UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Informática

An Environment for Services Offering in the Internet based on Peer-to-Peer Service Overlay Networks

Ambiente para Oferta de Serviços na Internet baseado em Redes de Sobreposição de Serviços

Adriano Fiorese

PhD Thesis Submitted to the University of Coimbra

(Tese de Doutoramento submetida à Universidade de Coimbra)

· 28 de Agosto de 2012 ·

Ambiente para Oferta de Serviços na Internet baseado em Redes de Sobreposição de Serviços

Tese de Doutoramento submetida à Universidade de Coimbra

Adriano Fiorese

Orientador: Prof. Doutor Fernando Pedro Lopes Boavida Fernandes

Co-orientador: Prof. Doutor Paulo Alexandre Ferreira Simões

· 28 de Agosto de 2012 ·

Acknowledgments

First of all, I would like to thank to my family. I especially thank to my mother Zuleide and my father Deoclides (*in memorian*). Thanks for your life example, and for feeding my curiosity and supporting my steps forward. Also, I thank to my two brothers, Adirciano and Adivan, for their companionship, support and friendship.

Second, I would like to thank to my advisors. I thank them for their faith on me, their advises and criticisms that helped me to achieve the end of this journey. Thank you for your patience and for being inspiration as researchers.

I would like to thank to Dr. Fernando Boavida for the talks; his support, and papers revision. I can say that I improved my writing abilities trying to follow his bright and clever interventions. I would like to thank to Dr. Paulo Simões for teaching me how to write a paper, his criticism and technical discussions.

I would like to thank to the Department of Informatics Engineering (DEI), the Centre for Informatics and Systems of the University of Coimbra (CISUC), and the overall University of Coimbra for the physical, administrative and human support that allowed me to conduct my research. I am grateful for having lived with people who work there.

I would like to express my gratitude to the Santa Catarina State University (UDESC), in particular to all people of Department of Computer Science (DCC).

During all my PhD Course I had the support of them.

I also would like to thank to the Portuguese Foundation for Science and Technology (FCT) and the European Commission for the financing support.

During my PhD Course I have met great people. Probably, in this acknowledgments section I will forget some of them. Please, forgive me for this. Anyway, I will risk quoting some of them. I would like to thank to Fernando, Alexandre, Vinicius, André, Dudu, Neto and Ricardo Mendão for their friendship and all "philosophical" and technical discussions that helped me to complete this walking. I would like to thank to Alexandre, Josi, Hugo, Cris, Gloriete, and Hilário for the friendship and for being comprehensive house mates.

Finally, I would like to thank my friends living in Coimbra and abroad that allowed this period of doctorate be not just an academic learning phase, but also a period of joy and companionship. Among friends, I quote Fernando, Alexandre, André, Dudu, Neto, Vinícius, Anderson, Celso, Rogério, Scheila, Josi, Suely, Cris, Juci, Alice, Renata, Pavla, Janaina, Roger, Thaísa, Lauro, Dien, Ricardo, and many others who did part of my life during the doctorate phase.

Foreword

The work described in this thesis was conducted at the Laboratory of Communications and Telematics of the Centre for Informatics and Systems of the University of Coimbra within the context of the following projects:

- FP6-IST CONTENT Network of Excellence - *Content networks and services for home users* - Project funded by the European Commission from July 1^{st}, 2006 to Jun 30, 2009. This project aims to boost the potential of European Community Networking by improving Content Distribution infrastructures for the delivery of live (streaming) content and interactive stored content, and by integrating, in an open way, tools and mechanisms that would enable the curation of multimedia assets and their subsequent access for the benefit of the communities of users, producing a set of appropriate services for them, both in the context of the long tail or applied to (re-purposed) assets created by traditional broadcasters.

- *Monitorização de overlays P2P e sua utilização na gestão de redes de computadores* - Project funded by the Portuguese Foundation for Science and Technology (grant SFRH/BD/45683/2008) from November 2008 to October 2011. This project aims the proposal and development of a framework enabling the formation of P2P overlays capable of monitoring operational information and sharing them accordingly with the applications that use this framework in order to enhance the applications working.

The work done during this thesis resulted in the following publications:

- A. Fiorese, P. Simões, and F. Boavida. "Approach for Service Search and Peer Selection in P2P Service Overlay". In: Proceedings of the 27^{th} International Conference on Information Networking (ICOIN). Bangkok, Thailand, January 27-30, 2013.

- A. Fiorese, P. Simões, and F. Boavida. "Peer Selection in P2P Service Overlays using Geographical Location Criteria". In: Proceeding of the 12^{th} International Conference on Computational Science and its Applications (ICCSA 2012). Salvador, Brazil, June 18-22, 2012.

- A. Fiorese, P. Simões, and F. Boavida. "Selecção de pares em redes de sobreposição usando critérios de localização geográfica". In: Actas da 11^a Conferência Sobre Redes de Computadores (CRC 2010). University of Coimbra, Coimbra, Portugal, November 17-18, 2011.
- A. Fiorese, P. Simões, and F. Boavida. "An Approach to Peer Selection in Service Overlays". In: Proceedings of the 7^{th} International Conference on Network and Services Management (CNSM 2011), Paris, France, October 24-28, 2011.
- A. Fiorese, P. Simões, and F. Boavida. "On the Dimensioning of an Aggregation Service for P2P Service Overlay Networks". In: Proceedings of the 5^{th} International Conference on Autonomous Infrastructure, Management and Security (AIMS 2011), Nancy - France, LNCS v. 6734, Jun 14-17, 2011.
- A. Fiorese, P. Simões, and F. Boavida. "Performance Evaluation of Service Searching using Aggregation in Peer-to-Peer Service Overlay Networks". In: Proceedings of the 12^{th} IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), Dublin, Ireland, May 23-27, 2011.
- A. Fiorese, P. Simões, and F. Boavida. "Avaliação de Desempenho da Busca de Serviços usando Agregação em P2P Service Overlay Networks". In: Actas da 10^a Conferência Sobre Redes de Computadores (CRC 2010). University of Minho, Braga, Portugal, November 11-12, 2010.
- A. Fiorese, P. Simões, and F. Boavida. "An Aggregation Scheme for the Optimisation of Service Search in Peer-to-Peer Overlays". In: Proceedings of 6^{th} International Conference on Network and Services Management (CNSM 2010), Niagara Falls, Canada, October 25-29, 2010.
- A. Fiorese, P. Simões, and F. Boavida. "OMAN - A Management Architecture for P2P Service Ovelay Networks". In: Proceedings of the 4^{th} International Conference on Autonomous Infrastructure, Management and Security (AIMS 2010), Zürich, Switzerland, LNCS v. 6155, Jun 21-25, 2010.
- A. Fiorese, P. Simões, and F. Boavida. "Service Searching based on P2P Aggregation". In: Proceedings of the 24^{th} International Conference on Information Networking (ICOIN 2010). Busan, South Korea, January 27-29, 2010.
- A. Fiorese, P. Simões, and F. Boavida. "Um Serviço de Agregação Baseado em P2P para a gestão de redes e serviços". In: Actas da 9^a Conferência sobre Redes de Computadores (CRC 2009). Oeiras, Portugal, October 15-16, 2009.

- A. Fiorese, P. Simões, and F. Boavida. "A P2P-based Approach to Cross-Domain Network and Service Management". In: Proceedings of the 3^{rd} International Conference on Autonomous Infrastructure, Management and Security (AIMS 2009), University of Twente, Enschede, The Netherlands, LNCS v. 5637, Jun 30 to July 02, 2009.

Resumo

Serviços e componentes de serviço têm se tornado os principais elementos em interações entre provedores e consumidores de serviços. Independentemente do tipo de serviço, eles podem estar dispersos entre provedores na grande "nuvem" que atualmente a Internet representa. Serviços estão no núcleo de negócio dos provedores e demandam gestão especial, de forma a permitir exposição máxima à potenciais clientes. Para alcançar esse objetivo, provedores de serviço podem auto-organizar-se em uma rede de sobreposição de serviços (SON). Entretanto, embora esta abordagem melhore as chances de aumento da fatia de mercado de um provedor, ela também apresenta algumas questões. Uma delas, a busca de serviços obedecendo requisitos específicos é um problema desafiador. Além disso, a busca pelo melhor provedor de serviço é da mesma forma uma demanda desafiadora.

Por estas razões, esta tese lida com os problemas da busca e seleção de serviços em um ambiente de larga escala organizado como uma rede de sobreposição de serviços (SON) sobre a Internet. Nesse contexto, uma arquitetura chamada *Management Architecture for P2P Service Overlay Networks* (OMAN) é proposta. OMAN oferece habilidades avançadas de busca e localização de serviços através da utilização da agregação de serviços e localização geográfica em nível da SON. Baseda na OMAN, esta tese apresenta as seguintes principais contribuições: i) a arquitetura OMAN dispobilizando um ambiente para oferecimento de serviços; ii) representação flexível de serviços; iii) mecanismo otimizado para busca de serviços; iii.1) proposta de um protocolo para busca de serviço em P2P SON; iv) mecanismo para a seleção do melhor provedor de serviço.

Simulações foram executadas de forma a avaliar a arquitetura OMAN tendo em conta seus principais componentes em vários cenários de diferentes tamanhos de P2P SON. Particularmente, o *Aggregation Service* (AgS) e o *Best Peer Selection Service* (BPSS) são avaliados de acordo com seus desempenhos. Os resultados das simulações são apresentados e discutidos nessa tese. Estes resultados satisfazem o pricipal objetivo desta tese, atingido pela arquitetura OMAN, que é disponibilizar um eficiente e dinâmico ambiente multi-provedor sobre a Internet para o oferecimento de serviços.

Abstract

Services and service components are becoming the main elements in interactions between service providers and service consumers. These services might include content, connectivity or complementary services. Regardless of which kind of services, they can be dispersed among providers on the big cloud which the Internet currently represents. As they are main elements, the services are in the business core and require special handling of the strategies and mechanisms involved to allow their maximum exposure to potential clients. To accomplish this, service providers can self-organize in a Service Overlay Network (SON). However, even though this approach is more likely to lead to an increase in market share, it also raises some key issues. In particular, the search of services regarding particular requirements is a challenging concern. Moreover, finding the best service provider among those that match all the searching parameters is a demanding issue as well.

For these reasons, this thesis addresses the problems of providing an efficient search and selection of services in a large-scale environment, organized as a Service Overlay Network (SON) over the Internet. An architecture called Management Architecture for P2P Service Overlay Networks (OMAN) is put forward which offers advanced abilities in service searching and location through the employment of service aggregation and geographical location at the SON level. On the basis of the work carried out, this thesis presents the following main contributions: i) the OMAN architecture that provides an environment for services offering; ii) a flexible representation of services; iii) an optimized service search mechanism; iii.1) a proposal of a service searching protocol for P2P SON; iv) a best service provider selection mechanism.

Simulations were performed to evaluate the OMAN architecture regarding its main components in several scenarios involving different sizes of Peer-to-Peer SON (P2P SON). In particular, the Aggregation Service (AgS) and Best Peer Selection Service (BPSS) were examined to assess their performances. The simulation results are presented and discussed in this thesis and were found to satisfy the main objective of this thesis which is to make available (with the support of OMAN) an efficient and dynamic multi-provider environment for services offering on the Internet.

Questions are the creative acts of intelligence. . . .

Frank Kingdon

Contents

List of Figures

List of Tables

List of Acronyms

AgS	Aggregation Service
AgS-P	Aggregation Service Protocol
AN	Ambient Networks
API	Application Program Interface
APL	Average Path Length
BPSS	Best Peer Selection Service
CAIDA	The Cooperative Association for Internet Data Analysis
CAN	Content Addressable Network
CDN	Content Distribution Network
CI	Confidence Interval
CM	Configuration Manager
DHT	Distributed Hash Table
DNA	Distributed Network Agent
FI	Future Internet
GNP	Global Network Positioning
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IP	Internet Protocol

IPv4	Internet Protocol version four
IPv6	Internet Protocol version six
ISP	Internet Service Provider
IT	Information Technology
KBR	Key Based Routing
LAN	Local Area Network
LCT	Laboratory of Communications and Telematics
Mbps	Mega bits per second
MB	Megabyte
OOP	Object Oriented Programming
OM	Overlay Monitoring
OMAN	Management Architecture for P2P Service Overlay Networks
OSI	Open Systems Interconnection
P2P	Peer-to-Peer
P2PSS	Peer-to-Peer Search Service
PBM	Policy Based Management
QoE	Quality of Experience
QoS	Quality of Service
RT	Response Time
rtt	Round Trip Time
SaaS	Software as a Service
SAP	Service Access Point
SLA	Service Level Agreement
SLP	Service Location Protocol
SLS	Service Level Specification

SOA	Service Oriented Architecture
SON	Service Overlay Networks
SSDP	Simple Service Discovery Protocol
TCP	Transmission Control Protocol
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
UML	Unified Modeling Language
VO	Virtual Organization
XML	eXtensible Markup Language
XSD	XML Schema Definition

Chapter 1

Introduction

This thesis addresses the problem of providing efficient search and selection of services in a large scale environment structured as Service Overlay Networks (SON) over the Internet. The proposed approach aims to optimize the management of service offerings that are made available by service providers. As a result, they can reach customers beyond their own administrative domains and thus ensure that their costs are shared. By adopting this approach, customers, in turn, are able to execute an optimized search for required services and retrieve the best peer providing the service. This approach was adopted in the context of the Management Architecture for P2P Service Overlay Networks (OMAN).

OMAN is an architecture developed at the Laboratory of Communications and Telematics (LCT) of the University of Coimbra and is designed to make available a set of software modules that are responsible for providing an environment facing several aspects of services management. In particular, the Aggregation Service (AgS) and Best Peer Selection Service (BPSS) are designed to allow efficient service searching in the context of multi-domain Peer-to-Peer (P2P) Service Overlay Networks (SON), and the selection of the best peer for providing a particular service.

This Chapter is structured as follows: Section 1.1 addresses some of the problems arising from inter-domain service offering and discusses the driving-force behind the development of AgS and BPSS. Section 1.2 sets out the objectives of this thesis and the contribution it seeks to make to this research field, while Section 1.3 outlines the main findings of the thesis.

1.1 Motivation and Problem Statement

The Internet is the driving-force behind new business perspectives in the world of Information Technology (IT). Its ubiquity and user-friendliness give service providers (network providers, content providers, and service providers) an opportunity to increase their share of the market by enabling them to reach more customers. Since service providers are expected to offer advanced and competitive services to satisfy their customer requirements, they may offer them on a large scale. A particular environment in the Internet is necessary to allow them to achieve this. This may depend on efficient cooperation between service providers, since the new customers may be located outside their domain boundaries. As well as the transactions between service providers that are needed to form this environment, it is also necessary to deal with the operational aspects of managing the offered services (searching, and selecting).

Currently, a particular service provider can take advantage of building its own SON to address the question of service offering, and service providing. This approach involves being responsible for all aspects ranging from the formation of SON to its operation and ending. These aspects include link communication leasing, topological arrangement, SON monitoring and enforcement, etc. [Fan 06, Duan 03]. Therefore, this approach incurs high costs that are only affordable for big service players. Thus, special service providers, generally small ones, addressing particular service market niches cannot afford the creation of their own SON. The sharing of costs among service providers seems a good way of supporting an environment that offers services on the Internet.

Moreover, the aforementioned approach only makes available services from the SON service provider owner. At the same time, it should be noted that currently, customers only require services from the portfolio of his/her Internet access provider, which had previously been contracted from a content provider. In the multi-provider environment that is envisaged, there should be no limitations of this kind because apart from the cost-sharing among the service providers, the customers can choose a service from several of the providers. Thus, in this environment, a provider offers services despite the existence of previous arrangements with other providers, and thus increases the range of those that are available.

However, there is a problem that arises from this scenario regarding the customers' options, since they have to choose the service provider that fulfils their service requirements. Thus, depending on the way the service offering environment is created, the service searching needs optimized searching mechanisms. Today, an approach is widely adopted that is based on a centralized searching mechanism called Universal Description, Discovery and Integration (UDDI) [Curbera 02]. Although the searching performance of this approach is recognised, load can be an issue, as well as it failure in the system. Moreover, the optimised searching mechanism should be flexible enough to meet the requirements of different customers, depending on the service being offered.

As mentioned earlier, in this envisaged environment, several providers will be able to offer the same service. In this case, the result of searching for a service may lead to a large number of service providers. Even when searching with refined parameters, this can still be the case. The selection of the best service provider for a particular service is a challenging issue. This issue plays a special role in such multi-provider environment for services offering on the Internet. Each service/application class has its own definition of a best partner. Thus, it is particularly difficult to define appropriate metrics that can be employed for this selection. Apart from the adoption of a metric, there is also the question of validating the selected service provider. In both cases, the problem can be overcome if the providers can rely on a common environment.

To address these problems, this thesis proposes a modular architecture called Management Architecture for P2P Service Overlay Networks (OMAN) [Fiorese 10c], which employs P2P mechanisms to provide a SON-like multi-provider environment for services offering in the Internet. Basically, OMAN makes an infrastructure available to the service providers for services trading/management in a cooperative/competitive services market. This infrastructure is based on a P2P overlay and allows the announcing (publishing), selection and providing of services across the service providers' administrative boundaries. OMAN also intends to manage aspects concerning this environment operation, such as its resilience, autonomic working and users' Quality of Experience (QoE).

Two particular problems are especially addressed by OMAN. They are the services searching (referred to above) and the selection of the best service provider. Both of them are made available as modules that work in layers. The service searching module involves the AgS [Fiorese 10d, Fiorese 10a], and makes use of service aggregation to improve the searching process. With regard to the second problem, the BPSS [Fiorese 11a, Fiorese 12] seeks to select the best service provider by employing a suitably designed metric. Therefore, the integration of these services with support services into OMAN creates a coherent infrastructure, which relies on the service providers' cooperation in a shared overlay environment, to allow them to offer, and use their services in an on-demand fashion.

1.2 Objectives and Contributions

The main goal of this work is to assist in the development of more sophisticated techniques in the Internet that can support providers in offering and providing services for customers beyond their domains. This means making an environment available where services can be offered in compliance with the customers' requirements and the service providers business expectations.

In particular, this environment can share service providers' efforts in cutting costs. Likewise, it is a particular place for customers to choose competitive services that go beyond the bonds imposed by Internet Service Provider (ISP)s. A modular architecture called OMAN has been developed to achieve this goal. The modules in OMAN allow the following to be carried out: creating and providing the environment for services offering based on a P2P SON; making available basic native service search mechanisms; providing advanced service search mechanisms based on service aggregation; providing best service provider selection based on metric comparison criteria, and making available management operations of the environment, such as support for legacy management systems, overlay monitoring, and overlay configuration management.

All in all, the contributions made by this thesis can be summarized as follows:

An architecture to provide an environment based on P2P SON for services offering in the Internet

OMAN is a service management architecture for P2P service overlay networks [Fiorese 10c] designed to work in a multi-provider environment and makes an infrastructure available to these service providers for services trading in a cooperative/competitive services market. This infrastructure allows the announcing (publishing), selection and providing of services across the service providers administrative boundaries by using a shared P2P SON. Moreover, OMAN is an architecture in three layers and its main purpose is to create a P2P SON as a service offering environment for service providers, service searching, and best service provider selection (using the best peer selection service).

Flexible representation of services

OMAN makes available a simple service representation based on service profiles. A service profile is an eXtensible Markup Language (XML) file filled by service providers which have the necessary information about a particular service. Service profiles are published in order to offer services. Besides that, the proposed environment and its optimized searching mechanism is based on the information conveyed in the service profiles. Thus, this mechanism standardizes the service offering pattern by allowing customers to make their choices easily. At the same time, it allows service providers to distinguish its services from those of competitors by using internal policies to specify its own sensitive definitions freely. As a result, this flexible representation of services can contribute to designing appropriated provider business strategies to attract more customers and hence increase market share and income.

An optimized service searching mechanism

The searching mechanism proposed by OMAN [Fiorese 10d, Fiorese 10a] provides a dynamic means of addressing the problem of searching and selecting service or service components in P2P SON environments. This mechanism is called Aggregation Service (AgS). Based on local (peer) service policies, AgS can compare the service requirements with the published parameters with a view to selecting the peers that are most suitable for providing the service. In this way, the results of the searching will match the user's required service parameters. This is carried out by an optimized aggregation scheme that takes into account the replication of previously searched service profiles, as well as the publishing service profiles distribution on the AgS.

This kind of strategy allows decoupling of service provisioning from service searching. Thus, service providers can focus on their business rules by offering their services based on the publishing of particular parameters (through service profiles) that hides the inner service polices. On the other hand, customers are only concerned about being able to find services that match the required parameters. The details of how service work or if the service is indeed a composition of several service components, are hidden. The last factor may represent a differential for those service providers that can agree to offer this kind of service.

A best service provider selection mechanism

The already stated AgS helps to optimize the search for peers that provide a particular service based on particular parameters. However, the AgS by itself, cannot handle the choice of the best peers among all those that execute that service. This means that, although the AgS optimizes the search of SON peers to provide a particular service, it does not guarantee that the found peers are those which provide adequate service performance for the requesters. Thus, in order to maximize the interaction performance between the service requester and the service peer provider, the best service provider peer, among all those that might be able to provide the desired service, must be found in the P2P SON.

Thus, accordingly the aforementioned context, the BPSS is proposed. It copes with the definition of a particular metric to carry out the selection. When it is taken into account that different services/applications have several particular ways of understanding what is meant by a best provider, it is clear that BPSS can offer a way to handle this. In fact, BPSS uses a particularly designed metric that prioritizes performance of the interactions between the service requester and service provider peers and cost reduction for service providers. Moreover, the BPSS architecture allows other metrics to be employed in accordance with the needs of the services involved.

Simulation and validation of the optimized service searching and the selection of the best service provider mechanisms

The construction of the environment was implemented in a P2P networking simulator. This construction involved several aspects of the OMAN architecture, namely the P2P SON formation, service searching, and best service provider selection. This implementation is designed to validate the proposed services and the overall potential of the OMAN architecture. Several simulations were performed and these yielded a set of data that support the stated conclusions.

