

# On the Dimensioning of an Aggregation Service for P2P Service Overlay Networks

\*Adriano Fiorese<sup>1,2</sup>, Paulo Simões<sup>1</sup>, Fernando Boavida<sup>1</sup>

<sup>1</sup> Centre for Informatics and Systems of the University of Coimbra - CISUC  
Department of Informatics Engineering - DEI  
University of Coimbra - UC

fiorese@dei.uc.pt, psimoes@dei.uc.pt, boavida@dei.uc.pt

<sup>2</sup> Department of Computer Science - DCC  
Santa Catarina State University - UDESC  
890233-100 Joinville, SC, Brazil  
fiorese@joinville.udesc.br

**Abstract.** An aggregation service (AgS) is a P2P overlay-tier whose purpose is to aggregate and optimize the searching of services and service components maintained by service providers in a P2P Service Overlay Network (SON). AgS dimensioning takes into account the AgS size, in order to allow it to adequately perform the searching when compared with P2P SON native searching mechanisms. Suitable AgS dimensioning helps service providers to plan their infrastructures and services, allowing them to keep costs under control. This paper presents an assessment of the dimensioning of an AgS overlay. The assessment takes into account the searching response time as metric in both the environments: 1) P2P SON and 2) AgS. The assessment also takes into account the AgS own maintenance overhead, in order to compare it with the searching response time in the P2P SON. The simulation results show that, on average, AgSs whose sizes are lesser than 90% of the P2P SON size present better searching efficiency than the same searching operations in P2P SON.

**Keywords:** Services Management, P2P, Service Aggregation

## 1 Introduction

Service providers use the Internet connecting “fabric” to generate revenue, offering and operating a large variety of services. Particular services might be a composite of several intermediary services, which, in turn, are operated by third party service or service-component providers. Nevertheless, these services and service components need to be searched, grouped, composed and provisioned, in order to offer the final user a complete service.

In this scenario, service providers face problems in the reachability of the services they provide. A possible manner to cope with this problem is the

---

\* This work was partially funded by Portuguese Foundation for Science and Technology (grant SFRH/BD/45683/2008).

organization of the service providers into a Service Overlay Network (SON) [22]. This approach leads to added coverage, which allows service providers to target bigger markets and, at the same time, share infrastructural costs.

Even though the services are made available by the SON, the need for service search optimization remains. To cope with this, an Aggregation Service (AgS) was proposed by the authors in [11] [10]. AgS optimizes service and service components searching in a multi-domain environment composed of multiple service providers organized into a common P2P SON. In essence, AgS is a second P2P overlay-tier that executes on top of the P2P SON, aggregating the published services and, thus, making search processes faster and more efficient. Peers that belong to service providers and that are also part of the P2P SON constitute the AgS service. These peers play the role of aggregators and are responsible for the search optimization.

In this context, an open issue is to determine the AgS size, i.e., the number of peers making up the AgS service in order to improve the search efficiency and performance, when compared to the native searching mechanisms provided by the underlying P2P SON.

This paper addresses the problem of the AgS service dimensioning. Having in mind the stated goal and approach, this paper is organized as follows. Section 2 discusses related work. Section 3 briefly describes the AgS service. Subsequently, Section 4 describes in detail the simulated scenarios, presents the simulation results and discusses them. Finally, Section 5 summarizes the findings and discusses guidelines for further work.

## 2 Related Work

The AgS service and, consequently, its dimensioning, spans different areas in the field of network and distributed systems, briefly identified below.

### 2.1 Network and Services Management

Some indirect contributions from the network and services management area are relevant for the optimization of search services and their dimensioning issues.

Management by Policies is used to enhance services management [15]. 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, which suggests the use of an appropriate Service Overlay Network (SON) to take care of this.

Currently, web services are the most developed approach to network and services management [24]. Also, they are the most popular solution for offering service interfaces and service composition [9], on which the Service Oriented Architecture (SOA) lays on [1]. Therefore, the searching of web services is a recurrent challenge. Works in this area comprises how to select and represent information about web services, as well as ways to overcome the limitations of

the single centralized Universal Description, Discovery and Integration (UDDI) repository. Among others, proposes in this area includes searching web services by their operations based on the similarity of the desired operation [7].

