, no. 10, pp. 46–52, Oct. 2003.

[16] X. Fu, T. Bultan, and J. Su, "Analysis of interacting BPEL web services," in *Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM, 2004, pp. 621–630.

[17] S. Dustdar and W. Schreiner, "A survey on web services composition," *International Journal of Web and Grid Services*, vol. 1, no. 1, pp. 1–30, Aug. 2005.

[18] L. Auer, C. Strauss, N. Kryvinska, and P. Zinterhof, "SOA as an effective tool for the flexible management of increased service heterogeneity in converged enterprise networks," in *Proceedings of the 2008 International Conference on Complex, Intelligent and Software Intensive Systems*. Barcelona, Spain: IEEE Computer Society, Mar. 2008, pp. 535–539.

[19] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. San Diego, California, United States: ACM, 2001, pp. 149–160.

[20] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. San Diego, California, United States: ACM, 2001, pp. 161–172.

[21] A. Brown, M. Kolberg, and J. Buford, "Chameleon: an adaptable 2-tier variable hop overlay," in *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*. Las Vegas, NV, USA: IEEE Press, 2009, pp. 770–775.

[22] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the gnutella network," *IEEE Internet Computing*, vol. 6, no. 1, pp. 50–57, Jan. 2002.

[23] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, "Freenet: a distributed anonymous information storage and retrieval system," in *Designing Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, 2001, vol. 2009/2001, pp. 46–66.

[24] K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[25] L. Z. Granville, D. M. da Rosa, A. Panisson, C. Melchiors, M. J. B. Almeida, and L. M. R. Tarouco, "Managing computer networks using peer-to-peer technologies," *Communications Magazine, IEEE*, vol. 43, no. 10, pp. 62–68, 2005.

[26] C. C. Marquezan, C. R. P. dos Santos, J. C. Nobre, M. J. B. Almeida, L. M. R. Tarouco, and L. Z. Granville, "Self-managed services over a P2P-based network management overlay," in *Proceedings of the 2nd Latin American Autonomic Computing Symposium (LAACS 2007)*, Petrópolis - Brazil, 2007.

[27] C. R. P. dos Santos, L. F. D. Santa, C. C. Marquezan, S. L. Cechin, E. M. Salvador, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco, "On the design and performance evaluation of notification support for P2P-based network management," in *Proceedings of the 2008 ACM Symposium on Applied Computing*. Fortaleza, Ceara, Brazil: ACM, 2008, pp. 2057–2062.

[28] C. Schmidt and M. Parashar, "A peer-to-peer approach to web service discovery," *World Wide Web*, vol. 7, no. 2, pp. 211–229, Jun. 2004.

[29] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, and J. Miller, "METEOR-S WSDI: a scalable P2P infrastructure of registries for semantic publication and discovery of web services," *Information Technology and Management*, vol. 6, no. 1, pp. 17–39, Jan. 2005.

[30] D. Bianchini, V. D. Antonellis, and M. Melchiori, "Service-based semantic search in P2P systems," in *Web Services, European Conference on*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 7–16.

[31] F. Banaei-Kashani, C. chien Chen, and C. Shahabi, "WSPDS: web services peer-to-peer discovery service," in *Proceedings of the International Conference on Internet Computing*, 2004, pp. 733–743.

[32] Z. Duan, Z. Zhang, and Y. T. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 870–883, Dec. 2003.

[33] E. Lavinal, N. Simoni, M. Song, and B. Mathieu, "A next-generation service overlay architecture," *Annals of Telecommunications*, vol. 64, no. 3, pp. 175–185, Apr. 2009, publisher: Springer Paris.

[34] A. Kovacevic, S. Kaune, N. Liebau, R. Steinmetz, and P. Mukherjee, "Benchmarking platform for peer-to-peer systems," *it - Information Technology*, vol. 49, no. 5, pp. 312–319, Sep. 2007.