1.3 Thesis Organization

The remainder of the thesis is divided into five Chapters that are structured as follows:

Chapter 2 sketches the general background of Service Overlay Networking, P2P Networking and how these two fields can be combined so that the research problems can be addressed. This chapter aims to show the need to overcome the challenges posed by the services offering market on the Internet, preparing the way to finding solutions.

Chapter 3 outlines the OMAN architecture and introduces the underlying assumptions along with the rest of this thesis; it also describes the adopted services concept and how services are represented in the service profiles. Chapter 3 also discusses the OMAN modular organization in terms of its layers and the services that compose it. This is a key chapter because it explains the organizational pieces on which lays the proposed attempts to tackle the service searching and best service provider selection problems.

Chapter 4 describes the AgS. It depicts in detail the model on which the AgS architecture is based, the way AgS is employed and the operations it relies on to search for particular services. AgS is shown here as a module that can be regarded as a central piece of the OMAN architecture which the participant nodes must implement. The AgS is also evaluated in this Chapter.

Following this, Chapter 5 outlines the BPSS, together with a description of the model and architecture that is employed for the best service provider selection. In this case, the selection result is a particular peer on a P2P SON structure since the required service is running on behalf of a service provider in a peer (node). Thus, this means that this chapter offers a way of reaching a best peer selection. A simulation evaluation regarding this matter is also presented in Chapter 5.

Finally, Chapter 6 concludes this thesis by summarizing its main contributions and pointing out directions for further work

Chapter 2

General Background

Services and service components are becoming the basic elements in interactions among service providers in the Future Internet (FI) [Domingue 11, Tselentis 09]. These services and service components might include content (e.g. finding vendors or dubbing and subtitling services for a specific movie), connectivity (e.g. interconnection links which satisfy Quality of Service (QoS) and security requirements between the consumer and the origin of the content) 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).

Currently, it is natural to consider the Internet to be the *de facto* infrastructure for providing services to users. Due to its ubiquity, service providers are able to reach users that are beyond their domain boundaries, and thus increase their market share. As a result, the Internet can be seen as the driving-force behind service convergence. Moreover, network convergence, which is one of the objectives of the Future Internet [Gutierrez 08], also leads to service convergence scenarios where service providers cooperate to provide their services.

Service Overlay Networks (SON)) [Fan 06, Liang 07, Andersen 01, Duan 03] can fulfil the requirements to leverage cooperation/competition between service providers.

These networks enhance the ability of the service providers to make their services or component services available. SON acts as an infrastructure on the Internet and allows services to be published/offered, whose users can access to select and use what they require. This infrastructure should be formed and supported by the service providers that are committed to creating a collaborative/competitive environment.

The attempt to form an infrastructure facing management of services over the Internet can be a huge undertaking. Peer-to-peer (P2P) technology can be used as a means of overcoming some of the obstacles. In addition, the use of P2P technology to create and manage a SON is beneficial because it can help 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 that the service providers organize themselves into a P2P SON, they also can keep their sensitive business information hidden while the services are being dispersed and offered.

Therefore, in order to achieve its objectives, this thesis straddles two particular areas in the field of network and distributed systems: Service Overlay Networks; and Peer-to-Peer Networking. The inter-relationship between these sub-areas shapes the attempts made to develop an efficient, scalable, and robust infrastructure that is able to cope with the services market.

In the following Sections will be examined previous work on each of these sub-areas, which are related with the problems of service searching optimization and peer selection in large-scale environments. Thus, this Chapter is structured as follows: Section 2.1 discusses Service Overlay Networks (SON), starting with aspects of basic overlay networks, and outlining issues regarding the formation of a large-scale service provider consortium to cope with the services market. Section 2.2 discusses Peer-to-Peer overlays as a suitable approach to deal with the services market environment. Finally, Section 2.3 discusses the joining of SON and P2P, and highlights the two main problems faced by this thesis: services searching (Subsection 2.3.1) and peer selection (Subsection 2.3.2).

2.1 Service Overlay Networks

Overlay networks facilitate the organization of application-specific address spaces by constructing a logical network on top of the underlying physical network [Aberer 05b]. This means that nodes in an overlay network can be thought of as being connected by logical links and thus correspond to a path in the underlying network [Ding 09]. This kind of organization is well suited to face issues in many fields of current digital network communications, such as telecommunications, Internet TV broadcasting, content distribution, and so on. Figure 2.1 depicts an overlay network for connection between several Local Area Network (LAN).

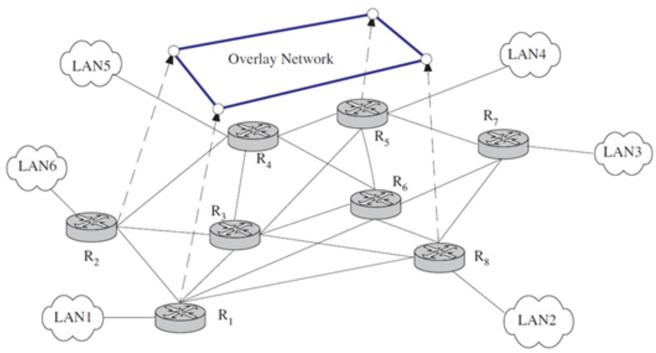


Figure 2.1: An Overlay Network for connections between LANs (Source: [Ding 09])

Overlay networks are well suited to cope with several problems in the current Internet services scenario. Some of these concern scalability, collaboration and services availability. There have been several studies of these matters. The improvement on networking communication is the reasoning behind some of them. Examples of this overlay network class are MBone [Eriksson 94], Overcast [Jannotti 00], and End System Multicast [Chu 00]. These systems are concerned with delivering data by using multicast paths built on existing underlying network infrastructures. They improve functionalities that rival those of IP Multicast protocol. Resilient Overlay Networks (RON) [Andersen 01] also frame the overlay network communication class. In a RON, each RON node monitors the quality of the links in the underlying Internet and spreads this information to the other nodes; this enables a RON to detect and react to path failures within several seconds, rather than several minutes, and allows it to select application-specific paths based on performance.

In its turn, content distribution also benefits from overlay networks. The Content Distribution Network (CDN), such as Akamai [aka 12], uses overlay techniques and data caching to improve the performance of content delivery for particular applications such as HyperText Transfer Protocol (HTTP) and video streaming.

In addition, SON is a particular class of overlay networks that address the Internet services market [Semret 00]. This overlay network class aims to provide an environment where service providers can make available their services and thus reach more customers and share costs. A SON is *"an application-layer network operated by a third-party Internet Service Provider (ISP) that owns a set of overlay nodes, residing in the underlying ISP domains, interconnected by overlay links"* [Capone 08]. On the basis of this definition, a SON is an overlay network though it is more specialized and standardized. Fan et al. [Fan 06] state that a SON is formed and deployed by overlay network providers. The last concept enables anyone to be a SON operator, and not necessarily an ISP. The SON concept leverages when QoS capabilities are added to the system [Duan 03, Tran 10, Vieira 04]. For this reason, Adami et al. [Adami 09] classify SON as *End-User Overlays* and *Backbone QoS Overlays*. The former is related to the SON construction between end hosts, without any QoS support from the intermediate nodes. The latter regards the SON formation as a means of dealing with QoS issues [Lam 07] by overlay providers that lease links.

It should be taken into account that the SON concept outlined in this thesis copes with an overlay environment composed of service providers and deployed to offer and execute service and service components. As a result, service providers can be particular overlay network providers that stand together to form an overlay to sell their own networked services, such as billing, video streaming and video on demand, transport connection, etc..

The concept of service overlay networks applies to a broad range of network architectures. The Ambient Network (AN) [Belqasmi 08] architecture comprises the ubiquitous provisioning of existing and new services over any access technology and any type of network. The composition concept plays an important role in this architecture. Thus, different protocols forming different networks (e.g. wired or wireless) can be used as "bricks" to compose a particular network or to join different networks to allow them to work together. In this context, ANs are overlays whose purpose is to provide services. Hence, in the context of AN, Mathieu et al. [Mathieu 07] proposes a self-management approach to create, configure, adapt, contextualise, and finally tear down specific overlay networks.

If similar objectives concerning the stated SON concept are examined, it is found that approaches like Virtual Organizations (VO) [Camarinha-Matos 05, Afsarmanesh 05] can cope with the provisioning of an environment that involves collaboration between service providers. This collaboration aims to provide benefits to service providers like sharing costs and infrastructure and leveraging market share. Consumers can benefit as well. A wider range of services can be made available, as well as an environment where price and quality can be competitive differentials. This environment needs the involvement of service providers and agreement between them. A way to implement these concepts is to use P2P as the network and communication infrastructure and the means of achieving the collaboration needed.

Collaboration is a key issue when composing a SON. Goel et al. [Goel 07] propose a network architecture that is similar to the VO proposal and supports an adaptive collaborative environment for engineering design through the use of services. These services represent specific engineering tasks for designing an aircraft engine. The proposed architecture ensures that these services are available in a P2P overlay network of collaborating expert service providers. The authors claim that for this reason this collaboration reduces the need for manual intervention in the project design.

SONs can rely on a P2P design to achieve the desired levels of collaboration. Several works defend the SON construction as an approach for services market on P2P [Zhou 05, Goel 07, Michiardi 07]. P2P presents several characteristics that are well suited to the SON construction and collaboration. Thus, some background regarding the main concepts of P2P are outlined in the next section.

2.2 Peer-to-Peer Networking

The concept of P2P Networking is not new. The basic Internet routing mechanism can be reported as a particular example of the application of the P2P concept [Aberer 05a].

However, with the widespread use of P2P and the misleading terminology employed, there is a need for a common understanding in this field. In order to overcome this lack of taxonomy, Aberer et al. [Aberer 05a] proposed a terminology for the P2P field as well as a modular architecture regarding the principles for P2P applications development. This taxonomy and architecture are intended to allow researchers and users to do the following: assess the properties of concrete P2P systems, establish a common vocabulary for scientific discussion, allow a qualitative comparison of P2P systems, and cope with P2P classification regarding the P2P concept.

According to Oram et al. (2001 apud [Steinmetz 05], p.10) a P2P system is "*a self-organizing system of equal, autonomous entities (peers) which aims for the shared usage of distributed resources in a network environment avoiding central services*".

Although this classical definition of P2P requires equality between peers, hybrid P2P systems can explore different roles for them [Vu 10]. Unlike the classical definition for pure P2P systems, hybrid P2P systems, such as Super Peer architectures [Jesi 07, Beverly Yang 03], introduce hierarchical layers among peers. In fact, P2P systems that are classified as hybrid, comprise some peers that act as equals between super peers and as servers or servants for a particular group of other peers. Yang and Garcia-Molina [Yang 01] have surveyed and compared several hybrid P2P systems.

Despite the dichotomy between pure and hybrid P2P systems[1] several other classification systems exist. This thesis copes with a classification system based on application-tiers that regards the development of generic multi-purpose P2P systems as first-tier application and the use of P2P systems to tackle or solve problems, as support-tier application.

The development of P2P networks (overlays) and their optimization, is the most common research subject to appear as a first-tier application. In the light of this, P2P networks can be classified according to different criteria. Using the topological criterion, they are generally distinguished between: 1) Unstructured and 2) Structured. Thus, on the one hand, the P2P network is structured when

[1]This thesis uses the term P2P systems in order to do not distinguish between Peer-to-Peer networking and Peer-to-Peer computing but to underline the fact that Peer-to-Peer can be regarded as a property, characteristic, method or mechanism.

the topology is tightly controlled and content is placed at specific peers instead of randomly chosen. Generally, this is accomplished by means of a Distributed Hashing Table (DHT) as the core of the P2P network. Despite many examples of structured P2P overlay networks exist, only a few seminal examples are listed here: CHORD [Stoica 01]; Content Addressable Network (CAN) [Ratnasamy 01]; Pastry [Rowstron 01]; Tapestry [Zhao 04]; Chameleon [Brown 09]. These few examples comprise a broad range of alternative topological arrangements, and several particular overlay routing schemata.

On the other hand, if the overlay network topology is not tightly coupled, which means the rules that govern the way peers join the network are loose, then this network is classified as unstructured. Generally, in this kind of P2P network, peers use a kind of flooding technique to send queries (searching) with a limited scope and through their neighbors. As in the case of structured P2P overlay networks, a myriad of examples can be found for unstructured networks. However, just for the sake of illustration, this thesis only presents few seminal and well known examples of unstructured P2P overlay networks: Gnutella [Ripeanu 02] and FreeNet [Clarke 01]. Lua et al. [Lua 05] conducted an extended discussion involving a comparison of structured and unstructured P2P overlay networks.

P2P overlay networks are also significantly used on the strand of support-tier. In this case, these networks are tools to achieve a goal. However, unlike P2P application-tier, typical P2P problems, such as P2P overlay routing, are not directly addressed. Thus, applications choose a particular P2P overlay network in accordance with the needed characteristics, which means that several fields can take advantage of P2P overlay networks.

For instance, in the network and services management field, some works [Granville 05, Marquezan 07, dos Santos 08] propose cooperation between network managers by mutual P2P-based exchange of messages, and also by allowing the management tasks to be carried out by groups of peers. Binzenhöfer et al. [Binzenhöfer 06] propose a distributed, self-organizing, generic testing and QoS monitoring architecture for IP networks. The architecture is based on equal agents denoted as Distributed Network Agents (DNA), and form the management overlay. The overlay self-organization is achieved by the Kademlia P2P network [Maymounkov 02]. In the context of AN, P2P is also used for network management. As already stated, network composition is a key issue for AN, and in this process it is possible that several different network management systems will combine and form a consistent management system for the composed network. Thus, Simon et al. [Simon 05] faced the challenges raised by the complexity of the composition and decomposition of ANs, by proposing a management architecture based on P2P and policy-driven management paradigms. That proposal structures the management application in accordance with a dynamic hierarchical overlay network that defines hierarchical management domains, and provides a scalable framework for several network management tasks.

Besides the traditional file sharing applications, resource discovery is commonly executed by structured and unstructured overlays. This discovery is usually carried out by either using simple (or a combination of) keywords or by the development and deployment of semantic overlays. The former strategy is also the earlier and most widely employed. Michel et al. [Michel 06] proposed the exploitation of keywords and attribute-value co-occurrences for the improvement of keyword-based searching in P2P. Their work aims to make an improvement in the queries routing, by enhancing the info-route indices kept in the peers, through multi-key statistics and the possible correlation between keywords used in the searches. An intelligent resource discovery mechanism based on weaving attributes in indices by means of using locality sensitive hashing and performing searches based on the geographical location of the indices in a structured P2P overlay is presented in [Shen 09]. The search performance of these approaches relies on the optimization of the metadata used for performing the queries routing. Despite achieving good optimization, the correlation and weaving still depends on the keywords received as input. Bad quality of input keywords can lead to inaccurate metadata creation, which can lead to low query hit ratios.

On the other hand, several studies have been introduced on the strand of semantic searching using P2P. Full text searching is a pursued goal in semantic searching. One approach to accomplish this, is clustering objects (content) according to particular rules. The construction of hierarchical semantic overlays based on clustering objects is the subject of several proposals [Kim 11, Jin 06]. Another approach is based on the semantics involving peers. Crespo et al. and others [Crespo 05, Wang 08] proposed that the neighbouring relationships between peers should be based on semantic connections, which increases the odds of finding similar results in a cluster of peers rather than clustering objects.

2.3 Joining P2P and SON

The joining of the SON and P2P fields offers a high potential for handling services. This combination as an infrastructure support-tier raises questions about service searching optimization as well as the interaction issues among the peers that execute these services.

This work proposes joining P2P overlay network and SON in order to provide QoS services when the participant service providers are in a consortium that establishes well-defined SLAs to regulate the contribution of each one to the network. This joining is also seen in a platform called ALASA [Zhou 05] that uses a structured P2P overlay network over the Internet to describe, discover, compose, and maintain the reputation of the services.

P2P is also used as a support for the SON architecture [Lavinal 09]. In their detailed study, the authors address discovery of services by considering QoS factors in their approach. P2P was chosen as the SON supportive infrastructure due to its ability to handle scalability and fault tolerance.

As a matter of fact, several SON with services being offered in P2P overlay networks, as supportive infrastructure, have already been proposed. Goel et al. [Goel 07], for example, outline an architecture of this type, where specialized services can be found and orchestrated by following SOA processes. Michiardi et al. [Michiardi 07] also use P2P for the design of SON topologies that take into account the Nash equilibrium [2] of clustered service replicas.

[2] According Michiardi et al. [Michiardi 07], overlays in equilibrium are characterized by interconnected clusters of nodes that instantiate the same service replicas.

Although the joining of P2P and SON can leverage the services consumer market, some issues arise. How can the services be found? Which peer can provide a service with the best properties? These issues are discussed in the next sections where some related work is referred to.

2.3.1 Services Searching

Services searching is a crucial bottleneck in a large-scale service market environment. Ultimately, service searching can improve the users' Quality of Experience (QoE) when it is conducted well. However, the services search optimization still lacks appropriate treatment. The utilization of a SON based on P2P as infrastructure for searching can help address this issue.

Thus, Meshkova et al. [Meshkova 08] went beyond the discussion about P2P overlay networks, and surveyed the discovery techniques used in some P2P overlay networks, as well as those in other networks. In this thesis, the interest lies particularly in service searching. Services can be described according to their actions, service providers information, etc.. Thus, approaches involving keyword or semantic searching can be used according to needs. Ultimately, services in general can fit these approaches.

Currently, web services are the most popular solution for offering service interfaces and service composition [Weerawarana 05, Peltz 03, Fu 04, Dustdar 05], which the Service Oriented Architecture (SOA) relies on [Auer 08]. Therefore, the searching of web services is a recurring challenge. Studies 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 other objectives, proposals in this area include searching web services through their operations, based on the similarity of the desired operation [Dong 04]. Thus, Schmidt and Parashar [Schmidt 04] propose a structured P2P overlay based on CHORD to improve searching by using multiple keywords. The introduction of semantics in the search has also attracted a lot of interest [Verma 05, Bianchini 09, Banaei-Kashani 04]. With regard to the service discovery, these proposals claim to have advantages over the centralized approaches that 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).

Mischke and Stiller [Mischke 03] propose searching based on group-of-interests (GoI). Their approach involves a P2P overlay lookup and routing schema where peer IDs and content keys are organized into group-of-interests. These GoIs are created on the basis of particular information about the content keys or peers when peers join the overlay. Thus, the content is looked up inside a particular GoI, which avoids flooding the whole network. The corresponding GoI is accessed by content key hashing. This approach is particularly efficient, even though the GoI formation is a sensitive point.

Beyond native P2P searching mechanisms on P2P networks, which can use hybrid approaches and particular decentralized searching protocols, there is a myriad of data searching methods, techniques, approaches and efforts that can assist services searching in a SON based on P2P environments. Ubiquitous computing is a field where acquaintance with a device or service is crucial. Therefore, searching plays an important role here. Edwards [Edwards 06] has carried out a survey of ubiquitous searching mechanisms. In this work, several discovery mechanisms and protocols are set out and compared, including Simple Service Discovery Protocol (SSDP), Bluetooth, Service Location Protocol (SLP), etc.. However, the surveyed discovery mechanisms lack an ability to handle large-scale environments for services searching.

Even in the wireless sensor networks field, searching plays the role of a cornerstone. Resource-less and energy limitation are dramatic constraints in this field, regardless of whether the search is centralized or decentralized. Thus, it is desirable for the service discovery process to be as less computationally-intensive as possible and at the same time reliable enough to meet the level of quality of service required. A virtual service overlay architecture for wireless sensor networks is presented in [Munir 09]. This architecture proposes a scheme that provides the service client with minimal interaction thus allowing efficient utilization of network resources. Even though this architecture achieves a good performance, it is well suited to wireless environments where sensors are widely dispersed and supported by directory agents that keep info about certain services. However, this approach is not well suited to large-scale environments delivered on the Internet.

2.3.2 Peer Selection

Although P2P eases the construction and maintenance of SONs, it does not guarantee adequate performance of the service utilization. In order to maximize this, the best peer must be found in the P2P SON, among all the potential partners that provide the desired service. Naturally, the choice of the best peer should take into account one or more of a set of Quality of Service (QoS) parameters, such as delay, jitter, available bandwidth, etc.. Nevertheless, to minimize inter-provider traffic and, thus, reduce costs for the user and service provider, choosing a peer belonging to a different, remote domain should be avoided as much as possible. Thus, locality should also be taken into account when choosing a SON peer as the best peer to serve a requested service.

Haase et al. [Haase 07] explore neighbouring peer relationships and shared peer expertise in order to select peers. The use of artificial intelligence techniques, like machine learning, is another approach to peer selection, which can also take advantage of the peers' expertise [Bernstein 03]. This approach is designed to adapt the selection process to the peers' requirements.

In file sharing, the free-riding problem encourages the adoption of incentive mechanisms as a part of the selection scheme. Thus, the fairness between uploads and downloads is used as a metric for the selection of the best peer. Bittorrent [Huang 08] is an emblematic example for this. However, our scheme does not target file(data)-sharing environments. Rather, the problem is to select the best peer that satisfies the requirements of the intended service. Therefore, in our case, performance is used instead of fairness. In addition, unlike file sharing applications, our approach considers the best peer selection process for long-lasting sessions, as opposed to relatively short burst chunk downloads/uploads.

In the P2P multimedia stream field, there have been several studies proposing the use of P2P as the delivery mechanism that face peer selection issues [Habib 06, Wu 05]. These are similar to our approach in so far as they aim at optimizing peer choice regarding the performance of the service.

Furthermore, P2P traffic is an issue faced by ISPs and over the Internet in general. Several proposals have been put forward to assist ISPs in avoiding costs by choosing the best peers from their own domains [Aggarwal 08, Choffnes 08, Zhuang 09]. These works advocate collaboration between providers and P2P applications. In contrast, this thesis proposes the Best Peer Selection Service (BPSS) scheme which selects best peers for service interactions inside the geographical domain of the service requester, without the need for collaboration between the service providers.

2.4 Summary

The evolving digital economy is increasing the need for new business arrangements. These new arrangements aim at reducing costs, while, at the same time, leveraging the market share to reach new consumers beyond the administrative boundaries set for the providers. One approach to achieve this is to form collaborative enterprise consortiums. This approach is well suited to the services market over the Internet, where service providers can organize themselves to ensure that their services can be offered through a Service Overlay Network (SON) and thus reduce costs and increase income. Several initiatives have been put forward to attain this objective. The joining of the classical concepts of SON and P2P network is one of them. As a result, the existing Internet infrastructure can be used by service providers in order to compose a SON that aims at offering its services without the cost of leasing particular communication links.