## 2.2 Peer-to-Peer

P2P networks 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 rather than at randomly chosen peers, the P2P network is said to be structured. Generally, this is accomplished using a Distributed Hash Table (DHT) as the core of the P2P network. Some examples of structured P2P overlay networks are: CHORD [21], Content Addressable Network (CAN) [18] and Chamaleon [5]. If the topology is not tightly coupled, which means the peers join the network according to some loose rules, then the network is classified as unstructured. In this kind of P2P networks there is not a coupling between topology and the location of data. Instead, peers form 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 (searches) to other peers, with a limited scope. Some examples of seminal unstructured P2P overlay networks are Gnutella [19] and FreeNet [6]. Reference [14] presents an extended discussion and comparison of structured and unstructured P2P overlay networks. Meshkova et. al. [16] presents a survey of the discovery techniques used in some P2P overlay networks as well as in other types of networks.

P2P overlay networks are significantly used as supporting-tier application. In addition to the traditional file sharing applications, resource discovery is commonly executed by these overlays. Michel et. al. [17] proposed the exploitation of keywords and attribute-values co-occurrences for the improvement of keyword-based searching in P2P. An intelligent resource discovery mechanism based on weaving attributes into indexes, using locality sensitive hashing and performing searches based on the geographic location of the indexes in a structured P2P overlay is presented in [20].

Search enhancement by combining Grid and P2P was proposed in [4, 23]. Some ideas on this topic concern the use of routing indexes and mechanisms to easily spread them through the Grid; the utilization of bio-inspired algorithms in order to achieve overlay self-organization and selective search flooding, exploiting particular conditions on local caches, is proposed.

P2P used in the searching of web services has been addressed by several authors. For instance, approaches using semantics for web services searching are well studied [2, 3]. All these proposals claim that the P2P approach has some advantages for the service discovery process, when compared to centralized approaches, such as UDDI.

### 2.3 Service Overlay Networks

A Service Overlay Network (SON), is a virtualized network composed of interconnected nodes, whose generic purpose is to provide the required Quality of Service (QoS) to applications that execute on those nodes [22]. A difference between a SON and a P2P overlay network is that the latter regards providing efficient searching and retrieval. This difference is claimed in [22]. The formation of the SON does not require its own communication infrastructure. Hence, the problem of bandwidth provisioning in a SON composed of nodes that lease links from different link providers is studied in [8].

A P2P overlay network can also provide QoS services. We claim this can be accomplished 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, these particular P2P overlay networks can be considered SON. This idea is based on the work of Zhou et. al. [25]. They proposed a SON platform called ALASA. This platform uses a structured P2P overlay network on the Internet to describe, discover, compose, and repute services.

Taking this into account, this work will use a particular P2P overlay in order to assume a SON among several particular service providers belonging to different network domains.

Lavalin et. al. [13] 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.

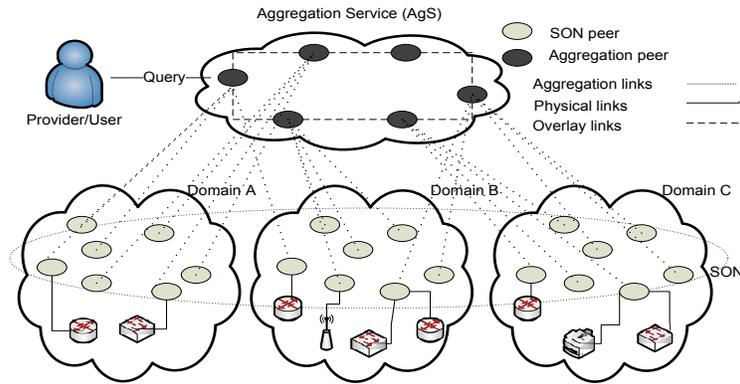
## 3 Services Searching using Aggregation Service

The Aggregation Service (AgS) is an unstructured P2P overlay, meaning there is no tight coupling between overlay topology and information location/placement. It executes on top of a P2P SON created by service providers. AgS is composed of peers that belong to these service providers, interested in advertising their services.

The purpose of the AgS service is to aggregate service and service components. This is accomplished by concentrating the service offerings in the AgS peers (nodes), in order to facilitate and optimize the search process. The architectural design of AgS is depicted in Fig. 1.

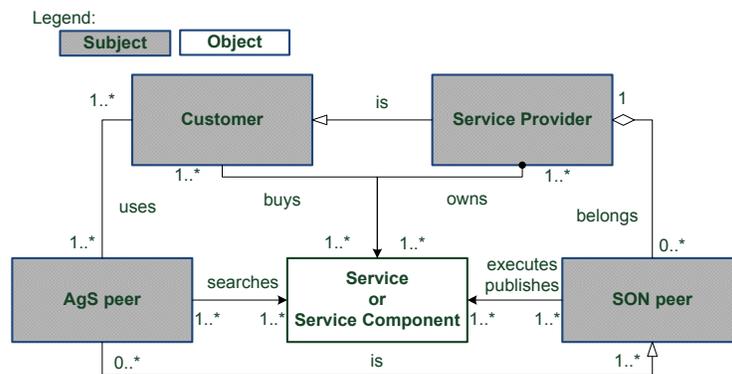
AgS consists of a P2P overlay without coupling between its logical ring topology and the exact location of the aggregated services. Fig. 1 also shows the SON peers belonging to particular administrative domains announcing (publish) their services and service components to the aggregation peers.

The AgS service operates according to the model depicted in Fig. 2. The service or service component object is the central piece in the AgS model. It is part of the interactions with every subject in the model. It represents the Service or Service Component that actually executes on the SON peer and that is published in an aggregation peer. Aggregation peers (or AgS peers) are



**Fig. 1.** Aggregation Service Architecture

specialized SON peers, chosen among the peers that compose the P2P SON in order to form the AgS P2P overlay.



**Fig. 2.** Aggregation Service Model

Each SON peer plays a double role. They execute the services and, in order to optimize the services searching, they also publish references to the available services in several AgS peers. A single service offering can be spread over multiple AgS peers in order to allow some redundancy and to overcome churn. The SON peers make the services indirectly available (through interfaces encapsulated in a service profile) to external entities (such as service composers and aggregators) located in the same or other network domains.

As service providers are customers as well, they can act as third party consumers of service components of other service providers. Nonetheless, to accomplish this step, first of all, the service or service component needs to be searched and found. According to the AgS model, the customer, which can

be a third party service provider, uses an AgS peer to accomplish this task. The AgS service performs the search and makes the result available to the customer. Searching for a service by means of the AgS framework results in a set of references to SON peers that offer an interface to the services that match 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 two-tiered AgS architecture enables the splitting of publish and search functionality from the services and service components management functions to be carried out in the P2P SON-tier. Thus, sensitive information and configuration of the services (e.g. the existing internal service provider management services, topologies, etc.) can be protected by only making available (publishing) a previously selected set of interfaces for services and service components.

The AgS working is based on a number of operations. Table 1 presents its key operations and the corresponding messages exchanged among peers.

**Table 1.** AgS Table of Operations and Messages

<i>Operation Goal</i>		<i>Executor</i>	<i>Message sent</i>
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.	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.

## 4 Dimensioning the AgS

In order to dimension the AgS service, several simulations were performed in order to determine the search response time as a function of the number of nodes in the AgS layer. The response time is the time elapsed since a Query Message is sent from the requesting AgS peer until the reception of the corresponding Query Reply Message. By determining the response time for several sizes of the P2P SON and several sizes of the AgS overlay, it is possible to decide on the number of peers that should form the AgS overlay in order to obtain a certain search performance.