Although this approach is suitable for large-scale environments, some issues arise regarding the optimization of services search and best peer selection. With the scaling on the service provider numbers, several providers can offer the same service leveraging a healthy cooperative/competitive environment. In addition, several proposals deal with the searching and best peer selection in these environments. However, they still lack a comprehensive approach that can effectively integrate these essential functionalities in a structured and coupled architecture. In its turn, this architecture must support the formation of these environments and integrate the management of their functionalities with other aspects regarding the services offering market. Such an architecture is proposed in this work and outlined in detail in the next Chapter.

Chapter 3

OMAN - A Management Architecture for P2P Service Overlay Networks

With the advent of Service Overlay Networks (SON) [Fan 06, Liang 07, Duan 03], new business players involved in service provisioning are beginning to appear. As in any other business investment, service providers weigh up costs and income, and resort to any competitive advantage in order to maximize their profits. A competitive advantage can be achieved through effective service management, and by exploiting the features of the overlay network used for service composition. Appropriate management of the overlay network abstraction, which can include autonomous systems (AS), cloud infrastructures, communication links, and so on, allows service providers to offer and run services in a way that improves the Quality of Experience (QoE) of the end users.

Service providers can benefit from the use of a SON. First of all, they can share costs when joining a group to offer services. Secondly, they can reach more users beyond their administrative domains and thus increase their market share. This latter advantage has a direct effect on their income.

In spite of this, current approaches to the SON offering services lack strategies of adaptation to the flexible needs of new applications and services intended for the Future

Internet. OMAN, in its turn, offers advanced search and location abilities by employing service aggregation and geographical location at the SON level. These abilities allow the SON service providers to ensure greater efficiency and a higher degree of flexibility in the provision of the required services and service components. In view of this, the SON users, which include third-party service providers besides the original P2P SON founder service providers, benefit from OMAN since it is a framework that can select the best located and desired service and service components to interact with.

Bearing in mind the stated, this Chapter is structured as follows: Section 3.1 discusses some basic assumptions underlying the context and use of OMAN. Subsection 3.1.1 defines the concepts of Service and Service Components used in this thesis and explains the service profiles which are used in the context of OMAN to represent the provider's offers. Section 3.2 provides an overview of the OMAN architecture and describes its functions. Following this, Section 3.3 outlines the services/modules that compose OMAN on a layer by layer basis. Finally, Section 3.4 concludes with a summary of the topics discussed in this chapter.

3.1 Assumptions

Offering services in a large-scale, highly dynamic multiple-domain environment is a highly complex task, since it requires handling several aspects of the interaction between the players involved in the process. Since OMAN only tackles some of these aspects, a few general assumptions have to be made to support its approach. These assumptions are as follows:

- A SON formed by service providers: It is assumed that a SON will be formed among the service providers. This grouping of participant service providers can reduce maintenance costs through peering communication links for instance. Long-term contracts can be drawn up to establish this SON [Pouyllau 10], and define some of the conventions that the providers must follow to support and guide their operations. These conventions include the following:

  - Universal identification: Every service provider in the SON has its own identification which is widely used within OMAN. Furthermore, this unique identification can be extended to the providers' available nodes, the services running in these nodes, service profiles, and so on. For instance, the global identification of a provider's node is a local identification (identification in a provider's context) which is concatenated with the provider's identification. An OMAN node is a node that fulfils the requirements of the OMAN architecture and belongs to a provider engaged in a P2P SON that plays one or more of the roles designed by the OMAN architecture;
  - Standard representation of service profiles: The information contained in the service profiles and in the SLAs, follows a form of representation that is recognized by all the providers in the SON. For instance, every service profile has the same tags to indicate the service name and the keywords by means of which it can be looked for;
  - OMAN nodes accessibility: Every OMAN node in the P2P SON knows the addresses of a particular set of other OMAN nodes that are hierarchically superior and is permitted to start publishing and search operations for services in that set; and
  - Synchronized timing: Synchronized timing between all the service providers in the SON.

- SON provisioning: The negotiation mechanisms employed by the participant service providers are not examined in this study. Thus, it is assumed that service providers (either on their own or using third party agents) are able to exchange and enforce the SLAs to form and provision the SON between the participants.

- Security and billing issues: Security and billing are important issues that have to be handled in order to create a complete inter-domain services offering infrastructure. However, as these are not within the scope of this study, it is assumed that providers are able to exchange information in a secure way, and that a third party billing agent can handle payment issues in the environment provided by the OMAN architecture.

3.1.1 Service and Service Components

Services are the core elements in interactions between providers and consumers. In this study, there are no constraints about the meaning of the expression service, except for the need of a communication network to enforce access, which means a particular application layer protocol, and ensuring that the access is performed by a software interface as in [Duan 03]. This study is particularly concerned with services that deliver some value added data which is recognized by the service consumer. Some examples are as follows:

Content: Video streaming, video on-demand, dubbing and subtitling, audio, voice, personalized news, are some of the most representative ones;

Connectivity : Access and interconnection services that satisfy QoS and security requirements.

The concept of service components used in this study comprises auxiliary services that can be a part of a chain of services in a new composite service. These complementary services are offered by specialized service providers and can be used by other service providers to offer a third party service in the SON and hence to all the consumers that have access to the SON. Some examples of service components are as follows:

Billing: This category of complementary services is essential at the content delivery chain;

Security: It is possible to use some service components to encrypt content so that it can be delivered to a particular group of authenticated paying consumers;

Controllers : Multi-session and video-conferencing controllers.

Service providers working with these categories are those that join to form the SON so that they can offer their services and service components. These services and service components are dispersed among the service providers that form the SON. They have to be offered and searched for in the SON allowing them to be subsequently used. A possible way to accomplish this is the use of service profiles.

3.1.1.1 Service Profiles

Service profiles are a simple abstract representation of services and service components conceived as eXtensible Markup Language (XML) documents. Providers make their offers available by publishing service profiles, which are used by particular modules of the OMAN architecture. The adoption of XML (as the data structure language to represent the profiles) improves the exchange of information as well as the handling of data. Since services and service profiles represent the resources providers are willing to make available, the service profiles can be considered the virtualization of these resources.

The service profiles can be regarded as a starting-point for the interaction between the service and its consumer. Thus, the service consumer will find useful and important information about how to connect to the service in order to negotiate and establish Service Level Agreement (SLA)s or how to negotiate the composition process, depending on the case. This information can be found at the service profile technical section, also known as Service Level Specification (SLS).

Figure 3.1 shows the XML Schema Definition (XSD) of the service profile. Owing to page size restrictions and the need to avoid affecting the readability of the XSD, Figure 3.1 only displays the root element and its direct child elements. Appendix A provides a detailed description of the entire service profile XSD.

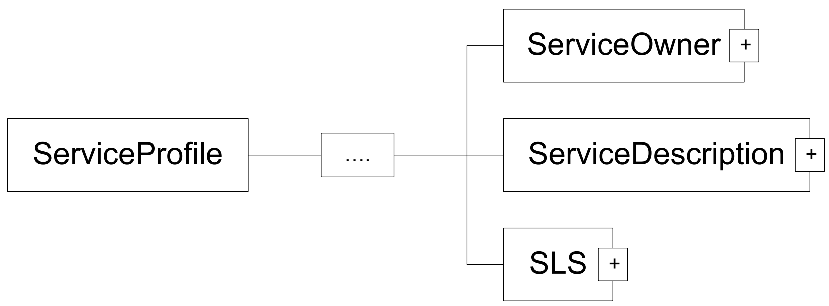


Figure 3.1: XSD - Root service profile

The ServiceProfile element (Figure 3.1) represents the root element of the XSD for service profiles. It is the main element of the profile, and encompasses the administrative and technical information regarding the services (or service components). Administrative information identifies the service and its provider. Technical information can be used by the service consumer (e.g. a third party service provider, service composition platform) to access the service. The child elements of the ServiceProfile are as follows:

- *ServiceOwner*: Encompasses the identification and the contact information of the provider that offers the service element;
- *ServiceDescription*: Contains a general description of the service or service component, such as the name, type and publication date;
- *SLS*: Contains technical information about the service element.

Service profiles allow providers to define their services and service elements in formal terms, thus creating a standard data structure where information can be published and searched in a seamless fashion. This means that each service provided must have a document containing data specified by the XSD service profile.

3.2 OMAN Overview

OMAN as a service offering architecture for P2P service overlay networks [Fiorese 10c] is designed to work in a multi-provider environment. OMAN makes an infrastructure available to these service providers for services trading in a cooperative/competitive P2P services market. This infrastructure allows the announcement (publishing), selection and provision of services across the administrative boundaries of the service providers by using the shared P2P SON.

OMAN is composed of layered services and uses a modular approach to assemble them. Thus, basically, services at lower layers provide functionalities (services) of their own, as well as for the above layers. The publish paradigm is used for OMAN to make a way available to the service providers to announce their services. Once the services have been announced, the service consumers can use OMAN to search and select what is required.

The design of OMAN allows services to make use of key information at SON level to leverage its execution. As an example, third party service providers can make use of the OMAN aggregation service in order to choose the service components they need to compose a new service. Moreover, they can also choose the closest nodes that offer those needed service components by using the best peer selection service.

Figure 3.2 introduces the OMAN architecture and shows that the OMAN purpose is twofold. On one hand, OMAN enables compliant applications and services to take advantage of its inner services. This means, for instance, that a service executing in an OMAN node and responsible for streaming video on demand, can *choose* the video source provider closest to it in the SON in order to provide its client with the video streaming with the required parameters of QoS.

On the other hand, external entities can also benefit from the OMAN capabilities. Thus, for instance, third party service providers, (not necessarily belonging to the OMAN compliant P2P SON), can use the OMAN to search for service components in order to compose them and thus be able to deliver a new composed service to a final user. Therefore, for instance, a third party streaming video service provider can use OMAN to search the video source, and dubbing or subtitling service components that fit the required parameters set by the final user so that they can be assembled in a new service and delivered.

Access interfaces are provided to make it possible to use the OMAN services. These interfaces act as service access points (SAP) that are implemented as API. These interfaces are represented as circle end lines in Figure 3.2. A circle end line means an interface for accessing the functionality offered by the service/module where the line starts, whereas the arrow end line means the service/module (where the line starts) is offering some particular information. As can be seen, the interfaces offered to external entities are the ones whose circle end line crosses the borders of the OMAN.

These interfaces are made available by modules, each of which executes one or more services. These modules compose an architecture in three layers. Each OMAN layer is

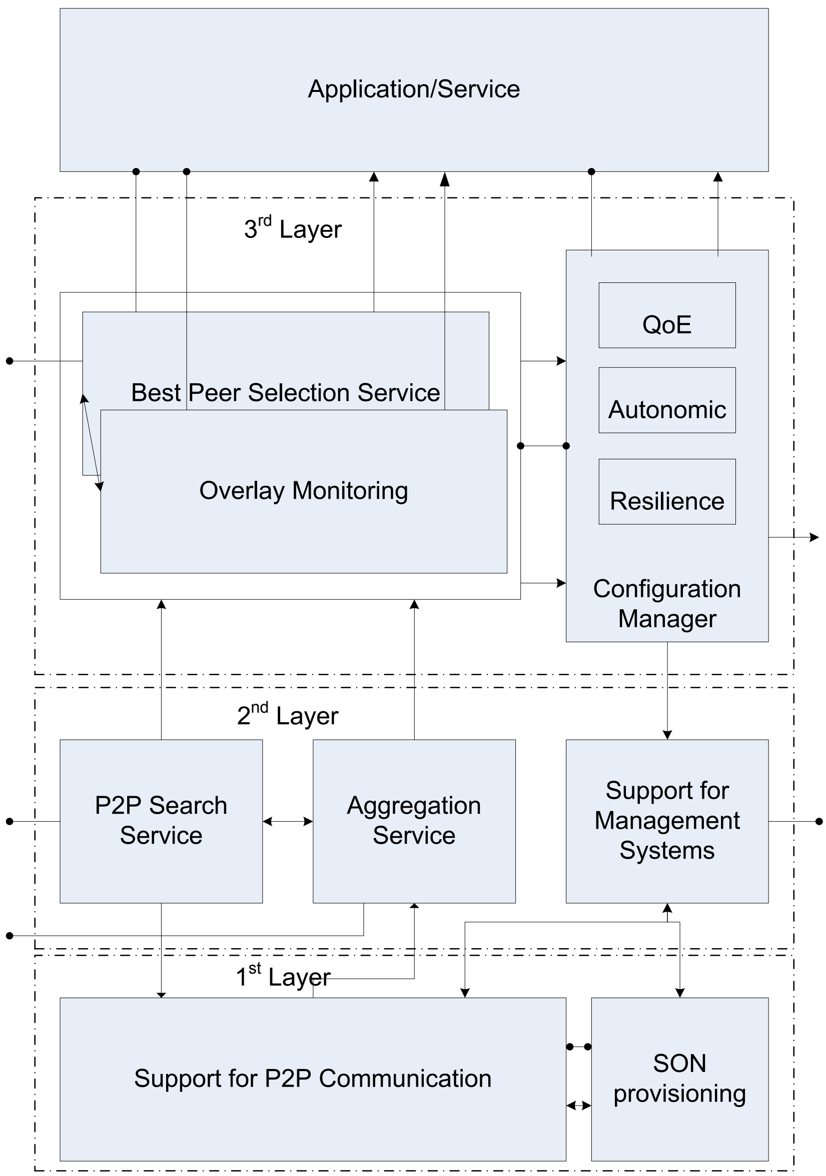


Figure 3.2: The OMAN Architecture

responsible for handling particular aspects of the architecture, ranging from the P2P SON formation to specialized support services.

Thus, according to Figure 3.2, at layer 1 the mechanisms for P2P communication will support the whole architecture. At this layer, both the modules and their services should be common to every OMAN node participating in the P2P SON. This module will be responsible for managing the aspects related to the maintenance of the P2P overlay, including the nodes' actions of joining and leaving. This layer is also responsible for handling the P2P SON provisioning. This includes the establishment and enforcing of the Service Level Agreements (SLA) and the administrative management of the service providers in the SON's group.

Some basic framework services are provided at Layer 2. This is the case of the services responsible for management and interfacing with external or legacy management systems. A specialized searching functionality is also provided at this layer. Thus, customers/consumers and third-party service providers can utilize the search functionalities at this layer to find whatsoever service they need (since it is published), including the following: overlay management services; particular specialist services; infrastructural services; and service components for use in composing new services. A particular service designed to cope with this searching functionality is the Aggregation Service (AgS) [Fiorese 10d, Fiorese 10a].

Specialized services in terms of overlay management and services improvement will take place at Layer 3. The Overlay Monitoring (OM) module collects information about the state of the overlay regarding its resources and execution conditions. The Best Peer Selection Service (BPSS) is a service that informs a set of best peers to position a service according a particular application metric. In its way, OM helps compatible OMAN applications and services on the use of the monitored P2P SON. Moreover, for external entities, BPSS provides the ability to choose, amongst the services available on the P2P SON, the one that fits the required parameters. The OM and BPSS support the Configuration Manager (CM) module. The CM has sub-modules: Resilience, QoE and Autonomic that cope with the dynamic aspects of the P2P SON management. The Resilience sub-module will be responsible for instantiating and looking for alternative nodes to execute the operations in the case of failure or disconnection of a particular node in the P2P overlay that supports the SON. The QoE sub-module will offer an interface to obtain information about the experience from the users, concerning the services provided by the whole system. The Autonomic sub-module is intended to provide some self-* capabilities [Hariri 06] for the OMAN architecture, in particular self-configuration, based on an initial management policy received from the controllers, i.e. from the federated providers that form the P2P SON, and on the information collected by the OM module.

This architectural design comprises the expected functionalities in a system that is fully implemented accordingly OMAN specifications. For the purposes of this thesis, the following modules and submodules are assumed to be present and implemented by third-party partners, or dependent on the choices of the P2P SON participants:

- Support for P2P Communication.
- SON provisioning.
- Support for Legacy Management Systems.
- Overlay Monitoring.
- Configuration Manager, and its submodules/services.

3.3 OMAN Components

The following subsections describe the functionalities and interfaces supported by the OMAN modules and their services. This description is conducted on a layer by layer basis in order to be easily followed in accordance with the Figure 3.2.

3.3.1 First Layer

This layer comprises the P2P SON itself. This layer is mandatory in the OMAN architecture. Therefore, every peer involved in the framework that implements OMAN must have the Support for P2P Communication and the SON provisioning modules/services.

3.3.1.1 Support for P2P Communication

This module comprises the necessary algorithms and strategies to implement a P2P overlay network. This means that it is responsible for providing the P2P interconnection between the service providers that form the P2P SON. The service providers are responsible for choosing the P2P overlay network that implements this service. To achieve this, they must agree on a set of parameters such as what kind of P2P topology and protocols to use. In order to cope with these decisions, external tools can be used by the service providers. The service providers can choose among several existing P2P overlay networks or implement and use a new one. However, they must take into account the most generic set of premises when deciding on the best way to satisfy the service provider's needs.

A particular case emerges when a nationwide service provider decides to form its own P2P overlay network. This decision can be made, for instance, considering the internal business policies that say to offer services in large geographical areas. Thus, this big service provider starts operating its P2P overlay network as the infrastructure for offering its services. After a period of time, it can allow its P2P overlay network to form a P2P SON with other service providers using for that a particular business model. It can thus be assumed that, in this case, (depending on the conditions), the other service providers will agree on the previous choice of P2P overlay network.

Several P2P overlay networks are available and can fulfil the purpose of supporting the P2P communication. As examples of generic P2P overlay networks, the following can be cited, among others: CHORD [Stoica 01], Content Addressable Network (CAN) [Ratnasamy 01], and, Chameleon [Brown 09]. Nonetheless, P2P service/applications can be offered that are not aware of the chosen P2P overlay network. In this case, this P2P application/service has to implement specific strategies for routing its information, including how to choose and manage its neighbouring peers.

3.3.1.2 SON provisioning

This module takes care of negotiating, establishing and enforcing the Service Level Agreement (SLA) among the service providers that compose the P2P SON. Among other responsibilities, this module should guarantee, the financial sustainability of the P2P SON. To accomplish that, a possible way is the use of billing policies based on the data content exchanged or shared and the bandwidth used for. This strategy can be used to share the operating costs between the participants.

These policies must be negotiated and stated in an agreed SLA. This is a task that has to be performed at the time when the P2P SON is formed. In addition, this module is required when a new service provider joins the P2P SON that has been previously established. In this case, depending on the particular P2P SON policies, new SLAs must be set up with the new service provider neighbours or with the entire set of P2P SON participants. Another task of the SON provisioning module, during the operating phase, is the enforcement of the SLA agreement. This task requires constant monitoring and testing. Due to the controversial nature of this task, it should be undertaken by a trustworthy third party entity.

However, when a single service provider operates its own P2P SON, the SLA negotiation and enforcement phases between the service provider participants are not needed. Despite this, SLAs are still needed between this provider and the communication providers responsible for the interconnection of its remote peers. The same applies during the enforcement phase. Thus, the primary attribute of the SON provisioning module is to assure communication between service providers and its communication providers through the SLA enforcements. The secondary feature is to do the same between the service providers.

Hence, the SON provisioning module is an architectural skeleton of strategies, algorithms and policies facing the problem of supporting the formation and operation of a P2P SON, in the context of a consortium of service providers.

3.3.2 Second Layer

This layer provides searching functionalities for the OMAN architecture.

3.3.2.1 P2P Search Service (P2PSS)

This module is in fact the basic interface that is employed to access the SON P2P overlay network formed by service providers. The P2PSS service is dependent on the P2P overlay network supported by the OMAN first layer through the Support for P2P Communication module. Therefore, P2PSS provides the P2P entry point for interconnection and sharing mechanisms that allow service providers to store, search, and withdraw information when providing services. The name P2P Search Service was chosen because it reports to the main service functionality.

Although the OMAN first layer provides the P2P communication, it does not make available standardized interfaces for storing, searching, and withdrawing native P2P algorithms. The reason for this is that at first layer, the service providers can choose any kind of P2P overlay network to support the SON. Thus, in the case of a SON supported by a structured P2P overlay network, the access to these methods, although producing the same effects, is different from a unstructured one. For instance, if the SON is formed by means of the CHORD P2P overlay network, both the peers' ID and the application/service information (e.g. service profiles published by service providers), must be keyed into a flat key space identifier of 128 bits. The ID or key that is generated is used to find out which peer in the overlay will store this information and on the same principle, this information can be recovered by just informing its key. A similar approach is adopted by non-DHT overlay networks. Nevertheless, the common concept involved here is the **key** which is the information that is handled by the P2P overlay network in several processes, whose routing is the core element. Thus, in this module, the use of a common Key Based Routing (KBR) API [Dabek 03] is welcome.

This approach can be adopted to standardize the access to storing, searching and withdrawing services for third-party service developers. In this case, services that make use of the established P2P SON for their operations (e.g., P2P services), can be implemented without the knowledge of what kind of P2P overlay network has been used at the first layer. This approach also caters for the needs of external OMAN entities. Thus, the offered external access interface (as can be seen in Figure 3.2) is used independently of the OMAN first layer.

To sum up, this module provides an abstraction for the process of searching for information in the OMAN architecture. This abstraction encapsulates details and hides the native P2P searching mechanisms used at the first layer by the Supporting P2P Communication module.

3.3.2.2 Aggregation Service (AgS)

The AgS is a higher level search service. In an architecture designed for offering services in a P2P environment, the searching process is critical. The finding of a service and the speed of this discovery can increase the income and satisfaction of the users. In view of this, even though a well-designed and fast P2P overlay network is used at the first layer (which depends on the choices made by the service providers), OMAN provides a search approach as a specialized service that goes beyond the general purpose search methods offered at its first layer. Thus, the AgS is a specialized search service, whose purpose is to enhance the searching of services and service components offered in the P2P SON that is maintained by a consortium of service providers that implements OMAN.