## 4.1 Definitions

Let's consider a set  $P$  of service providers that create a consortium to provide services to a large-scale community over a multi-domain environment. In order to do this, they create a P2P SON, in which the available SON peers are responsible for providing the services. In this case let  $|P_n|$  be the number of peers a service provider  $p_n$  makes available as SON peers. Thus,  $|SON| = \sum_{n=1}^{|P|} |P_n|$ , and  $SON = \{p | p \in P_n \in P\}$ . On the other hand, the AgS overlay is constituted by the subset of the SON peers. Hence, in principle,  $|AgS| \leq |SON|$ . Let's define  $\mathcal{E}$  as the search efficiency, which is given by the response time metric. Thus,  $\mathcal{E}_{SON}$  is the search efficiency in the P2P SON using internal, native searching mechanisms, which is inversely proportional to the response time, i.e.,  $\mathcal{E}_{SON} = 1/rt$ . On the other hand,  $\mathcal{E}_{AgS}$  is the search efficiency in the aggregation service. However, in this case, rather than taking only into account the search performance,  $\mathcal{E}_{AgS}$  must take also into account the AgS overlay set-up and maintenance performance, to which we collectively refer as overhead. Hence,  $\mathcal{E}_{AgS} = 1/(rt + ovhd)$ , where  $rt$  is the response time and  $ovhd$  is the overhead. This overhead results from the time spent by the AgS overlay in performing control operations. Each control operation, i.e. join (j), publish (h) and leave (l), takes a varying amount of time, which depends on the size of the exchanged messages and on the underlay bandwidth and latency. Thus, the objective of the AgS service is stated in equation (1).

$$( \max |AgS| ) | \mathcal{E}_{AgS} \geq \mathcal{E}_{SON} \quad (1)$$

## 4.2 Methodology

Fifteen hundred individual simulations were performed, involving a sample of thirty particular P2P SONs with different sizes, starting with 100 peers and going up to 3000 peers, at 100-peer steps.

Each particular SON executes, makes available and publishes its services, spreading them over 10 different domains. For the sake of simplicity, a particular SON peer can only publish, 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). In the interests of simplification, the search concludes with the first match, though AgS has the ability to return all matches.

For each simulated P2P SON, four particular AgS overlays running on top of it were set up. Each one of these AgS overlays was composed of a percentage of the peers that form the P2P SON. These percentages were 10%, 50%, 80% and 90%.

Each execution simulated 50 hours of work. Each simulation performed 1,000 search operations, and they were repeated 10 times in order to get the averaged result. First, the P2P SON environment was simulated, executing the query operations. After that, each AgS overlay was simulated, over the

previous simulated SON P2P, executing the same number of query operations. A configuration file with the query operations discrete-times was used to feed the simulations. Thus, the execution of the operations followed a pre-defined temporal sequence that was kept the same for all simulations.

In order to optimize the search process, caching of the search results was also taken into account. This means that when a Query Message found the desired information then a Query Reply Message containing that information was sent to the originating aggregation peer, which stored 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, the search would then get fewer hops due to the cache hit, thus improving the search efficiency. The PeerFactSim.KOM [12] discrete events simulator was used in all simulations.

### 4.3 Results

The simulations primary result is the average response time (RT) of the search operations. When a search operation starts, the corresponding Query Message receives a time stamp (TS). Each peer along the search path forwards the Query Message in the case the service is absent from its local cache. When there is a cache hit, the time stamp of the Query Message is copied into the Query Reply Message. On reception of this message, the initiating peer can calculate the elapsed time. RT can then be calculated as the ratio between the accumulated time for all successfully accomplished search operations and the number of search messages, according to equation (3).

As already mentioned the AgS efficiency, which is expressed in (4), also depends on the overhead time. The overhead results from the time spent in setting up and maintaining the AgS overlay, by way of join (j), publish (p) and leave (l) control operations, according to equation (2).