AgS comprises the aggregation of particular information about the published services to bring about an improvement in the searching. The **aggregation** is the key concept for this module/service. In the context of OMAN, aggregation is a process that, in its first phase, involves the publishing of service profiles. In its second phase, the caching of published service profiles takes place. This latter phase occurs during the search process while the former is initiated as an independent event by the service provider. The aggregation process depends on trust between the service providers regarding the information that is made available at the service profiles. OMAN handles this issue by separating the process of offering services by mean of the service profiles from the question of gaining access to these services. This means the service profile only conveys enough information to announce the service and state how it should be accessed. Thus, the access is obtained outside OMAN through the interfaces stated in the service profiles. This two-tiered approach allows service providers to restrict sensitive configuration information regarding details related to the service, such as internal domain topology, communication technology, filters and personalization configuration.

The peers that form a part of the P2P SON play particular roles when implementing AgS. On the one hand, the peers that execute services and service components are responsible for producing the service profiles and announcing (publishing) them. On the other hand, there are other peers that receive these announcements, cache them and form a second-tier overlay where the searches are performed. Therefore, each service offering is published and aggregated in special peers at the P2P SON so that it can be found easier and faster.

To sum up, AgS serves as a second-tier for the P2P Search Service, and increases the response time for a particular kind of search (service and service components). In the validation framework for this thesis, a particular P2P SON uses the AgS to implement the search for service components that can be assembled in new services. Thus, it is worth mentioning that the search results from AgS can be used by external service composition platforms. The adopted strategy for AgS *per se*, reduces the time needed for searching, and in addition, the use of the replication/caching of the search results ensures a better performance. The AgS is discussed in detail in Chapter 4.

3.3.2.3 Support for Management Systems

The operation of a P2P SON as an infrastructure for services offering is a sensible system. This operation may include offering, searching and provisioning. Service providers have different equipment, policies and expectations. They manage their internal domains by relying on different solutions including third-party or legacy management systems. In particular, the management systems are responsible for keeping things running, which allows the business to move forward. However, the wide range of management solutions can lead to heterogeneity problems when several service providers are joined together to form a P2P SON.

OMAN proposes the use of a module to deal with these issues. The Support for Management Systems can offer legacy management functionalities by providing an interface for standard management systems. This module can exchange information with the SON provisioning module to allow automatic SLA negotiation. In effect, this module can influence the decision of what kind of P2P overlay network will be used in the module Support for P2P Communication. In this case, legacy or specific tailored negotiation and voting systems can be used by the P2P SON by means of the interface provided by this module. Furthermore, third party software operators can use this module to offer network management services to the P2P SON.

This module can be adapted and improved in accordance with the needs of the P2P SON operators (service providers). In the best interests of P2P SON, this module should be deployed by reliable third-party operators. This approach can overcome the concerns between service providers regarding the matter of trust.

3.3.3 Third Layer

This section presents high level functionalities comprising the P2P SON operations and support for services offering.

3.3.3.1 Overlay Monitoring (OM)

The Overlay Monitoring is responsible for capturing and offering of data related to the P2P SON for the applications and services running on the P2P SON. This module collects this information through monitoring peers (using pooling or publish/subscribe paradigm, depending on the information being monitored). The applications can use these data to improve their routing, storing, and processing capabilities. The Overlay Monitoring module is closely bonded to the Configuration Manager module since the application or service triggers it to adapt the application in accordance with its needs.

The approach used in the design of the Overlay Monitoring module is as follows. Each node belonging to the P2P SON should report the monitored information in a time period interval (or in the case of publish/subscribe paradigm, when the information is ready for this) to a specialized node in the P2P SON. These nodes must be chosen by following certain criteria, which can be, for instance, how long the nodes are alive in the P2P SON. These nodes also are responsible for providing the necessary monitored information to its partner module, Configuration Manager.

In this way, there is a non-exhaustive list of information that is set out and has to be monitored during the operation of a P2P SON. This list represents a useful body of information for a wide number of applications/services. Thus, the purpose of presenting this collection of information is to comprise general purpose services regarding their performance. However, specialized services that make use of P2P SON can have particular needs that should be elaborated in future enhancements of this module.

Processing Load: This information reveals how idle or busy the node's processor is. It is closely related to the node's load, although this information also takes account of memory and disc usage per process. Hence, the processing power information is appropriate to specialized applications and services that share processing capabilities. This information is kept at an average value between the measurement intervals. The use of this strategy allows high processing power services to avoid choosing a peer that processes data intensively, and relies heavily on calculations, including those with a low memory consumption.

Storage Capacity: How much main or secondary memory space is available for the application or service in the running node? This is what this metric depicts. This information is required for content sharing (especially file sharing), and database-intensive services. This information is also important, to instantiate or clone the same application or service in another node in the P2P SON. This latter functionality is essential in the case of load balancing or fault tolerance and also for P2P applications that face high churn rates.

Bandwidth: Each node involved in the P2P SON has communication interfaces that can be active or otherwise. Each interface provides access to a communication link with a certain available bandwidth, depending on the link utilization which, in turn, depends on the number of sessions handled by the node. The available bandwidth determines how fast the nodes can communicate with each other.

Thus, this information is especially valuable for routing, content delivery and almost any networked operation that an application seeks to perform.

Node Load: The node load depicts how busy the node is as a whole in the P2P SON. This metric not only takes into account the used processor capacity and how many time slices it can attend but also how much memory and disc storage are being used by the node to execute its demands in a time interval. This monitored metric is important for applications that need some kind of processing information about other peers in a session. For example, a service that attempts to encrypt or decode some video traffic in the middle of a transmission could use the node load information to choose which nodes can participate in the forwarding route.

In a shared environment these stated information are critical. Security aspects, comprising the information misuse, adulteration, repudiation and other problems should be handled by well-designed protection mechanisms that needs to be stated in SLAs and enforced by the responsible third-party OMAN module (SON provisioning). These protection mechanisms can work together with the Support for Management Systems, which can adopt a global Policy Based Management (PBM) that must be enforced for all the service provider participants.

3.3.3.2 Best Peer Selection Service (BPSS)

The Best Peer Selection Service (BPSS) is a module that is designed to face the peer selection problem. This problem occurs when an application or service has to make a choice about which of two or more counterparts it should interact with. In the case of P2P applications, this is the standard since an application generates an overlay where a single request can be served by more than one peer. Thus, the choice of which one will deal with the request is a cornerstone problem. In our case, where services offered in a P2P SON can be (but not necessarily are) P2P-like, peer selection is still an issue. Consider the case of a service which is composed of several small service components spread over the P2P SON. Several service providers can offer the same small service components in the P2P SON. Thus, the composer can be interested in the particular components that are closest to him. Moreover, when specialized services (e.g., on demand video streaming), even running in a P2P SON infrastructure, have to deliver their data flow, the choice of which video sources are the best to achieve the agreed QoS for the end user, is also overwhelming.

In fact, the choice of the best peer depends on the particular objective an application has. When performance issues are considered, generally, metrics comprising the capabilities and interconnection of the peers must be taken into account. The BPSS natively offers the topological parameters to the applications/services to be used in the best peer selection process. The topological parameters are the information about the peers geolocation associated with underlying network performance parameters. These parameters allow the use of the Euclidean distance between the peers belonging to the P2P SON as suitable metric to ensure best peer selection to a wide class of application/services.

All in all, in order to use this module an application or service must specify the minimum number (greater than one) of best peers it needs. Among the various reasons for having this exigency, it is useful for providing fault-tolerance to situations where the first best peer selected is unavailable for some reason at the time of the request.

In certain cases, the BPSS can be used, along with the Overlay Monitoring (OM) module, to provide an answer with a bunch of peers in the P2P SON that correspond to particular metrics. When the only metric desired is the distance between the node where the service is running and other particular nodes, the BPSS will be used by itself.

3.3.3.3 Configuration Manager (CM)

The Configuration Manager (CM) module is responsible for adapting the P2P SON to the conditions required by the applications or services, as well as for controlling the underlay in order to get the desired behaviour for the application. The CM is also responsible for the following: receiving the configurations that will be applied to the overlay; creating groups of peers to execute particular management operations; controlling the authorizations for applying configurations; and with the help of the

BPSS module it is responsible for applying the configuration to the appropriate peer(s). It can also offer information about the result of the configuration or delegate the task of configuring to another peer. The CM will use the information collected by the OM module autonomously to adjust the storing, sharing, and other capabilities.

CM can use the Support for Management Systems to control the underlay. The decisions the CM should make to control the underlay depend on the overall policy of the P2P SON (based on SLAs agreed between operators or providers), and autonomously by the metrics monitored by the Overlay Monitoring (OM) module. The Autonomic sub-module deals with this aspect.

The information monitored by the OM module also influences the Resilience sub-module. This last module is responsible for keeping the P2P SON working when disruptive abnormal conditions occur. An example of its operation appears when a peer in the chosen P2P network overlay does not respond to the requirements and its application or service must be cloned and started again in another peer. The task of cloning and starting the "new" application is one of the responsibilities of this module. The choice of the best peer to start the "new" application can be made by using the BPSS module.

When there is a constant need to enhance applications and services, the user's general opinion about them is required even more. The Quality of Experience (QoE) is the concept that is tailored to capture this notion about the systems from the end users. The CM module is also responsible for implementing this feedback mechanism through the QoE sub-module.

The capacity of analyzing the initial management policy and automatically adapting the P2P SON is executed by the sub-module called Autonomic. The self-configuration is one of a series of self-* characteristics [Hariri 06] designed to ensure the self-management concept. At first glance, the P2P SON adaptation (which is based on an initial policy integrated with the information collected by the OM module) can also support the Resilience sub-module that executes some self-healing functions.

To fulfil its responsibilities, the Configuration Manager needs the cooperation of the BPSS and the OM modules. Hence, the nodes that belong to the P2P SON and implement the Configuration Manager also have to implement the other modules of the third layer of our architecture.

3.4 Summary

In this chapter, an attempt has been made to provide an overview of OMAN. This chapter has shown the three layers that OMAN is composed of. At the first layer, OMAN depicts basic P2P SON composition functionalities and it is here that the choice of which P2P overlay network will be used, along with the mechanisms used for its provisioning, is made. At the second layer, basic and advanced modules for search service and service components are provided. The aggregation mechanism used to improve the search services is implemented at this layer. In addition, an interface is designed at the second layer for the integration of management systems. Finally, at the third layer the main contribution is the selection of the best peers though the overlay monitoring and configuration procedures are also addressed.

On the basis of these layers, OMAN handles offers of services in a P2P SON environment. Service providers form the P2P SON supported by OMAN through a process of P2P joining. Each service provider makes available the nodes that execute its services and service components. Services and service components are represented as service profiles which are XML files that describe the main technical features of a service, as well as the business information comprising its utilization. Service providers publish service profiles at other special purpose SON peers and make them available to their customers (e.g. other service providers, and end users).

Chapter 4

Aggregation Service - AgS

Services and service components are becoming the main elements in interactions among service providers and service consumers. These services might include content (e.g. finding vendors offering a specific movie, 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 content) or complementary services (electronic billing and/or payment systems; multi-session controllers for video-conferencing sessions, etc.). Regardless of what kind of services, they might be dispersed anywhere among providers on the big cloud the Internet currently represents.

Since they are the main elements, services are in the business core and require special handling in an architecture such as OMAN. This handling entails strategies and mechanisms that allow maximum exposure of services to potential clients, (i.e. the providers' customers), and thus make them more likely to be used and lead to an increase of income to their providers. The establishment of a P2P SON among the service providers can constitute a first step in this direction. This allows service advertisements to spread beyond single service provider domain boundaries with the costs being shared with other service providers.

However, despite the forming of P2P SON, services still have to be searched. The intrinsic P2P searching mechanisms are well suited to this task. However, in foreseen scenarios such as the Future Internet (that include services negotiated in a huge market and even more virtualized environments such as Clouds with Infrastructure as a Service (IaaS) and Software as a Service (SaaS) [Foster 08, Youseff 08]), where the number of services and service components can easily scale to considerable numbers, more specialized and efficient searching mechanisms are required.

The developed Aggregation Service (AgS) aims to supply the optimized solution for searching services that are lacking in a P2P SON. In short, AgS takes advantage of the service providers' self-organization in the P2P SON to explore the concentration of the published service profiles in a second overlay-tier to improve the searching process. This is carried out by specific purpose messages that are used for the interaction between the overlay-tiers and results in an optimized mechanism for searching services.

This chapter sets out the specifications and evaluation of an Aggregation Service that is suited to the problem of optimising the search for service and service components, that are integrated into the general conception of the OMAN architecture. It is structured as follows: Section 4.1 describes the roles of the parties involved, their interactions and the type of information exchanged. These parties include service providers, end users, peers and services interacting in an appropriate architecture. The AgS architecture is outlined and discussed in Section 4.2. The AgS architecture aims to present the components and functionalities designed to optimize the search of services in a P2P SON. In addition, Section 4.3 analyses the AgS evaluations and the obtained results. Finally, Section 4.4 summarizes the chapter, highlights the key findings and presents open issues.

4.1 AgS Model

A logical model of the AgS represents the entities that are required for coping with the designed functionalities to ensure the search of services optimization process. The AgS model is an important means of understanding the roles of the parties and entities involved, their interactions, and the type of information exchanged between them in a service searching process in the P2P SON. Some of the entities, which are depicted in the AgS model, comprise the formation of the P2P SON and play specific roles.

This is the case of the peers involved in the process. The peers that form the AgS P2P overlay are called **aggregation peers** (or **AgS peers**). If the service providers choose to make available their own aggregation peers, these peers should be chosen from among the peers that compose the P2P SON. Thus, aggregation peers (or AgS peers) are specialized SON peers.

The peers that compose the P2P SON are called **SON peers**. These peers form the overlay network where the real services and service components are actually running. The SON peers make the services indirectly available to external entities (such as service composers and aggregators) which are likely to be located in other network domains. The administration and management interfaces of the services are used to put this into effect. These are the interfaces that are encapsulated in the service profile already mentioned (see Section 3.1.1). 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 at the same domain as the SON peers or, in some cases, at different domains.

The AgS operates in accordance with the model displayed in Figure 4.1. This Figure depicts the class diagram of the AgS model in Unified Modeling Language (UML) [Rumbaugh 05] but with the details on attributes and operations left out. The model consists of 5 classes divided into 4 roles (subjects), i.e. *Customer, Service Provider, AgS Peer,* and *SON peer* and one data class (objects), i.e. *Service or Service Component*. Each class and its relationships with other classes in the model are described in detail in the next Subsections.

4.1.1 AgS Model Description

The Service or Service Component object is the key piece in the AgS model. It is a part of the interactions that occur between every subject in the model. It represents the Service or Service Component that actually executes on the SON peer and is published in an aggregation peer. Thus, Service or Service Components are represented by service profiles according the specifications of the Service Providers, which are their owners. The commercial exploitation of Services and Service Components depend

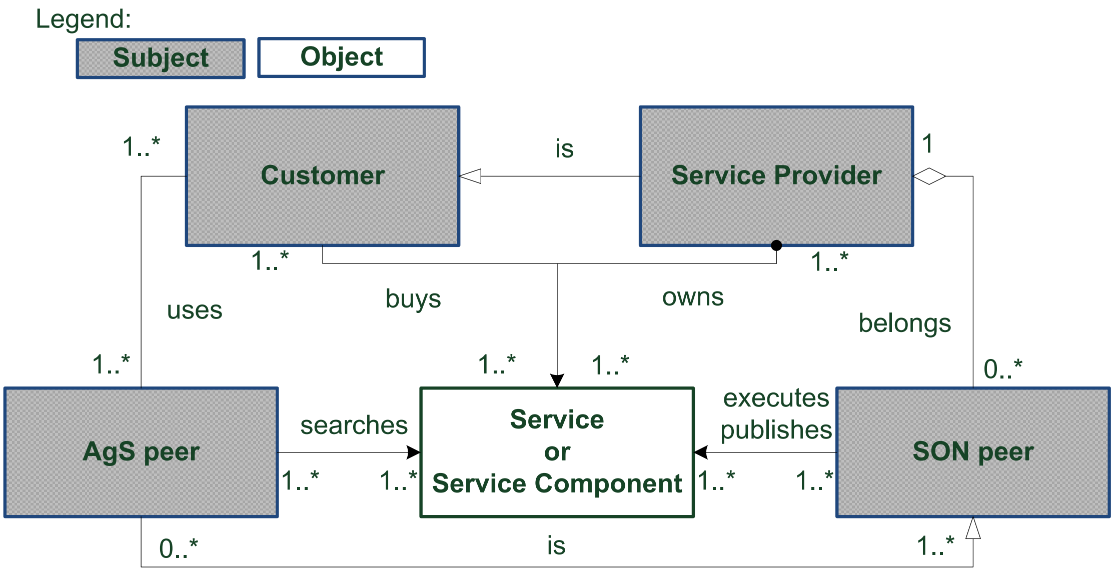


Figure 4.1: Aggregation Service Model

on the business model that is undertaken by the Service Providers. A commercial relationship will take place between the data class Service and its customer and also its owner in a way that is independent of the current business model. In the AgS Model, this relationship is represented between the Customer and Service or Service Component classes, where the former buys the latter.

Service Providers are the owners of Service and Service Components. They offer their services using the P2P SON in order to achieve broader consumer markets beyond their own network domains. Following this rationale, Service Providers can be Customers as well. This means they can act as third-party consumers of the service components of other service providers. Nonetheless, before reaching this stage, first of all, the service or service component has to be searched and found. According to the AgS model, the customer, which might 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.

Aggregation peers, or simply AgS peers, represent the specialized SON peers, that are chosen from among the peers that compose the P2P SON in order to form the AgS P2P overlay. Each SON peer plays a double role. They execute the services and, also publish references to the available services at several AgS peers to optimize the search of services. A single service offering can be spread over multiple AgS peers to allow for some redundancy and 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.

4.2 AgS Architecture

The core of the Aggregation Service is based on the service and service components profile aggregation. This aggregation is implemented as the concentration of these files in the aggregation peers. Nevertheless, an interaction involving SON and aggregation peers is necessary to achieve this kind of concentration. This interaction comprises the publishing of the service profiles offered, the reception of the publishing messages and the organized storage of these profiles. This procedure seeks to improve the searching process and for this reason, the developed AgS architecture is split into two layers as follows:

Support : At this level, the SON peers, that have been made available by the P2P SON participant service providers, will support the aggregation process by publishing the service profiles. These service profiles must be generated by the service providers in accordance with their internal policies.

Aggregation : This level is responsible for the storage of the service profiles and ultimate recovery, i.e. when the searching has been performed. This happens at the aggregation peers.

The interactions between these two layers, the roles of the AgS components and the entire working of the AgS architecture are examined in the next Sections.

4.2.1 AgS Description

The Aggregation Service (AgS) is an unstructured P2P overlay, which means there is no tight coupling between the AgS overlay topology and the exact location/placement of the information (aggregated services). It executes on top of a P2P SON that is created by service providers. Ultimately, AgS is composed of peers that belong to these service providers that are interested in advertising their services. These peers are chosen from those (SON peers) in the P2P SON, and they form a new overlay where they play the roles already explained and, for that reason, are called aggregation peers.

The purpose of the AgS is to improve the search for service and service components. To achieve that goal, aggregation is the chosen solution. In this case, aggregation is achieved by concentrating the service offerings on the aggregation peers, to improve and optimize the search process. The architectural design of AgS is depicted in Figure 4.2.

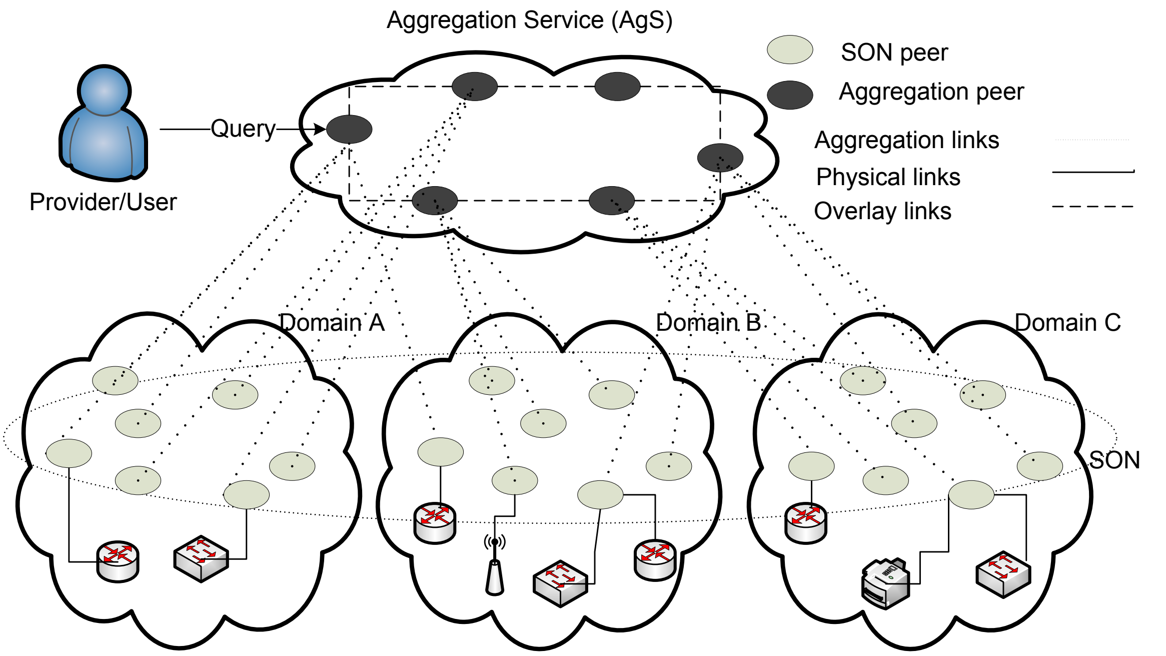


Figure 4.2: Aggregation Service Architecture

Figure 4.2 shows that AgS consists of a P2P overlay using a logical ring topology where the exact location of the aggregated services is unknown. However, despite the depicted, other P2P topologies can be easily applied. Nevertheless, it is worth mentioning that the aggregation peers keep an internal cache for storing service profiles, and also maintain pointers for successor and predecessor peers in order to form the logical ring topology. Figure 4.2 also shows SON peers that belong to particular administrative domains that announce (publish) their services and service components to the aggregation peers.

These peers belong to the service providers that offer services/components. Unlike the aggregation peers, SON peers participate in the AgS by making the administration and management interfaces of the services indirectly available to external entities (such as service aggregators), which are probably located in different domains. These 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 (i.e. the service profile) 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.

The AgS executes the search for services and service components by means of keywords. Thus, it is expected that the keyword used in the service profile as well as in the search, will be representative of the desired service. Besides the keyword, the service profile should also keep information about the administration and management interfaces so that services can be accessed.

Each peer involved in the AgS can take care of several service offerings. A single service offering can be spread over multiple AgS peers which allows some redundancy and enables it to overcome churn.

Taking into account the way the AgS works, the P2P SON underlay can be composed of either an unstructured or structured overlay. Since the underlay SON is a means of organizing the participant service providers, the AgS takes on the role of booster for the search and of contact interface for the users (e.g. the providers of new composed services and end users). Even if the SON underlay is not composed of P2P bindings, the AgS can still be used.

4.2.2 AgS Utilization

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 enables the publish and search functionality to be split off from the services and service components management functions that are 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 making available (publishing) a previously selected set of interfaces for services and service components. In addition, this two-tier approach allows answers to be obtained more quickly at a low computational cost through the aggregation of the services and service component offerings in the higher tier. Moreover, as it is the service providers that maintain this aggregation tier, this ensures that there is cooperation between them when they offer their services. This approach is suited to high dynamic and large-scale multi-domain environments.

Therefore, 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.

4.2.2.1 Usage Scenario

AgS is used by service providers as an infrastructure tool. In contrast, service provider customers use this tool to achieve their business goals by means of using the found service. It is worth mentioning that an AgS client can also be a service provider that forms a part of the AgS itself. Of course, an end user can also use the AgS. The way the system operates, however, is the same for both kinds of users.

In addition, the way the AgS is used by the provider of a composed service, for instance, is very simple. This provider uses the AgS to search the SON peers where the required service components are available. Taking Figure 4.2 as a reference-point, let's consider that a service provider from Domain A subcontracts services located in Domains B and C (e.g. a connectivity service with QoS guarantees and access to multimedia content). In this scenario, the AgS ensures that the SON peer reference for this service can be recovered. Further access to the service interfaces (contracting, monitoring and life cycle management, etc.) is provided by the SON peer that represents that service component. Interaction between the service contractor and the SON peer is conducted outside of the AgS and the OMAN architecture.

A specific usage scenario for a third-party service provider or end user might be the on-demand stream-like movie delivery system with its associated service components. With this example, these service components might include content (e.g. finding vendors offering a specific movie end-user wants to see; and finding dubbing or subtitling services for that same movie), connectivity (e.g. interconnection links with QoS and security requirements between the consumer and the origin of the content) and complementary services (electronic billing and/or payment systems; multi-session controllers for video-conferencing sessions, etc.). These service components can be found on different service providers belonging to several particular domains (e.g. B or C, as shown in Figure 4.2), and once composed, they can be offered as a complete third-party service.

4.2.3 AgS Operations

The AgS is based on a number of operations. These operations cope with the formation, operation and finalization of the AgS P2P overlay and are called Join, Publish, Search and Leave. The execution of these operations depends on the occurrence of particular events. This is the case, for instance, when a new aggregation peer joins the AgS, or when an aggregation peer starts to looking for references of SON peers that can execute some particular service or service component.

In order to accomplish the AgS goals, messages are exchanged between aggregation peers regarding particular operations. The set of these messages constitutes a protocol named AgS-P. The AgS-P payload message formats and objects are defined in Appendix B. Each message associated with the AgS operations is presented and discussed in the perspective of the operations as follows:

4.2.3.1 Join Operation

This operation is responsible for the AgS P2P overlay formation. As already mentioned, the participant AgS peers form a logical ring. The Join operation faces the double linking arrangement between AgS peers and needs a particular kind of message to be carried out. The JoinMessage is suited to this task, and can be described as follows:

JoinMessage : This message is launched by the Join operation and starts the joining process of a new aggregation node into the AgS. Taking advantage of the simplicity of the logical ring topology, an AgS joining aggregation peer sends this message to its successor and predecessor in the ring. These messages carry the peer identifier as a message field representing the predecessor or successor of the receiver peer. This peer ID is used by the JoinMessage receiver to update the corresponding pointer to its successor or predecessor in the ring. The choice of what will be the successor and predecessor of a joining AgS peer is based on the values of the peer ID. Hence, a joining AgS peer will be located between the immediately greater peer ID, which will be its successor, and the immediately smaller peer ID, which will be its predecessor. In this way, the ring is formed in accordance with the increasing values of the peer ID.

Figure 4.3 shows an ingress AgS peer and the JoinMessage participation on the Join operation.

4.2.3.2 Leave Operation

AgS peers need a way for a normal exit from the AgS P2P overlay. This operation is responsible for this and generally takes place when a service provider discontinues its last running service in a SON peer. The Leave Operation is responsible for keeping the logical ring consistent without breaks. Like the Join Operation, the Leave Operation relies on the AgS peers ID to keep this consistency by handling the successor and predecessor links. This is carried out in a similar way to the Join Operation, and a special kind of message is used for this purpose. This message is the LeaveMessage and can be described as follows:

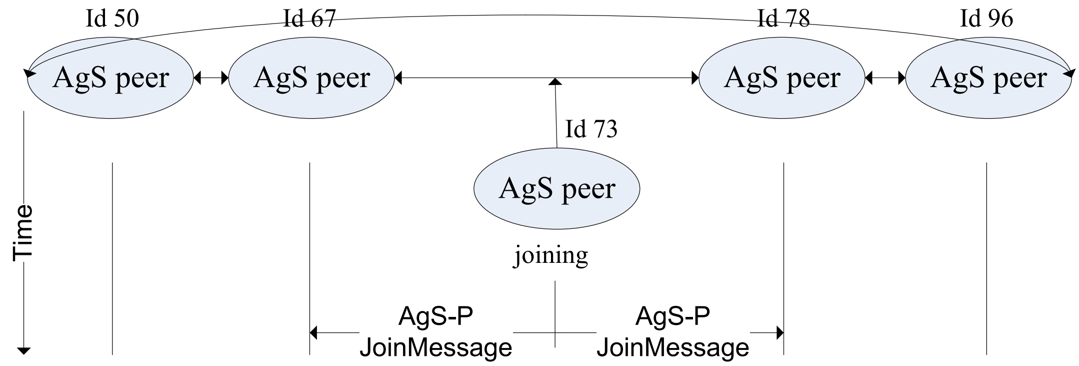


Figure 4.3: Message Sequence for Join Operation

LeaveMessage : When an aggregation peer leaves the AgS overlay in a regular way, the operation Leave is triggered and the LeaveMessage is sent. As in the case of the Join operation, for each leaving peer, this message is sent twice in order to update the predecessor link of the current leaving peer's successor, and the successor link of the current leaving peer's predecessor. To accomplish that, the LeaveMessage sent to the current leaving peer's successor carries the peer ID of its predecessor. Moreover, the LeaveMessage sent to the current leaving peer's predecessor carries the peer ID of its successor.

Figure 4.4 depicts the Leave operation and the LeaveMessage participation on the sequence of events.

4.2.3.3 Publish Operation

This operation is responsible for feeding AgS with service profiles. In spite of the previous and remainder operations, Publish Operation is started by SON peers. Nevertheless, it only occurs when a SON peer wants to advertise a service or service component that it is executing. This means the SON peer must have a valid service profile and be authorized by the service provider to publish it. This publishing is undertaken by SON peers through PublishMessages sent to a group of AgS peers. This message and its interaction can be described as follows:

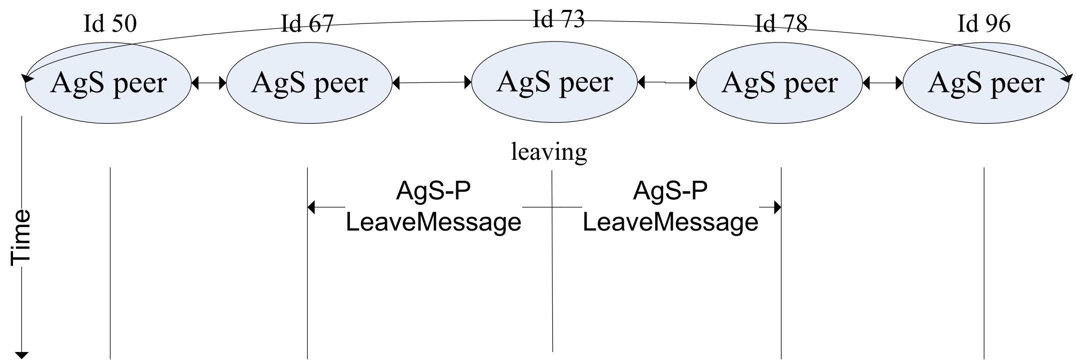


Figure 4.4: Message Sequence for Leave Operation

PublishMessage : The Publish operation is executed by means of the PublishMessage. This message is sent by the SON peers as a way of announcing the services they are executing and authorized to advertise. It is then sent to a set of AgS peers in a unicast manner since the group of AgS peers can be different for each SON peer. This set of AgS peers is randomly chosen from among the active AgS peers. This strategy enables a high publishing distribution that can reduce the effects of the churn of the AgS peers and lead to faster searching as well. This increased speed can be achieved because when there are more copies of the service profiles, the SearchMessage does not have to travel so many hops to reach its goal. The payload of the PublishMessage carries the service profile for a service or service component.

Figure 4.5 shows the Publish operation that is possible through the interaction between SON peers and AgS peers. This interaction occurs when SON peers execute the service advertisement by sending the PublishMessage to its set of AgS peers. Figure 4.5 shows that the set of AgS peers for the SON peer A is AgS peers ID 50 and 96, for the SON peer C the set of AgS peers is AgS peers ID 96, 78 and 50. In spite of this, Figure 4.5 also shows that SON peer B is advertising any service at the snapshot taken.

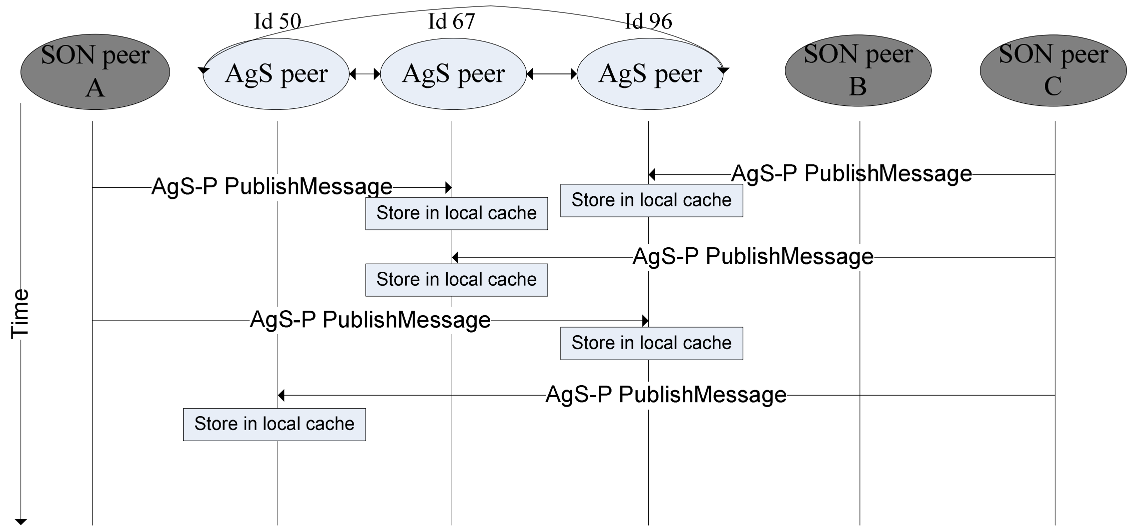


Figure 4.5: Message Sequence for the Publish Operation

4.2.3.4 Search Operation

In order to execute the Search Operation, AgS-P provides two standard messages. These messages are responsible for walking through the AgS overlay requesting for a particular service or service component, as well as to report the result(s) to the requesting AgS peer. This process involves visit neighbouring peers and asking them to check their local caches and forward the requesting message in case they do not find the requested service. For the case they find the requested service, a new message has to be generated and sent to the requesting AgS peer with the result(s). These messages are described as follows:

SearchMessage : The SearchMessage is sent on the occurrence of a search request in an AgS peer. Customers who employ the Aggregation Service use this interface to execute the search for a service. The SearchMessage is sent by an AgS peer to its successor in the AgS ring and is forwarded in a clockwise manner until it reaches an AgS peer that keeps the searched service profile in its local cache. This message carries the keyword representing the service or service component searched, as well as the peer ID of the requesting AgS peer. The AgS framework is also prepared to use this message as a collector of all the SON peer references that execute the desired service or service component. When working in this mode, this message also carries a list of SON peer references and the forwarding only stops when the SearchMessage reaches the requesting AgS peer.

SearchReplyMessage : This message is the message used as a response to the SearchMessage. It is triggered by an AgS peer when it receives a SearchMessage and after checking its local cache. If there is a service profile for the service being searched, which is associated with a SON peer reference, then the SearchReplyMessage is sent straight on to the requesting AgS peer. This message carries a list of SON peer references in its payload, which can be fulfiled with one or more SON peer references. The reason for this is that an AgS peer can keep in its local cache the references for several different SON peers that are running the same service or service components. It is worth remembering here that service providers from different domains can offer the same services and publish them at the same AgS peer.

Figure 4.6 depicts the SearchMessage forwarding process which occurs when a SearchMessage is received by an AgS peer. After checking its local cache, this AgS peer decides if it will forward the SearchMessage or respond to the request. This forwarding process continues until the searched/requested service or service component (service profile) is found in an AgS peer's local cache. When this occurs, a SearchReplyMessage is generated and sent back to the requesting AgS peer carrying the service profile and the associated reference for SON peer responsible for the profile.

Table 4.1 summarizes the AgS operations and the corresponding messages exchanged among peers.

4.3 AgS Evaluation

In this section, the AgS proposal [Fiorese 10d, Fiorese 10a] is evaluated to determine its efficiency in providing a framework for storing and searching service and service component references in a large-scale multi-domain environment formed by multiple

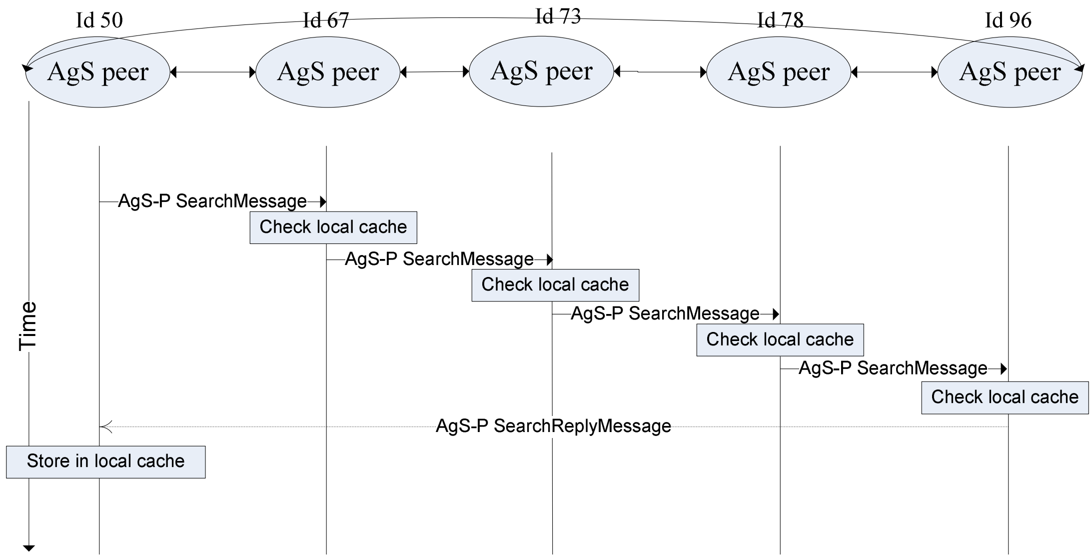


Figure 4.6: Message Sequence for the Search Operation

Table 4.1: 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.
Search	Look for a peer that provides a particular service/service component.	aggregation peer	SearchMessage 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 SearchReplyMessage containing it, is then created. This message is directly transmitted to the AgS peer that triggers the Search 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.

service providers (P2P SON). Several different features have been evaluated with regard to performance and scalability, comprising response time and average path length metrics. Furthermore, a comparative analysis has also been conducted of related work as well as the use of replication/caching as a mechanism to improve searching performance. Additionally, dimensioning of the AgS was also studied and evaluated to assess the efficiency of the searching.

The evaluations involved carrying out several simulations comprising scenarios with multiple service providers spread across multiple administrative domains. In all of these scenarios, service providers cooperate in the SON formation and ultimately in the formation of the Aggregation Service, in compliance with the proposed OMAN architecture.

Having in mind the stated, this Section is organized as follows: Subsection 4.3.1 provides an overview of the objectives and methodology applied to the experiments that were carried out. Subsection 4.3.2 presents and discusses how the experiments were conducted. Finally, Subsections 4.3.3, 4.3.4, and 4.3.5 analyse each of the experiments and its results to evaluate the AgS.

4.3.1 Objectives and Methodology

Several experiments were conducted to determine the effectiveness and performance of AgS, and the main objectives and methodology employed in the simulation experiments are discussed in this Section.

The first group of experiments analyses the replication/caching of the AgS performance with regard to the average path length of searches [Fiorese 09,Fiorese 10d, Fiorese 10a]. The evaluation of proposed experiments involves verifying the Average Path Length (APL) SearchMessage travelled in the AgS in order to find a particular desired service or service component. The number of hops on the average path length was compared in two scenarios, each with two environments. In the first scenario, AgS was used in the experiments but not in the second. The decision about whether or not to use replication/caching for the search results in each of these scenarios depended on the environment. The analysis of the average path length is an important metric that acts as a means of detecting some evidence or clues regarding the overall performance of AgS and whether it should be adopted. For this reason, simulation was carried out both with and without the AgS, and either using or not using replication/caching. Additionally, analysis of the average path length in the environment with AgS provides valuable information about the distribution of the service profiles in the AgS. Therefore, it is reasonable to postulate that the lower the average search path length in AgS, the better the AgS performance.

The second set of tests determines the AgS efficiency with regard to its performance and scalability by using the search response time as the metric [Fiorese 10b,Fiorese 11c]. The objective here is to analyse the impact of AgS on its users by measuring the time they have to wait for an answer to their requests. Therefore, in this context, response time is regarded as the time between when a SearchMessage is sent and the SearchMessageReply reception, and the average of this time is used in the analysis. Two environments are simulated/tested regarding this set of experiments. Initially, the average response time was compared between searching with AgS and searching without AgS, in the P2P SON environment. Furthermore, the same simulations were repeated for a comparison made between AgS and an environment executing gnutella [Ripeanu 02] in its version 6. In addition, the search hit ratio between the compared solutions, i.e. AgS and gnutella v6, was also tested. When analysed together, these two metrics, i.e. the response time and search hit ratio, allow an evaluation to be made of trade-offs eventually existent between performance and the accuracy of the search.

The last set of tests aims to evaluate AgS regarding its utilization by service providers [Fiorese 11b]. This analysis takes into account its search efficiency, and the size required to provide that efficiency. Thus, this set of tests seeks to determine the maximum number of AgS peers (AgS size) that are needed to ensure that the search is more efficient than just using the native P2P SON search mechanisms. In view of this, suitable AgS dimensioning helps service providers to plan their infrastructures and services, thus allowing them to keep their costs under control. The searching response time is used as a metric in this assessment of dimensioning, by comparing them in the following environments: 1) P2P SON and 2) AgS. The assessment also takes into account the maintenance overhead of AgS itself, in order to compare it with the searching response time in the P2P SON.

Table 4.2 provides an overview of the experiments and states the corresponding section where they are discussed in detail.

Table 4.2: Summary of the evaluation steps of the AgS simulation strategy

Experiment identification	**Experiment name**	**Section**
I	AgS Evaluation using Average Path Length and Data Replication	4.3.3
II	AgS Performance and Scalability Evaluation using Response Time	4.3.4
III	AgS Dimensioning Evaluation	4.3.5

4.3.2 Validation Framework

The simulations were performed using the PeerFactSim.KOM simulator [Kovacevic 07]. This simulator is developed in Java and provides a series of interfaces that allow the development of a P2P overlay network from scratch, depending on the case. The PeerFactSim.KOM simulator allows the development and simulation of large-scale P2P networks and overlays. The Object Oriented Programming (OOP) paradigm that was adopted, along with some consolidated design patterns, allows the abstraction of the several levels involved in the development of these overlays. The Factory design pattern [Gamma 94] is used for the instantiation of the components used in the simulation.

The PeerFactSim.KOM has a modular architecture and allows the simulations to be executed by providing an architecture with implementations of several components used in the simulations. This is the case, for instance, of the transport and network layers of the Open Systems Interconnection (OSI) reference model. The implementations of TCP/UDP/IP were particularly used for the validation experiments that were conducted. The next stage in the implementation of a new P2P overlay is the development of the overlay classes that rely on the node behaviour. This behaviour comprises how events are executed, how messages are routed, and so on. Ultimately, applications execute on top of the nodes. The applications are responsible for offering the interface required for the execution of the events. This means, for instance, the application that is running on top of an aggregation node, must implement a method for starting the join operation.

Figure 4.7 shows the architecture of the used simulator, including the aforementioned modules, and highlights the interaction between the module Node and the formation of the overlay. The used simulator allows the implementation of any kind of P2P overlay topology, which depends on the implementation of the routing scheme.