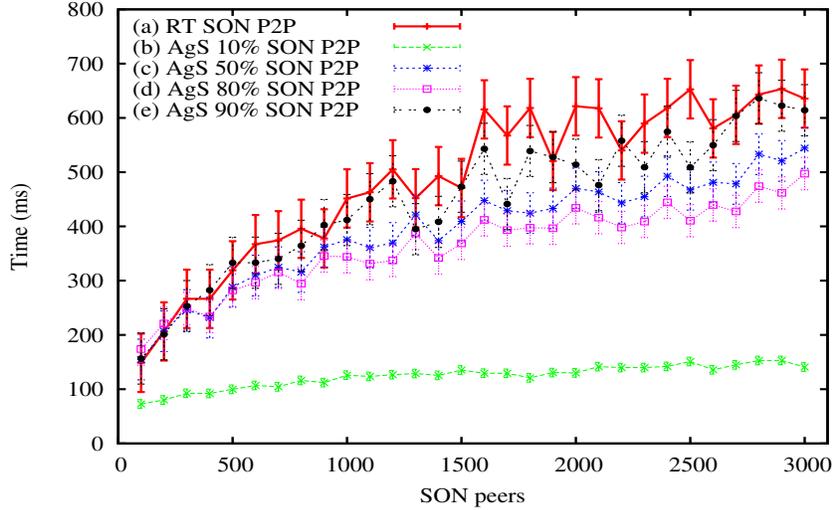
$$ovhd = \sum_{m=1}^{n_j} time(j_m) + \sum_{m=1}^{n_p} time(p_m) + \sum_{m=1}^{n_l} time(l_m) \quad (2)$$

$$rt = \frac{\sum_{m=1}^{query\ msgs} (CurrentTime - TS_m)}{QueryReplyMessages} \quad (3)$$

$$e_{AgS} = \frac{1}{(rt + ovhd)} \quad (4)$$

Fig. 3 depicts the results concerning the response time and overhead for the simulated scenarios. It is worth mentioning the results rely on a confidence interval of 95%.

It is possible to notice that the AgS service is very efficient since, for the majority of results, AgSs whose size is up to 90% of the P2P SON size still lead to faster searches (even with overhead) than the plain P2P SON. Moreover, as one can see, the smaller the AgS overlay, the better. This can be explained by



**Fig. 3.** AgS efficiency with overhead

the high concentration of services in these relatively few peers, as is the case of AgSs with 10% of the P2P SON peers.

For some P2P SON and respective AgS sizes, a particular behavior can be observed. Especially in the cases of small P2P SONs (the ones whose size is smaller than 600 peers), one can observe that the search time (efficiency) of the AgS is worse than the search time for the P2P SON. All in all, the observation of this behavior in these conditions allows us to conclude that for small market niches, where service providers create small P2P SON, for the sake of searching, the AgS must not be greater than 80% of the P2P SON.

The influence of the overhead is stronger in the smaller P2P SON and respective AgSs. As services are equally heavily concentrated in the P2P SON and in the AgS overlay, the searches are fast. Thus, even a small AgS overhead has negative effects on the searching efficiency. The influence of the overhead can be seen in more detail in Fig. 4.

Fig. 4.(b) shows that even when 90% of the SON peers are part of the AgS overlay, AgS not only searches faster than P2P SON but also the entire  $\mathcal{E}_{AgS}$  (i.e., search plus overhead) is greater than the  $\mathcal{E}_{SON}$ . On the other hand, Fig. 4.(a) shows situations for which  $\mathcal{E}_{AgS} < \mathcal{E}_{SON}$ . These cases highlight that, for AgS sizes starting at 80% (as a matter of fact, with less than 80%) of the P2P SON size, the overhead is responsible for degrading the AgS efficiency. Nevertheless, it is worth mentioning that when the overhead is dismissed,  $\mathcal{E}_{AgS}$  is always greater than  $\mathcal{E}_{SON}$ .

Finally, Fig. 5 shows the histogram for the arithmetic mean values for all experiments. The average response time in the P2P SON is approximately 487 ms whereas for the AgS service it is approximately 77, 270, 403 and 448 ms for

AgS sizes of 10%, 50%, 80% and 90% of the P2P SON size, respectively. The overhead is quite similar for all AgS sizes, with a value close to 47 ms.