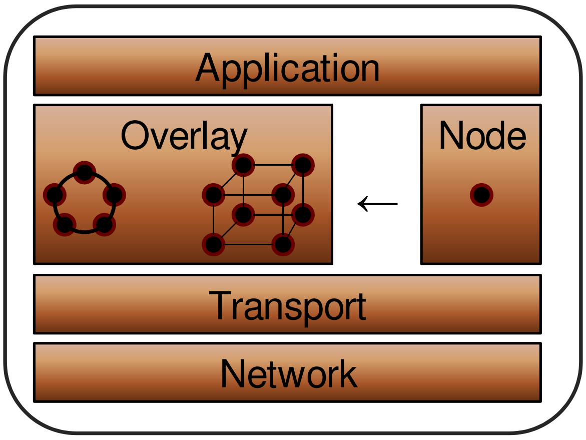


Figure 4.7: Simplified PeerFactSim.KOM Simulator Architecture

The PeerFactSim.KOM is an event-based simulator. This means, its core handles operations based on a discretized time-line. The operations are the implementation of the events. As mentioned earlier, the execution of an operation generally involves the sending or exchanging of messages. The simulator provides interfaces and abstract classes for the construction of operations and messages, as well as other necessary components such as nodes, applications and analysers.

Taking this into account, PeerFactSim.KOM uses a configuration file where the occurrence of operations are specified. This file is called *actions file*. Figure 4.8 shows an excerpt of an actions file used in one of the simulation sets. This fragment shows, as an example of the simulation execution, that 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 only be a SearchReplyMessage to A628 if some other SON peer belonging to Domain 6 had published the S4 service to another aggregation peer that had not yet left the overlay.

The actions file name is in fact an element of the XML main configuration file. This file specifies which modules and factory classes the PeerFactSim.KOM simulator

```
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
```

Figure 4.8: Fragment of the operations configuration file

will use, such as network, transport, nodes, overlay, and application. Furthermore, the nodes configuration is also set up in this file, either as an individual or in a group. Each node or group of nodes can have particular parameters that allow their roles to be differentiated. This is the case for both the nodes (peers) that implement the role of the AgS peer and the nodes that implement the role of the SON peer. In our simulation model, the nodes factory uses a logical parameter to identify which role a node should play and which appropriate node class to instantiate. All in all, these two files (actions file and configuration file) feed the simulator's core so that it can start scheduling the operations, which in their turn, send messages and allow the simulation to be executed. Figure 4.9 shows a fragment of a real main configuration file used in one of the simulations executed.

As a matter of fact, simulator core, i.e. the scheduler, configuration files and implemented classes implementing the whole Aggregation Service, are combined in order to model the simulations that are carried out. Analysers are also implemented that are responsible for filtering and storing the results of the simulations. This means that the results are analysed in a further stage, figures are produced and new clues learned. The simulation model adopted is shown in Figure 4.10.

The specific features of each scenario simulated are discussed when the results are shown, i.e. in Sections 4.3.3, 4.3.4 and 4.3.5.

```xml
<?xml version='1.0' encoding='utf-8'?>
<Configuration>
  <Default>
     <Variable name="size" value="13000"/>
     <Variable name="finishTime" value="3000m"/>
     <Variable name="actions" value="./3000AggNodes/1000-moucaf-actions.dat"/>
     <Variable name="nAggNodes" value="3000"/>
  </Default>
  <SimulatorCore class="de.tud.kom.p2psim.impl.simengine.Simulator" static="getInstance"
  seed="$seed" finishAt="$finishTime"/>
    ...
  <MOVerlayNode class="impl.overlay.MOverlayNodeFactory"/>
  <SONNodesApplication class="impl.application.SNodesApplicationFactory"/>
  <AggregationNodeApplication class="impl.application.ANodesApplicationFactory"
  numberOfAggregationNodes="$nAggNodes"/>
  <HostBuilder class="de.tud.kom.p2psim.impl.scenario.DefaultHostBuilder" experimentSize=
  "$size">
        <Host groupID="A0">
           <Network/>
           <TransLayer/>
        <MOVerlayNode isAggregatorNode="true"/>
           <AggregationNodeApplication/>
        </Host>
        ...
        <Host groupID="M1">
           <Network/>
           <TransLayer/>
           <MOVerlayNode isAggregatorNode="false" aggregationNode="A1719,A854" service=
           "S1@Domain7,S3@Domain7,S5@Domain7"/>
           <SONNodesApplication/>
        </Host>
  </HostBuilder>
  </Configuration>
```

Figure 4.9: Fragment of the main configuration file

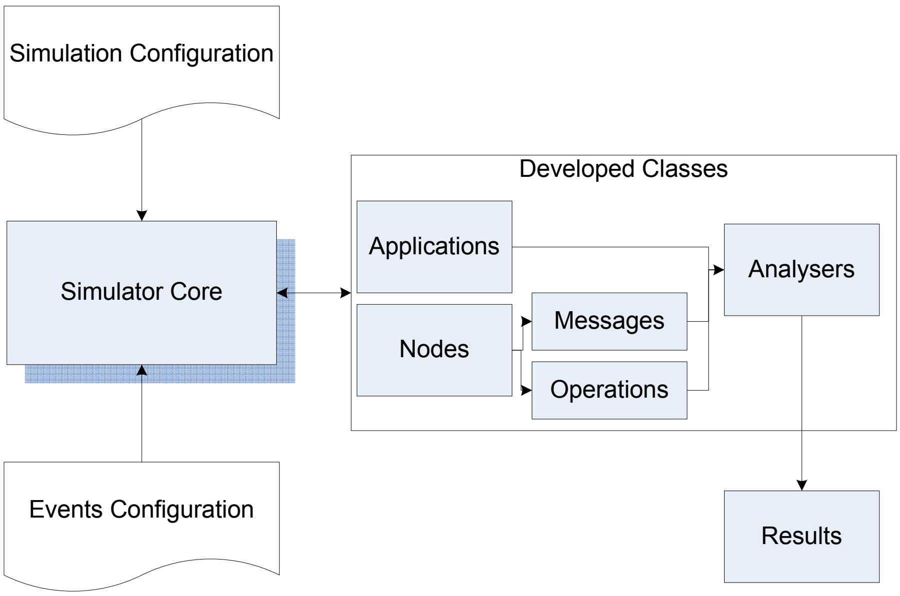


Figure 4.10: Simulations Model

4.3.2.1 Tools Employed

Besides the Java™ language, which is used in the simulator, some other tools have been employed for this study. Eclipse™, an Integrated Development Environment

(IDE), helped with the project organization and its integration with the Java compiler and debugger. The gnuplot project [Janert 09, Williams 10] was also used to generate figures. Some statistical data were produced with a spreadsheet application that exported data that could be used with gnuplot.

4.3.3 AgS Evaluation based on Average Path Length and Data Replication

As already discussed, the AgS copes with the problem of optimizing the search of services in a P2P SON. The task of validation carried out by AgS through its experimental evaluation, is an important factor in its acceptance and the way it is used. However, a question arises regarding what factors should be taken into account in the experimental evaluation. At first sight, performance was viewed as the criterion and this is important since it reflects the extent to which the AgS optimizes the searching process. This can be measured by comparing its performance with environments which lack the use of AgS and also with related solutions.

Regarding these aspects, some research questions can be raised in order to evaluate the performance of the AgS:

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

A way to answer these questions is to investigate some metrics related with the established criterion. These metrics must be of a comparable level in different environments so that they can point out with an appropriate statistical confidence interval (CI), which is the best solution and the most suitable environment. In view of this, the first metric assessed was the Average Path Length (APL).

The APL, as a metric, shows the difficulty or complexity to find a desired service in the AgS. The APL has to take into account the SearchMessage travelling around the AgS. It should be mentioned as a means of understanding this process, that the Search operation generates an initial SearchMessage 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 what kind of analysis is conducted. In the evaluation that was carried out, the APL for successful findings was used. Eq. (4.1) shows the formula for the APL.

It is worth mentioning that Search operations and SearchMessages are different things. Operations send messages and receive replies. However, in the case of the Search operations this is not always what occurs. When the aggregation peer that starts the Search operation finds the desired service or service component in its local cache, a new SearchMessage is not necessary and as a result, a reply is never received. As Eq. (4.1) shows, our analysis is based on the number of received replies to the sent query messages, so that it is possible to actually represent the APL as the average path travelled by a SearchMessage until it reaches its destination.

$$APL = \frac{\sum_{hops=0}^{n_{serviceFound}} hops}{SearchReplyMessages} \tag{4.1}$$

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

All in all, it should be noted that replication in the context of this thesis means the caching of information. The term replication is used because the information (service profile) is explicitly and actively copied from the source, which is the AgS peer where the service is found, to the destination, which is the requesting AgS. Ultimately, replication is performed with the help of a SearchReplyMessage.

4.3.3.1 Simulation Framework

In order to assess the AgS behaviour and answer the aforementioned research questions, it has been conducted a simulation study to measure the APL necessary to locate the SON peer that makes a particular service available.

The simulations 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 the 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 executed according the time specified in the simulation actions file.

As mentioned earlier, replication was activated during the execution of the simulations in its particular environment. The comparison with other scenarios/environments forms part of the results for this Section.

Two sets of simulations were executed comprising two scenarios involving the P2P SON: 1) with the AgS and 2) without the AgS. In each scenario, simulations were executed in two environments: 1) replicated search results, and 2) non-replicated search results. Figure 4.11 summarizes the simulation framework for the conducted experiments which comprised the performance analysis where the average path length is used as the metric.

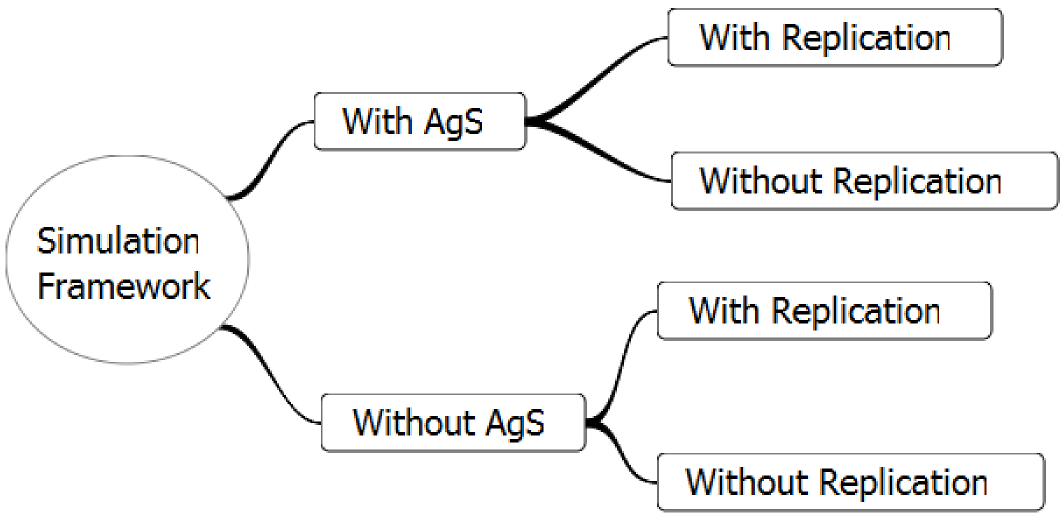


Figure 4.11: Simulation Framework for APL

4.3.3.2 Results

The APL analysis takes into account the number of hops required for successful SearchMessages, i.e. the number of aggregation peers that are counted between the query from the requesting peer and the providing peer through a traveller SearchMessage. The results rely on a confidence interval of 95% for the mean APL value.

Table 4.3 provides the values of APL for the experiments with and without AgS. The CI columns represent the values of the margin of error calculated for the 95% confidence interval. The first column called SearchOp represents (in descending order) the number of Search operations that were executed for each scenario and environment. The columns called Repl represent the values of APL in the environment where there is replication of the search results. In contrast, the columns called ∼Repl represent the values for the environment where replication was not used.

Table 4.3: Average Path Length Values

	With AgS				**Without AgS**			
SearchOp	***Repl***	***CI(±)***	***∼Repl***	***CI(±)***	***Repl***	***CI(±)***	***∼Repl***	***CI(±)***
1000	16,90	1,083836	17,85	1,135426	42,08	2,715183	44,49	1,993238
900	16,48	1,083836	17,30	1,135426	44,16	2,715183	46,55	1,993238
800	16,93	1,083836	17,70	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	17,50	1,135426	37,92	2,715183	45,19	1,993238
300	16,92	1,083836	17,30	1,135426	36,83	2,715183	45,44	1,993238
200	14,25	1,083836	15,88	1,135426	36,59	2,715183	46,00	1,993238
100	12,47	1,083836	12,47	1,135426	35,65	2,715183	46,01	1,993238

Figure 4.12 shows the APL for the scenarios and environments already mentioned. In this figure, two representative sets of curves can be seen. Those on top represent the scenario without AgS, and the higher curve indicates the environment without any 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 where there is replication. All the curves are drawn with a margin of error that considers a 95% confidence interval.

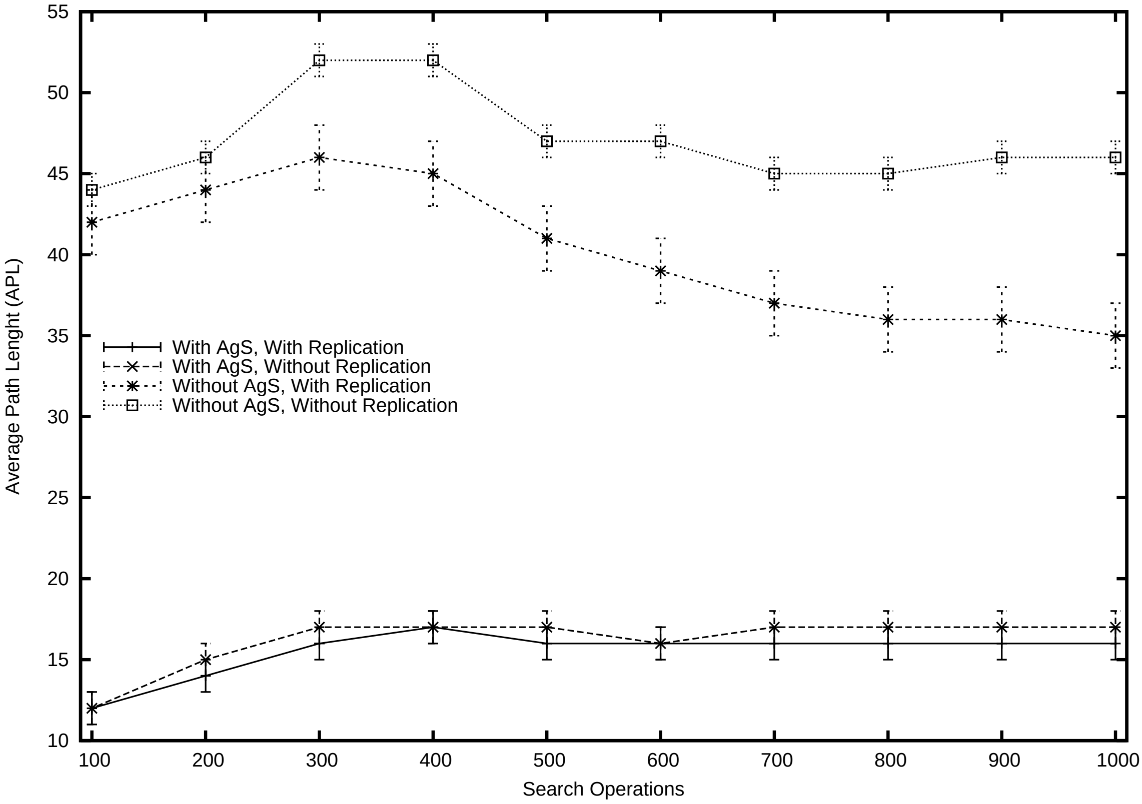


Figure 4.12: Compared Average Path Lengths

Figure 4.12 is quite revealing in several ways. First, the replication of the search results is worthwhile when there are repeated queries for the same service/service components. Taking the arithmetic mean of the APL from Table 4.3 as a 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 AgS. The APL is also low when the replication is used in the scenario with AgS, although, in this case, the difference is only approximately 3.73%.

These differences are explained by the concentration of the services and service components profiles in the AgS. 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 by concentrating the services on a lower number of peers and consequently making the queries less dependent on replication.

Figure 4.12 is also revealing with regard to the use of the AgS. Indeed, the AgS dramatically reduces the APL of the search. A comparison between the arithmetic mean values for both scenarios, that takes 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 and the environment where replication is taken into account, the difference is approximately 60.26%. These drastic reductions in the APL values show the effectiveness of using AgS to improve the searching of services and service components in a P2P SON and thus gives a positive answer to the first of the questions raised.

The behaviour of APL was evaluated to determine the statistical dependability of the results achieved in the experiments. This included a regression analysis where equations were found that describe how APL behaves in accordance with the increase in the number of search operations for the simulated scenario. This behaviour can be seen in Figure 4.13.

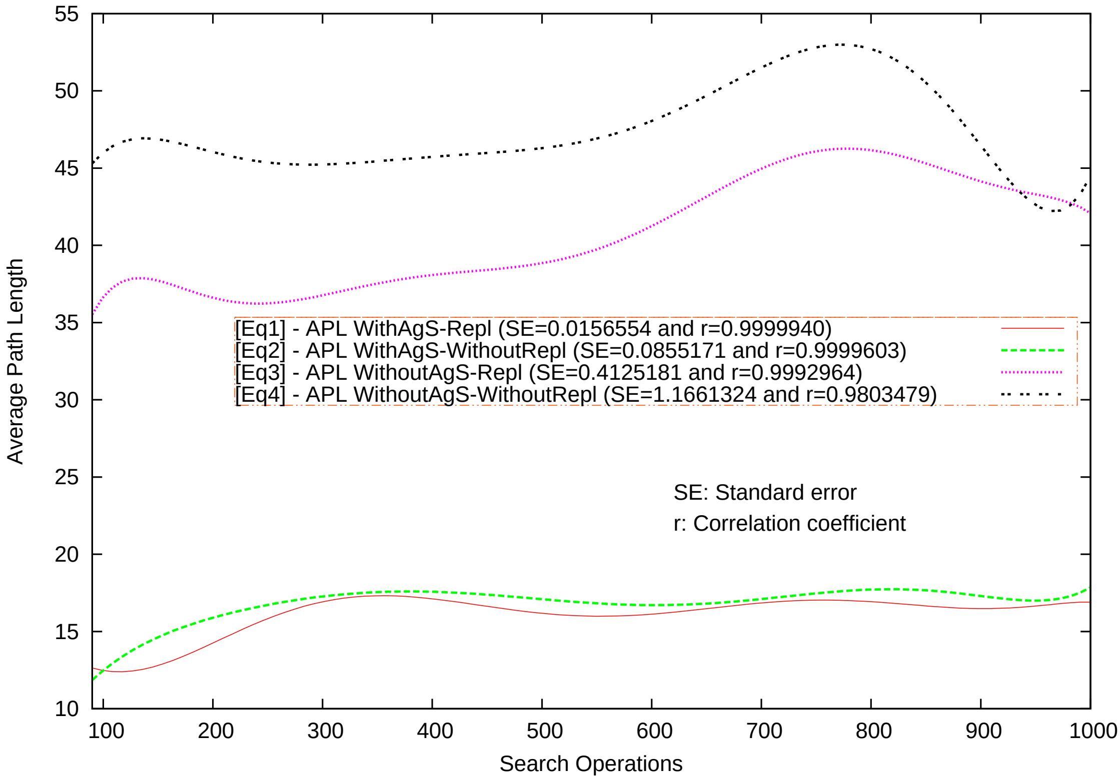


Figure 4.13: Average Path Length Probability Density Functions

The curve adjustment by polynomial regression (the standard error and correlation coefficient are shown in the graph) is expressed in 4 different equations that describe the behaviour of the APL where x is the number of search operations and y is the APL. Table 4.4 shows the approximate coefficients for the curves displayed in Figure 4.13. Given the high correlation coefficient and relatively low adjustment error (at least for the scenario with AgS), we assume the remaining behaviour can be predicted by using these polynomials. Moreover, this allows the functions in Figure 4.13 to be considered as the probability density functions for the APL behaviour in the carried out simulations.

These findings were obtained from 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 ranging from a minimum of 1 to a maximum of 7. Nevertheless, what happens when the distribution of services changes in the AgS or in the P2P SON? This indeed is the third question to be answered in this Section. Thus, on the basis of the assumption that the set of 1000 search operations is the most significant set of search operations, we have simulated the changing distribution of the services. Figure 4.14 shows the results. There, the x axis represents the maximum number of services a peer can offer (published

Table 4.4: Approximate Coefficients for Polynomial Functions in Figure 4.13

	≈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	

in AgS or kept locally). All the simulations triggered 1000 search operations. The y axis represents the APL. As expected, the higher the number of services a peer can offer, the lower the APL will be. It is worth to mention that the peers are still constrained to publish their offer to a maximum of 10 aggregation peers.

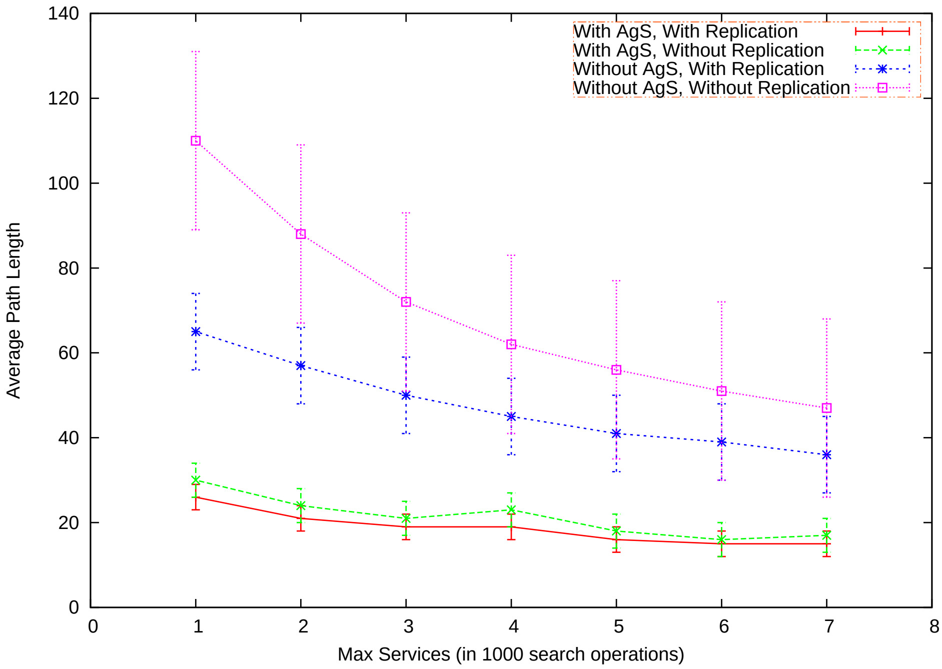


Figure 4.14: Influence of Services Distribution on the APL

This preliminary result means that the more services a peer can publish the better for the AgS in terms of APL and thus, more easily these services can be found. This helps to explain the low APL with AgS in Figure 4.12. Since peers publish a maximum of 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 to the searches.

4.3.3.3 Summary

This section summarises the experiments and results comprising APL; whether or not replication is used and its influence on the APL; and the service distribution regarding the Aggregation Service (AgS).

The AgS is a second-tier P2P overlay whose *modus operandi* is based on the publication of service offerings by peers residing on an underlay P2P SON. This publishing scheme supports the optimized search of service performed by the AgS.

This Section has evaluated the AgS through simulations by using the average search path length (APL) as the assessment metric. The APL is the ratio between the forwarding hops a Search Message experiences from the search requesting peer to the peer that keeps the searched service reference, and the number of positive returned Search Reply messages.

The results show that the AgS reduces the APL by 64.67%, and 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 reduce 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.

4.3.4 AgS Performance and Scalability: an Evaluation using Response Time

As already stated, performance was chosen as a criterion in order to provide clues about the AgS efficiency on optimizing the search of service and service components in a P2P SON environment. Section 4.3.3, provided results of performance using the Average Path Length (APL) as a metric. Although APL explains the difficulty and complexity of finding a desired service in the AgS comprising the hops of the SearchMessage between the requester and replier, it does not consider how fast this is carried out.

Thus, another metric was employed, instead of APL, to consider how long the users have to wait for the AgS to answer their requests. This metric, like the APL, allows to make comparative measurements between environments, which is helpful to evaluate the AgS as a search optimizing solution. The metric that best fits these definitions is response time (RT).

The response time as a metric plays the role of an assessment parameter not only for the efficiency of the AgS itself, but also the user's Quality of Experience (QoE) indicator for the use of AgS.

With regard to the response time and the use of the AgS, new questions arise that go beyond the answers to the questions in Section 4.3.3:

1. Is the search response time, faster in an environment which has an Aggregation Service?
2. Which is the scenario best suited to AgS with regard to response time?

In a similar way to the APL experiments, the response time is obtained by the SearchMessage. The way a SearchMessage travels around the AgS ring remains the same, i.e. it is forwarded to the next aggregation peer in case the receiving aggregation peers do not find the requested service or service component at its local cache. Therefore, the response time is computed by how long it takes for a SearchMessage, which is triggered by an aggregation peer, to go from the requesting aggregation peer to the receipt of a SearchReplyMessage for that search.

In order to accomplish that, when the Search operation starts, the SearchMessage receives a local time stamp (TS) that is deducted from the current local time at the arrival of the SearchReplyMessage for that search. Hence, the RT will be the ratio between the accumulated time for all the successfully accomplished search messages and the number of these messages. In fact, the considered RT, according to this definition and illustrated by the Eq. (4.2), is the average RT for experiment executed, comprising a confidence interval of 95%.

$$RT = \frac{\sum_{t=0}^{n_{infoFound}} CurrentTime - TS_t}{QueryReplyMessages} \tag{4.2}$$

The replication/caching of the searched service profiles is performed in the same way as for the experiments with APL. However, considering that the aforementioned results pointed out that the replication supports searches with a smaller APL, then only environments using this mechanism were compared.

Despite the comparisons between environments using replication either with or without AgS, the gnutella in its version 6 was also used to answer the second question that was raised. The gnutella [Ripeanu 02] is a well known unstructured P2P overlay network that does not make use of a DHT to bond data and peers location, and this allows services to be distributed randomly among the peers. These characteristics match the same AgS characteristics, which allow a fair comparison to be made. The popularity of gnutella's systems was another factor that was taken into account when this decision was made.

All things considered, the search hit ratio must also be incorporated in any evaluation of performance based on the AgS response time. This is important since it can provide further data which can be drawn on to analyse the eventually existent trade-offs in the compared scenarios.

4.3.4.1 Simulation Framework

The results of the performance evaluation comprising the response time (RT), were achieved by simulation. The simulations involved a sample of thirty particular sets of aggregation peers, which meant thirty particular AgS with different sizes. Each set kept the services that were made available and published by 10,000 SON peers spread over 10 different domains.

As in the experiments using APL as the metric, a particular SON peer can only publish a maximum of seven services or service components randomly chosen (using a uniform distribution) from the service set S={S1,S2,S3,S4,S5,S6,S7}. Each administrative domain has its own set S. For the sake of simplicity, each service or service component profile is treated as a single service descriptor consisting of the name of the service or service component.

Moreover, 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 subset of services at the same domain and publish them on the same aggregation peer. In the interests of simplification, the search concludes with the first match though AgS has the ability to return all the matches.

Each execution simulates 50 hours of work. In the case of each AgS, every simulation used the same scenario with a different number of search operations, which ranged from 100 to 1000 search operations (step by 100).

Two sets of simulations were executed in two scenarios involving the P2P SON: 1) with the AgS and 2) without the AgS. A third scenario that involved the P2P SON with gnutella was also simulated. Figure 4.15 shows the scheme for the simulation framework regarding the performance analysis, which uses the response time as the metric.

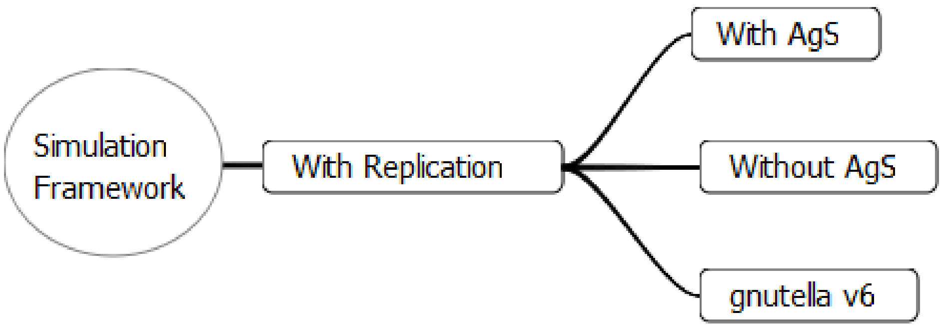


Figure 4.15: Simulation Framework for Response Time

It should be noted that the points in the graphs for Section 4.3.4.2 express the average value for the entire set of 10 individual simulations, each of which consists of a particular number of search operations, which ranged from 100 to 1,000 (in stages of 100). Caching of the search results was also included to optimise the search process.

Finally, as in the case of the APL experiments, the PeerFactSim.KOM [Kovacevic 07] discrete events simulator was used in the simulations.

4.3.4.2 Results

Although this Section is concerned with the performance and scalability of the AgS, it is of prime importance to know the AgS search hit ratio to be able to analyse the big picture. According to the simulations, the AgS search hit ratio is almost 100%. Despite AgS is an unstructured overlay, its arrangement and way of working (see Table 4.1) ensures that it will find the information if it is published. The relatively unsuccessful search hit ratio can be explained by the fact that some of the Search operations had looked for non-existing services at the time of their execution. For example, the SearchMessage circulated the whole AgS overlay ring but was unsuccessful. Figure 4.16 shows the AgS search hit ratio in terms of the number of search operations and the AgS size.

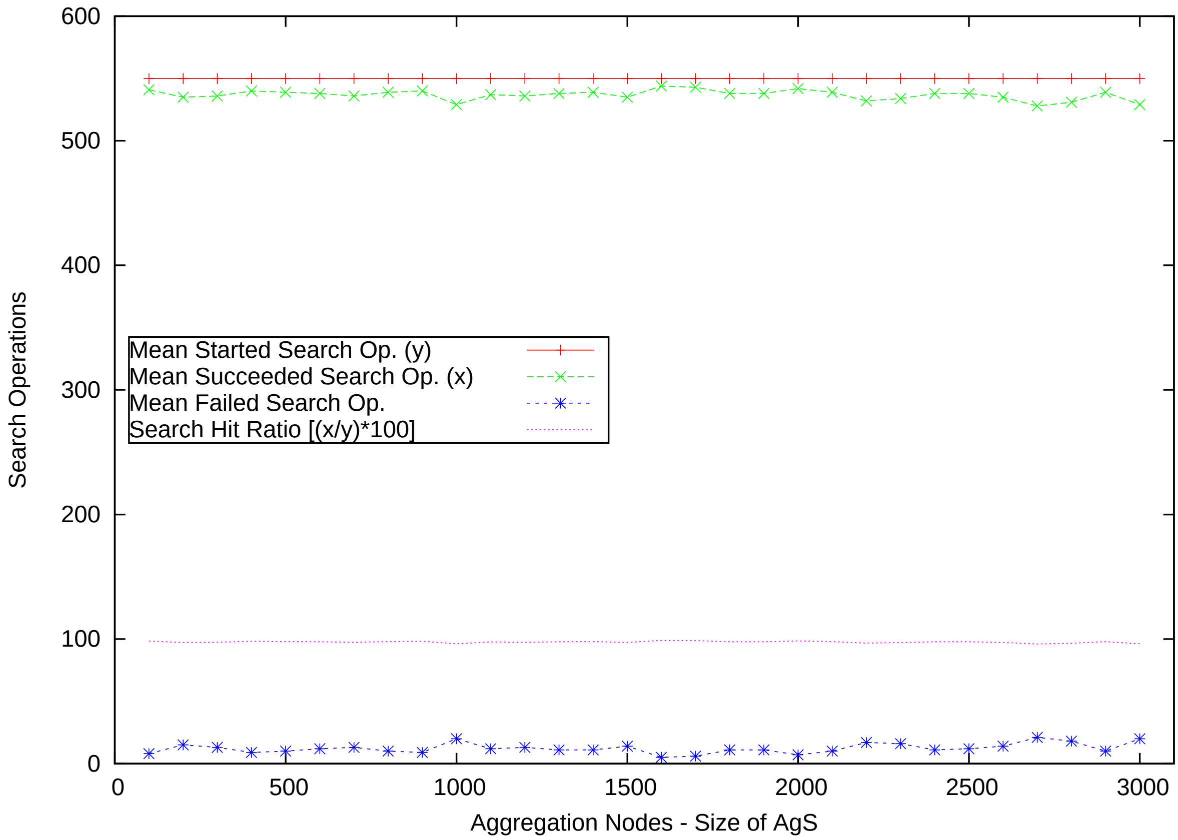


Figure 4.16: AgS Hit Ratio for Search Operations

The search hit ratio is very similar to the experiments carried out in an environment without AgS. This is explained by the fact that the same overlay mechanisms were employed at the P2P SON level, to ensure the process of making fair comparisons. This result means that almost any service or service component can be found, if it is present in the P2P SON. This similarity seems to testify against AgS, since the search hit ratio is expected to be greater in AgS. However, in spite of this similarity, this does not mean that the AgS does not make any difference. The advantage of AgS is that it improves the performance of the searches, as will be demonstrated.

Figure 4.17 shows a comparison between the environments with and without AgS regarding the RT. In this figure, it is possible to notice two representative curves. On top is the one that represents the environment without AgS. On the bottom is the curve for the scenario with AgS.

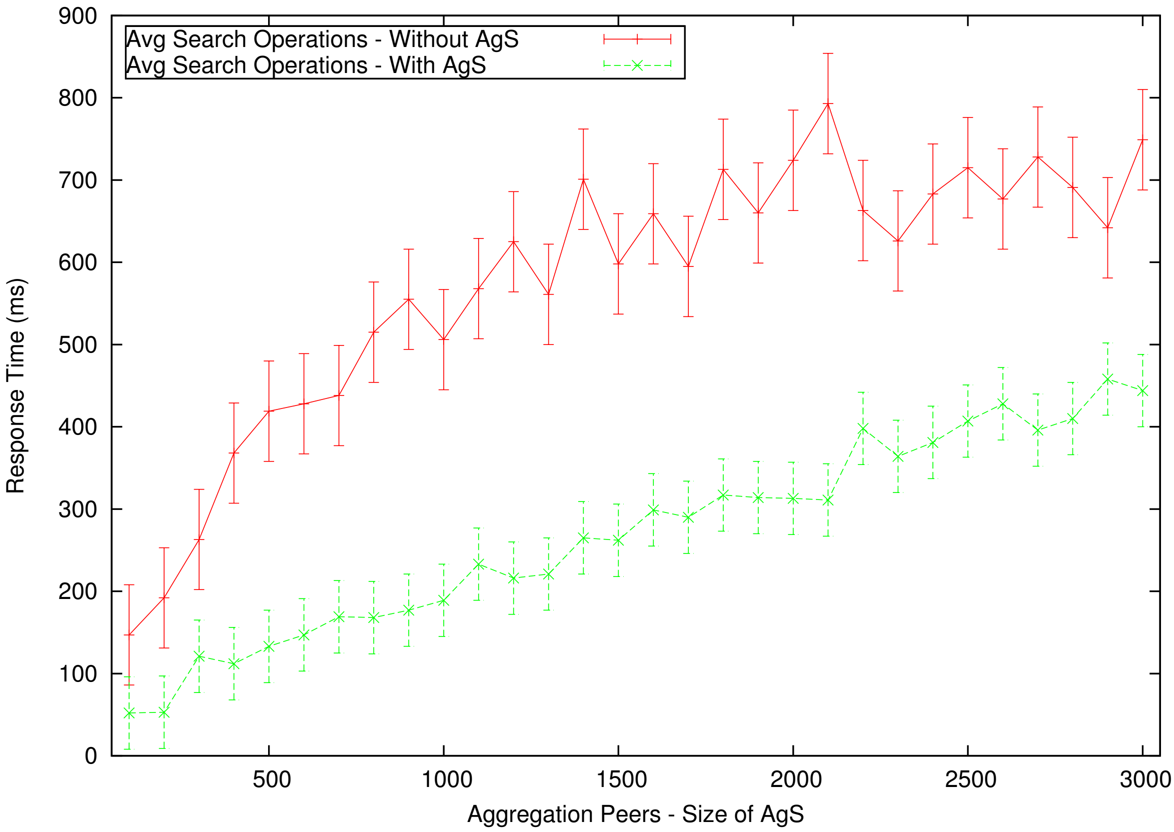


Figure 4.17: Compared response time - with and without AgS

Figure 4.17 is quite revealing in several ways. First, it shows that the response time is lower when AgS is used and that the RT remains lower than 500ms. When a comparison is made with the arithmetic mean of the plotted values, it is possible to figure out that the RT when AgS is used, is around 53% lower than in the P2P SON environment. This difference gives a positive answer to the research question about the performance of the AgS.

There is also evidence of the AgS scalability from Figure 4.17 since while the increasing factor for aggregation peers is 100, the increasing factor for RT is around 8.5. This means that the response time increases more slowly than the number of aggregation peers, i.e. the service provider can use a bigger AgS to reach more users with a relatively minimal increase in response time for searching their services and component services.

Comparison of Performance

The results given above and their analysis show the AgS performance and scalability in a positive light. However, a comparison with other environments is necessary and this was carried out by simulating and testing a number of search operations in the same scenarios but using another P2P overlay. The chosen one was gnutella, which is an unstructured P2P overlay in the sense that it does not make use of a DHT to bond data and peers location, and thus allows services to be distributed randomly among the peers. These characteristics correspond to the AgS characteristics, and so a fair comparison can be made. The popularity of gnutella's systems was also a factor when making this decision.

The version of gnutella used for the comparison was number six. The main parameters of the gnutella performance concerning the overlay formation, were as follows:

- maximum of 20% of super peers,
- ping max ttl of 3,
- query max ttl of 5,
- maximum number of 12 leaf contacts for each super-peer,
- maximum number of 12 super-peer contacts for each super-peer, and
- maximum of 3 contacts for each leaf peer.

Figure 4.18 shows the plotted averaged RT values for the environment using AgS and for the environment using gnutella. It is possible to note the curve representing gnutella shows a lower response time than the one representing AgS. Despite some small overlays, this is the general rule. However, in environments that are composed of few nodes, (which in the experiments conducted represents around 20% of the total), the environment with AgS had a slightly better or at least a similar performance. Nonetheless, for the remaining 80% of the simulated environments, the gnutella showed a distinctly better performance. Thus, when the mean values of these two curves are compared, it is clear that RT for the environment with gnutella will be around 48% lower than in the environment with AgS. This difference is an argument in favour of using gnutella for the searching and might represent a drawback for AgS. However, the searching performance is not the only one characteristic a provider should take into account when choosing the right environment for offering their services.

The search hit ratio provides clues about the search efficiency in terms of started and successful concluded searches. As already mentioned the search hit ratio for the environments with or without AgS is almost 100%. Thus, although the AgS response time is not as good as in gnutella, the AgS search hit ratio is better. The simulations in the scenario with gnutella reveal lower values for search hit ratio than in the environment with AgS. When the arithmetic mean is calculated for these values, the result points out the search hit ratio for the environment with gnutella is around 69%. This value is considerably lower than the 98% achieved in the AgS.

Figure 4.19 shows a comparison of the search hit ratios in both the scenarios - AgS and gnutella. This figure shows that the search hit ratio for the environment with gnutella is lower than in the environment with AgS. Given the averaged values for the search hit ratio of the AgS and gnutella environments, the "bias" that weighs in favour of AgS is approximately 42%. This bias can be explained by the well known fact that gnutella lacks reliability when searching for information even when that information is present. The reason for this is that gnutella's searching mechanism is based on flooding and has to be scrimped to avoid an overload of search messages in the network.

An important finding emerges when both the Figure 4.18 Figure 4.19 are examined together. The search hit ratio as well as the searching performance are better in AgS

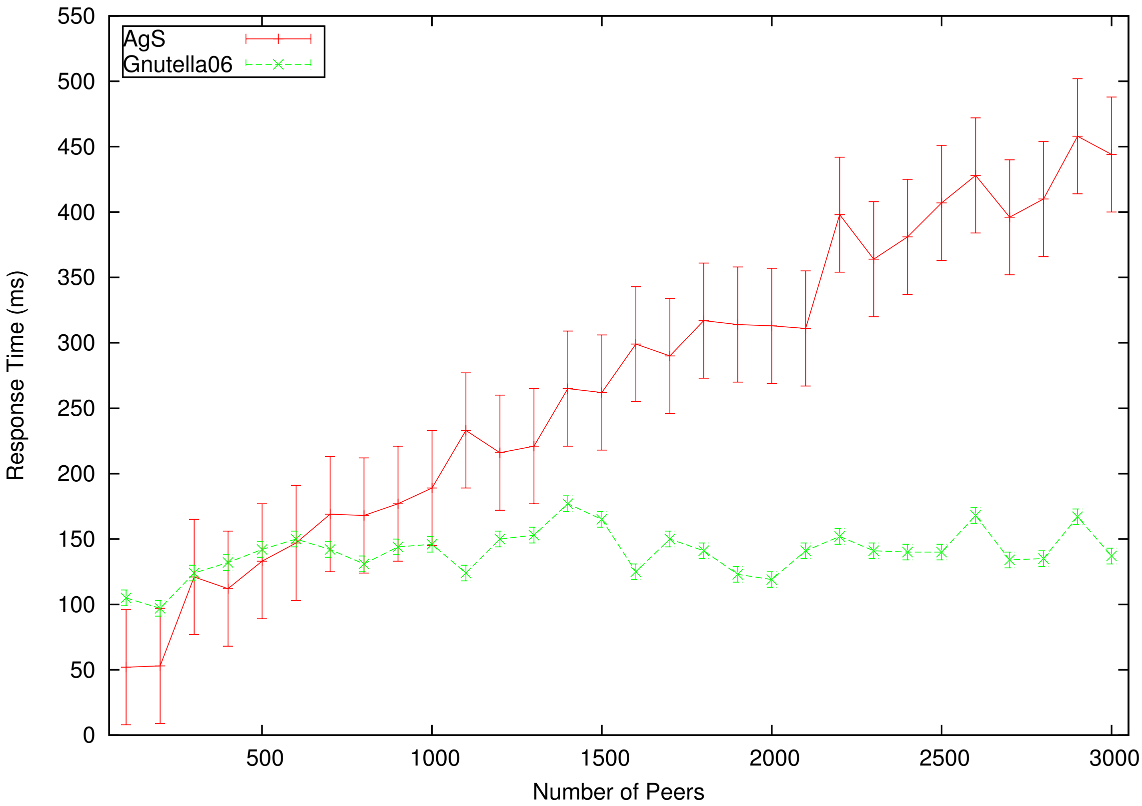


Figure 4.18: Comparison of response times - AgS and gnutella

than in gnutella for small overlays, i.e. in P2P SON with a maximum of 600 nodes. On the basis of this finding, it can be claimed that the AgS is in a better position to satisfy consortiums of service providers that use a small number of nodes to form their P2P SON and offer their services in this infrastructure. This might be the case, for instance, for specialized service providers that attend a specialized market niche such as the subtitling and dubbing video content market.