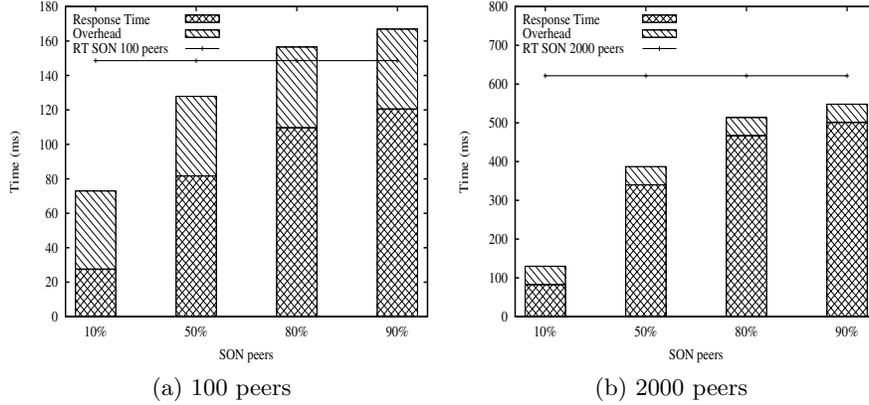


Fig. 4. AgS efficiency x P2P SON size

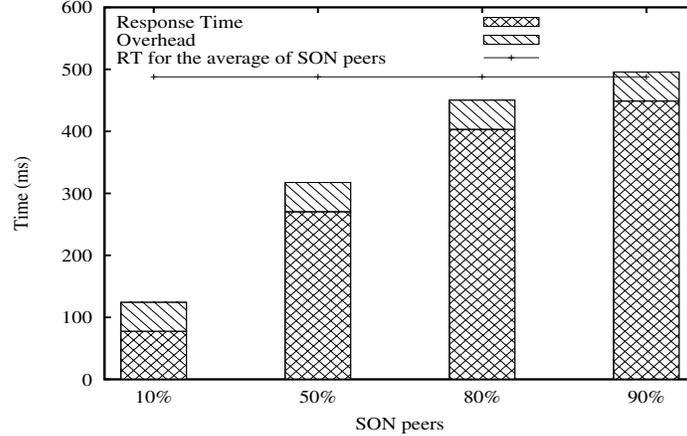


Fig. 5. Average AgS efficiency x P2P SON size

Looking at Fig. 5 it is possible to conclude that the AgS service can lead to very good performance gains relative to the plain P2P SON approach. In addition, the average overhead remains stable even with the increase in the number of aggregation peers. All in all, according the obtained results it is possible to claim that the AgS approach is highly beneficial and gives service

providers ample freedom to decide on the number of 2nd-level peers without jeopardizing the performance gain.

## 5 Conclusions

The Aggregation Service under evaluation in this paper aims at improving the service search efficiency in large-scale, multi-provider peer-to-peer service overlay networks. When the number of peer increases, search operations can be performed more efficiently if a second-tier overlay is established, comprising special peers that maintain information of the various services available in and published by the peers belonging to the first tier.

Specifically, this paper addressed the issue of determining the relation between the number of peers in the Aggregation Service and the efficiency of service search operations. The results, obtained by simulation, clearly show that the proposed service has very good potential to improve the overall search performance when compared to the realization of search operations in a single-tier P2P overlay, at the cost of a very small overhead. The obtained results can easily be used by cooperating ISPs in order to dimension the Aggregation Service overlay.

Further work will address optimization of search operations, data consistency assurance, and robustness of the aggregation overlay.

## References

1. Auer, L., Strauss, C., Kryvinska, N., Zinterhof, P.: 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 - Volume 00. pp. 535–539 (2008)
2. Banaei-Kashani, F., Chien Chen, C., Shahabi, C.: WSPDS: web services peer-to-peer discovery service. In: Proceedings of the International Conference on Internet Computing. pp. 733–743 (2004)
3. Bianchini, D., Antonellis, V.D., Melchiori, M.: Service-based semantic search in P2P systems. In: Web Services, European Conference on. pp. 7–16 (2009)
4. Brocco, A., Malatras, A., Hirsbrunner, B.: Proactive information caching for efficient resource discovery in a self-structured grid. In: Proceedings of the 2009 workshop on Bio-inspired algorithms for distributed systems. pp. 11–18 (2009)
5. Brown, A., Kolberg, M., Buford, J.: Chameleon: an adaptable 2-tier variable hop overlay. In: Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference. pp. 770–775 (2009)
6. Clarke, I., Sandberg, O., Wiley, B., Hong, T.: Freenet: a distributed anonymous information storage and retrieval system. In: Designing Privacy Enhancing Technologies, Lecture Notes in Computer Science, vol. 2009/2001, pp. 46–66 (2001)
7. Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J.: Similarity search for web services. In: Proceedings of the Thirtieth international conference on Very large data bases - Volume 30. pp. 372–383 (2004)
8. Duan, Z., Zhang, Z., Hou, Y.T.: Service overlay networks: SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Trans. Netw.* 11(6), 870–883 (2003)

9. Dustdar, S., Schreiner, W.: A survey on web services composition. *International Journal of Web and Grid Services* pp. 1–30 (2005)
10. Fiorese, A., Simões, P., Boavida, F.: An aggregation scheme for the optimisation of service search in Peer-to-Peer overlays. In: *Proceedings of the 6th International Conference on Network and Services Management (CNSM 2010)*. pp. 481–486 (2010)
11. Fiorese, A., Simões, P., Boavida, F.: Service searching based on P2P aggregation. In: *The 24th International Conference on Information Networking (ICOIN 2010)* (2010)
12. Kovacevic, A., Kaune, S., Liebau, N., Steinmetz, R., Mukherjee, P.: Benchmarking platform for peer-to-peer systems. *Information Technology* 49(5), 312–319 (Sep 2007)
13. Lavinal, E., Simoni, N., Song, M., Mathieu, B.: A next-generation service overlay architecture. *Annals of Telecommunications* 64(3), 175–185 (Apr 2009)
14. Lua, K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials* 7(2), 72–93 (2005)
15. Lymberopoulos, L., Lupu, E., Sloman, M.: An adaptive police-based framework for network services management. *Journal of Network and Systems Management* 11(3), 277–303 (Sep 2003)
16. Meshkova, E., Riihijärvi, J., Petrova, M., Mähönen, P.: A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer Networks* 52(11), 2097–2128 (Aug 2008)
17. Michel, S., Bender, M., Ntarmos, N., Triantafillou, P., Weikum, G., Zimmer, C.: Discovering and exploiting keyword and attribute-value co-occurrences to improve P2P routing indices. In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. pp. 172–181 (2006)
18. Ratnasamy, S., Francis, P., Shenker, S., Karp, R., Handley, M.: A scalable Content-Addressable network. IN *PROCEEDINGS OF ACM SIGCOMM* pp. 161–172 (2001)
19. Ripeanu, M., Iamnitchi, A., Foster, I.: Mapping the gnutella network. *IEEE Internet Computing* 6(1), 50–57 (Jan 2002)
20. Shen, H.: A P2P-based intelligent resource discovery mechanism in internet-based distributed systems. *Journal of Parallel and Distributed Computing* 69(2), 197–209 (Feb 2009)
21. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking* 11(1), 17–32 (February 2003)
22. Tran, C., Dziong, Z.: Service overlay network capacity adaptation for profit maximization. *IEEE Transactions on Network and Service Management* 7(2), 72–82 (Jun 2010)
23. Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M., Pennanen, M., Popov, K., Vlassov, V., Haridi, S.: Peer-to-Peer resource discovery in grids: Models and systems. *Future Gener. Comput. Syst.* 23(7), 864–878 (2007)
24. Vianna, R.L., Almeida, M.J.B., Tarouco, L.M.R., Granville, L.Z.: Investigating web services composition applied to network management. In: *Proceedings of IEEE International Conference on Web Services*. pp. 531–540 (2006)
25. Zhou, S., Hogan, M., Ardon, S., Portman, M., Hu, T., Wongrujira, K., Seneviratne, A.: ALASA: when service overlay networks meet Peer-to-Peer networks. In: *Communications, 2005 Asia-Pacific Conference on*. pp. 1053–1057 (2005)