4.3.4.3 Summary

This Section has undertaken a performance evaluation in the context of the Aggregation Service that has employed response time as its assessment metric. Simulations were performed to carry out the evaluation. These simulations used RT and involved making comparisons between the search hit ratios of scenarios with and without AgS to assess

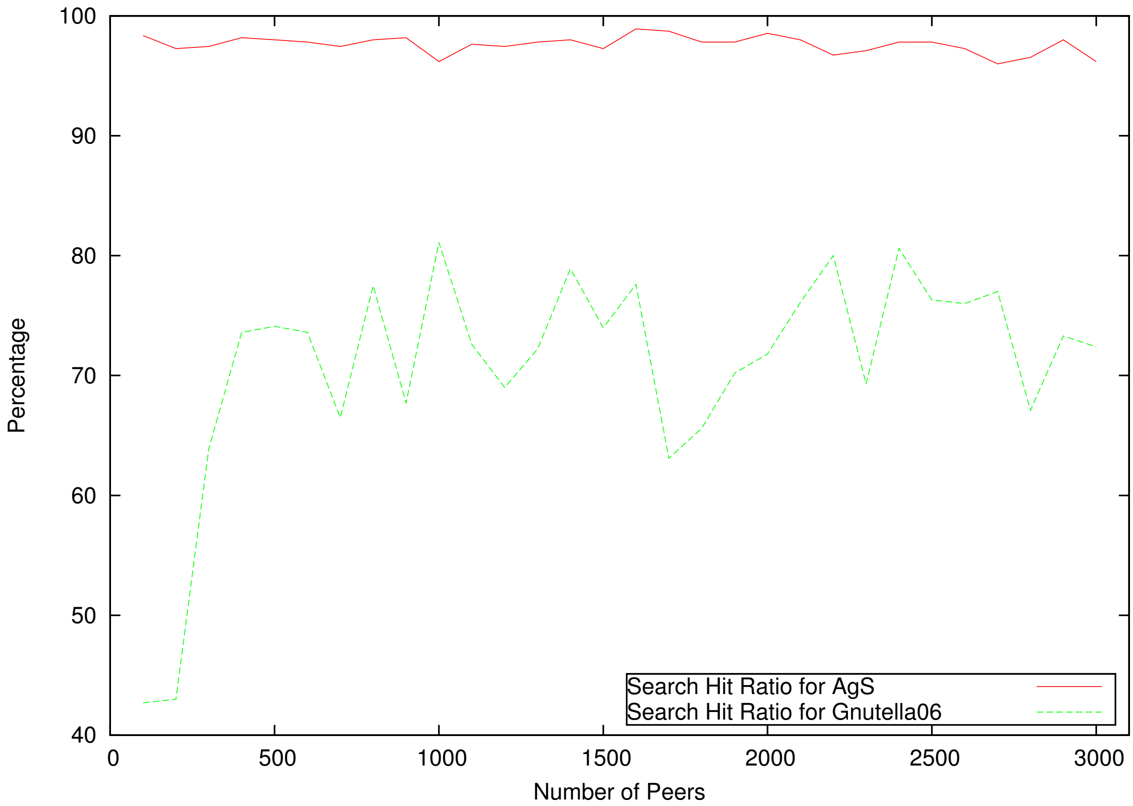


Figure 4.19: Comparison of search hit ratios - AgS and gnutella

performance and scalability. In addition, the performance of AgS was also compared with the performance of a scenario using gnutella as a search mechanism.

The results show that the AgS reduces the search response time by about 53% when compared with the same searching operations in a scenario that does not use AgS. Even with this reduction, the AgS search hit ratio, (which is around 98%), remains unaltered. The results also show that the AgS scalability is good. Although there is a direct relation between the number of nodes and response time, this rate is low. While the increasing factor for aggregation peers in AgS is 100, the increasing factor for RT is around 8.5, i.e. around 11:1.

The comparisons between AgS and gnutella show a trade-off. Despite gnutella is faster, it has a search hit ratio that is approximately 42% lower. On the other hand, AgS is around 48% slower. Nonetheless, for small environments (maximum of 600 nodes) the "bias" favours AgS, regardless of response time or search hit ratio. This finding leads to the conclusion that AgS is well suited for specialized market niches, where the overlays formed by the active service providers are small.

4.3.5 AgS Dimensioning Evaluation

As already demonstrated, AgS plays an important role in the search of services optimization process. This role is especially highlighted by the AgS operation over a P2P SON. Thus, even though services are published and searched in the AgS level, AgS still depends on a number of selected peers chosen among those belonging to the P2P SON. These are called aggregation peers, and play the role of aggregators in the search optimization process. Bearing this in mind, a question arises:

- What is the best size for the AgS in terms of the number of aggregation peers?

This question concerns the number of peers that are needed to compose the AgS service to improve its search efficiency and performance, when compared with the native searching mechanisms provided by the underlying P2P SON. As the aggregation peers are part of the P2P SON, it is important to know the maximum number of peers that can be selected to be part of the AgS in order to perform searches more efficiently than just by searching in the P2P SON. This is the context of this evaluation. It means that suitable AgS dimensioning helps service providers to plan their infrastructures, e.g. how many peers should be made available to the P2P SON and consequently to the AgS, so that they can keep costs under control.

In view of this, several simulations were carried out to dimension the AgS service. These simulations took into account the search response time and the AgS overhead performance as metrics. These metrics are used as a baseline between the environment with AgS and the P2P SON environment. This means that by determining these metrics for a fixed set of search operations for several particular sizes of P2P SON and of AgS, it is possible to decide how many peers are needed to form the AgS overlay and obtain a reasonable degree of search efficiency. As in the previous AgS evaluations, the response time is obtained in the same way by using the SearchMessage traveling time and this is shown in Eq. (4.2).

4.3.5.1 Some Definitions for AgS Dimensioning

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 a subset of P2P SON peers. Hence, in principle, $|AgS| \leq |SON|$. Let us define e as the search efficiency, which is given by the response time metric. Thus, 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. $\mathrm{e}_{SON} = 1/rt$. On the other hand, e_{AgS} is the search efficiency in the AgS. However, in this case, rather than taking only into account the search performance, e_{AgS} must also take into account the AgS overlay set-up and maintenance performance, which in this work are collectively referred as overhead. Hence, $\mathrm{e}_{AgS} = 1/(rt + ovhd)$, where rt is the response time and $ovhd$ is the overhead. This overhead results from the time spent by the AgS overlay 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 dimensioning evaluation is stated in Eq. (4.3).

$$(\ max|AgS|\ ) \mid \mathrm{e}_{AgS} \geq \mathrm{e}_{SON} \tag{4.3}$$

Having stated the AgS dimensioning objective based on the search efficiency, a validation framework supported by simulations was established to assess the size of the AgS.

4.3.5.2 Simulation Framework

Simulations were performed to achieve the maximum size of AgS in accordance with the P2P SON size. To accomplish that, fifteen hundred individual simulations were carried out and involved a sample of thirty particular P2P SONs of different sizes, starting with 100 peers and rising to 3000 peers, in stages of 100-peers.

Each particular P2P SON executes, makes available and publishes its services, which are spread over 10 different domains. As in the previous experiments regarding AgS evaluation, a particular SON peer can only publish, a maximum of 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 (following a uniform distribution).

In the case of each simulated P2P SON, four particular AgS overlays running on top of it were set up. Each 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 and performed the same 1,000 search operations, which were repeated 10 times to obtain an average result. First of all, the P2P SON environment was simulated, with the search operations being executed. After that, each AgS overlay was simulated, over the previous simulated P2P SON, and with the same search operations being executed. A configuration file with the discrete-times of the search operations 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 the simulations.

As in the previous AgS evaluations, replication/caching of the search results was also involved. The PeerFactSim.KOM [Kovacevic 07] discrete events simulator was used in all of the simulations.

4.3.5.3 Results

As already mentioned, the AgS efficiency, which is expressed in Eq. (4.5), also depends on the overhead time. The overhead results from the time spent in setting up and maintaining the AgS overlay, by means of join (j), publish (p) and leave (l) control operations, according to Eq. (4.4).

$$ovhd = \sum_{m=1}^{n_j} time(j_m) + \sum_{m=1}^{n_p} time(p_m) + \sum_{m=1}^{n_l} time(l_m) \quad (4.4)$$

$$\mathrm{e}_{AgS} = \frac{1}{(rt + ovhd)} \quad (4.5)$$

Figure 4.20 shows the results concerning the response time and the overhead for the simulated scenarios. It is worth mentioning the results rely on a confidence interval of 95%.

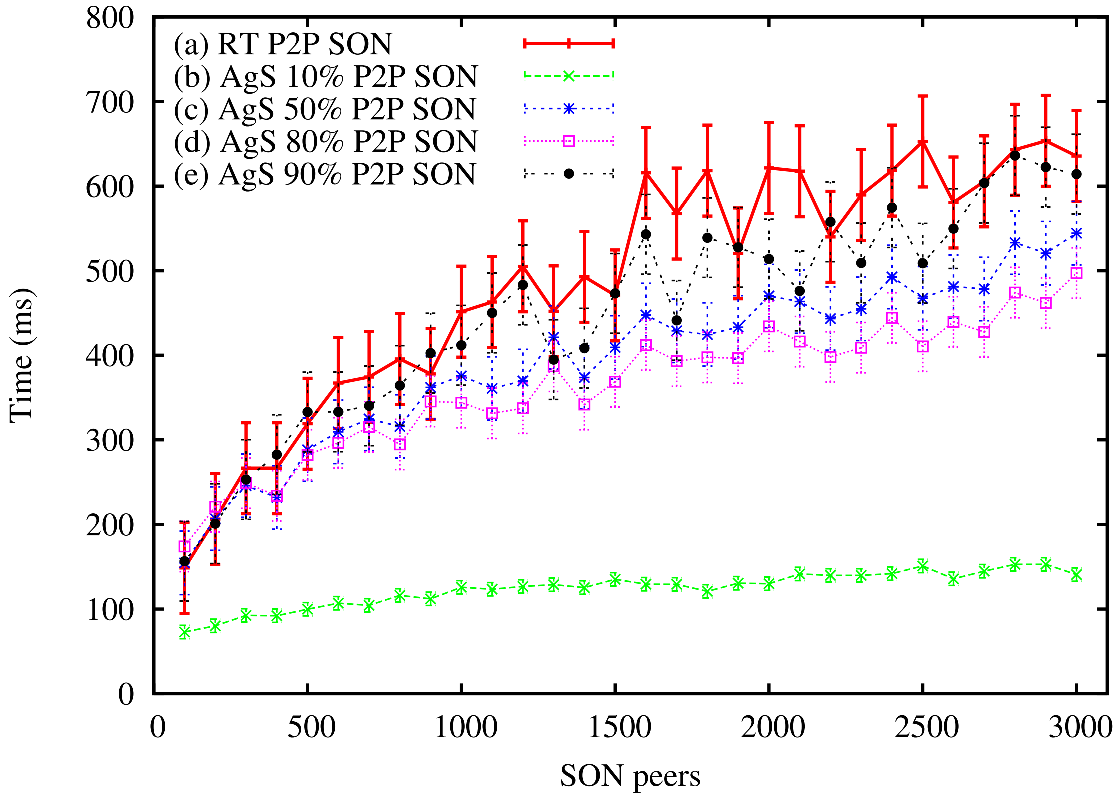


Figure 4.20: AgS efficiency with overhead

It is possible to notice that the AgS service is very efficient since, for most of the results, AgSs (whose size is up to 90% of that of P2P SON) still leads to faster searches (even with overhead) than the plain P2P SON. Moreover, as can be seen, the smaller the AgS overlay the better. This can be explained by the high concentration of services in these relatively few peers, as is the case of AgSs with 10% of the P2P SON peers.

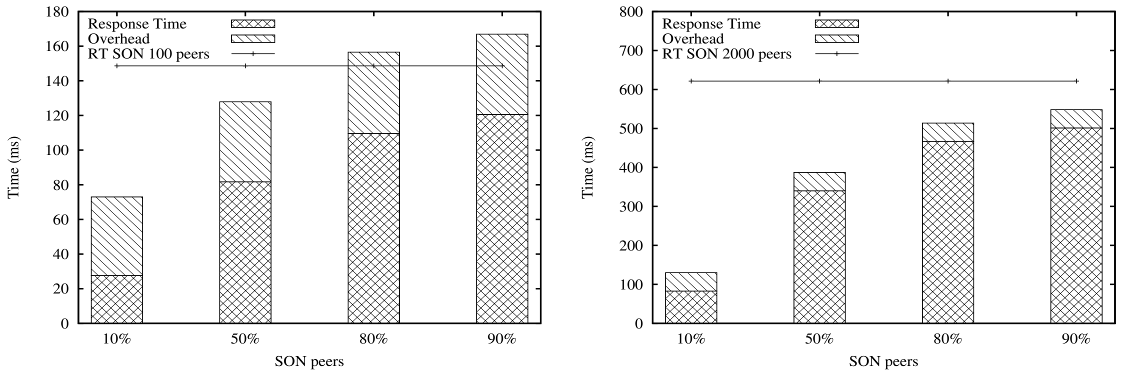
Some P2P SON and respective AgS sizes display a particular kind of behaviour. This is especially the case with small P2P SONs (those whose size is smaller than 600 peers), where can be observed that the search time (efficiency) of the AgS is worse than the search time for the P2P SON. All in all, the observation of this behaviour in these conditions allows us to conclude that for small market niches, where service providers create a small P2P SON, for the purposes of searching, the AgS must not be greater than 80% of the P2P SON.

The influence of the overhead is stronger on smaller P2P SON and the respective AgSs. As services are concentrated in the P2P SON and in the AgS overlay in equal proportions, the searches are fast. Thus, even a small AgS overhead has a negative effect on the searching efficiency. The influence of the overhead can be seen in more detail in Figure 4.21.

Figure 4.21.(b) shows that even when 90% of the SON peers form a part of the AgS overlay, AgS not only searches faster than P2P SON but also the entire e_{AgS} (i.e. search plus overhead) is greater than the e_{SON}. On the other hand, Figure 4.21.(a) shows situations in which $\mathrm{e}_{AgS} < \mathrm{e}_{SON}$. These cases highlight the fact that, when AgS sizes start at 80% (in fact, with less than 80%) of the P2P SON size, the overhead is responsible for degrading the AgS efficiency. Nevertheless, it is worth remembering that when the overhead is dismissed, e_{AgS} is always greater than e_{SON}.

Finally, Figure 4.22 displays a histogram giving the arithmetic mean values for all the experiments. The average response time in the P2P SON is approximately 487ms whereas for the AgS it is approximately 77, 270, 403 and 448ms for AgS sizes of 10%, 50%, 80% and 90% of the P2P SON size, respectively. The overhead is quite similar for all the AgS sizes, with a value close to 47ms.

Figure 4.22 allows to conclude that the AgS service can lead to a very good performance when compared with the plain P2P SON approach. In addition, the average overhead remains stable even though there is an increase in the number of aggregation peers. Generally speaking, the obtained results suggest that it is possible



(a) 100 peers (b) 2000 peers

Figure 4.21: AgS efficiency x P2P SON size

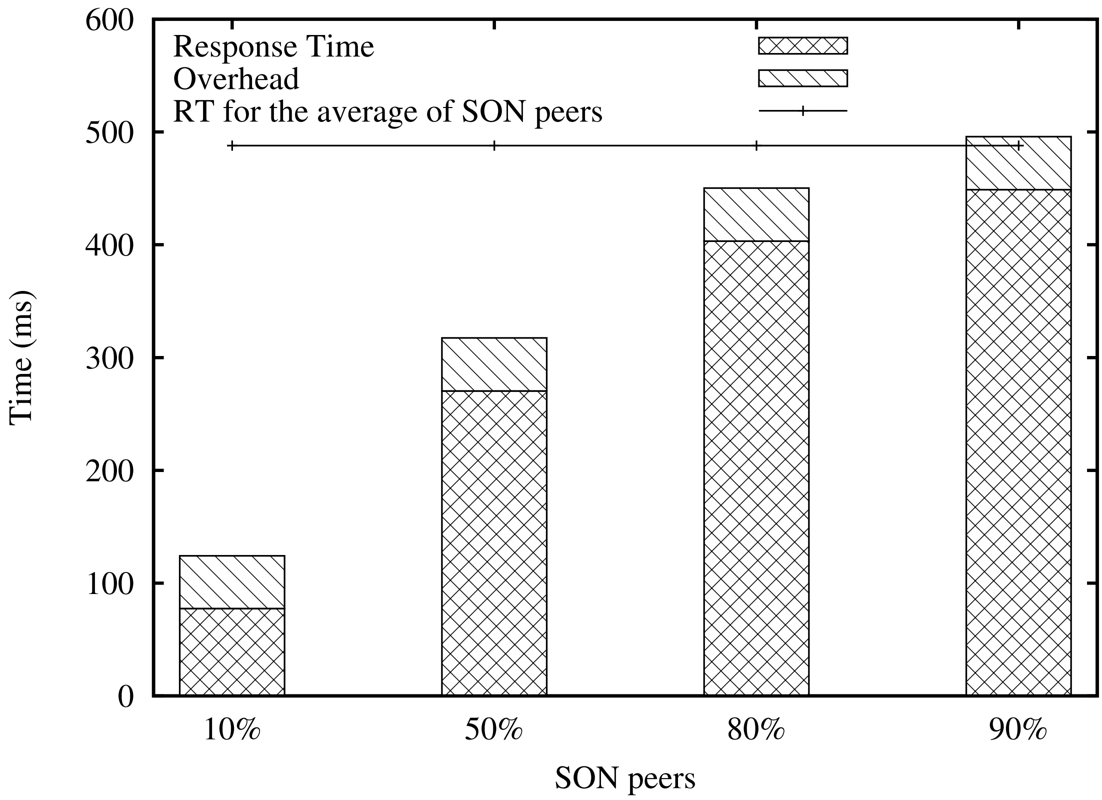


Figure 4.22: Average AgS efficiency x P2P SON size

to claim that the AgS approach is highly beneficial and gives service providers ample freedom to decide on the number of 2^{nd}-level peers without jeopardizing the gains in performance.

4.3.5.4 Summary

This Section addressed the issue of determining the relation between the number of peers in the Aggregation Service and the efficiency of the service search operations Simulations were carried out to compare several particular sizes of P2P SON and AgS. These simulations employed the RT as metrics for a fixed set of search operations and the AgS overhead with its maintenance operation. The reasoning is that by making these comparisons, it is possible to decide on the number of peers that are needed to form the AgS overlay to obtain a degree of search efficiency that is better than just searching on the P2P SON.

Two important findings are shown in this Section. First, 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. Second, the influence of the AgS overhead maintenance operations is stronger at small AgS overlays. This latter finding explains the fact that for small P2P SON, i.e. those with a maximum of 600 peers, the AgS should have at most 80% of the P2P SON size to have a better search efficiency than the P2P SON itself.

Therefore, according to the results, the AgS has a very good potential to improve the overall search performance when compared to the capacity for search operations of a single-tier P2P SON overlay, and is able to achieve this at the cost of a very small overhead. In addition, the obtained results can be easily used by cooperating with ISPs to dimension their AgS.

4.4 Summary

In this Chapter, there has been an examination and evaluation of the Aggregation Service (AgS) model. An attempt has been made to show how AgS, which was integrated in OMAN, can optimize the search of services and service components in an environment composed of several service providers that are organized in a P2P Service Overlay Network (P2P SON). Initially, AgS allows a consumer (e.g. service provider, final user, etc.) to find service(s) that can meet the desired service-specific requirements. AgS, which is implemented as a ring overlay-tier running upon the P2P SON, achieves this by analysing and comparing, (in a clockwise manner), the parameter values of the interrelated service desired. AgS stores service profiles on an individual basis, thus selecting the suitable peer(s) that execute it and this is made possible by peers at the P2P SON level, which execute the services, and publish service profiles to the peers that compose the AgS.

Furthermore, the two-tiered AgS architecture allows the publishing and searching functionalities (provided by the AgS) to be split off from the service management functions that are carried out in the P2P SON-tier. Thus, the sensitive information and configuration of 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. This flexibility of the AgS model allows service providers to hide their business strategies so that they can cope with the competitive business demands.

This chapter has conducted an analysis of the AgS model and architecture with regard to its composition, usage and advantages. It has also carried out an AgS validation in several scenarios comprising several evaluation experiments, which has been undertaken in three axes: 1) Performance coping with average path length (APL); 2) Performance coping with response time (RT); and 3) AgS dimensioning.

The experiments and results including APL, are based on how many hops a search message takes to reach the target being searched. In this case, the results showed that the use of AgS reduces the APL in both of the compared environments both with and without the caching/replication of the searched results. The comparison between the scenarios with AgS and without AgS have been shown, on average, a positive difference of 64,67% and 60,26% for the environments without caching/replication and with caching/replication, respectively.

The results that concern RT are based on how long a requesting peer has to wait for an answer about the requested service. The experiments that were carried out showed that the AgS reduces the search response time by about 53% when compared with the same searching operations in a scenario that does not use AgS. However, even with this reduction, the AgS search hit ratio, which is around 98%, remains unaltered. Furthermore, a comparison between AgS and gnutella v6 has been carried out and the results showed that gnutella is faster, though its search hit ratio is approximately 42% less than AgS. Nonetheless, for small environments (maximum of 600 nodes) AgS is more effective, regardless of response time or search hit ratio.

Finally, the results for AgS dimensioning were analysed and discussed. The AgS dimensioning copes with determining the relation between the number of peers in the AgS and the efficiency of the search operations. These results revealed important findings. First, the results showed that 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. Second, the influence of the AgS overhead maintenance operations is stronger at small AgS overlays. This latter finding explains why in small P2P SON, i.e. those with at the most 600 peers, the AgS should have at most 80% of the P2P SON size to have a better search efficiency than the P2P SON itself.

Chapter 5

Best Peer Selection Service - BPSS

In the context of a large utilization of OMAN, the growth in the number of service providers and hence in the number of offered services, can lead to questions regarding which peer, (among several providing the same service), is the most appropriate to serve a request.

Particularly considering P2P SON as infrastructure to offer services, the already mentioned Aggregation Service helps to optimize the finding of peers that provide a particular service. However, the AgS by itself cannot handle the choice of the best peers among all those that execute that service. This means that, although the AgS optimizes the search of SON peers to provide a particular service, it does not guarantee that the found peers are those which can provide a suitable service performance for the requesters. Thus, the best peer must be found in the P2P SON, (from among all the potential partners that provide the desired service) to maximize performance interaction between the service requester and the service peer provider.

The selection of the best peers is an important task in the context of offering of services in P2P SON environments because it allows the interaction performance between the requesting peer and the selected one to be improved. In fact, performance is not the only pursuit metric at the best peer selection process. In fact, the metric depends on the application/service type that uses the BPPS. Generally, a good metric should take into account one or more of a set of Quality of Service (QoS) parameters, such as delay, jitter, available bandwidth, etc. Nevertheless, the choice of peers belonging to a different, remote domain should be avoided as much as possible to keep inter-provider traffic to a minimum and reduce the costs for the user and service provider. Thus, locality should also be taken into account when selecting the best SON peer. However, locality as a metric, by itself, can only achieve cost reduction. The geolocation, i.e. locality, used together with underlay network performance parameters such as bandwidth, jitter, and so on comprises a much better metric. The use of this metric does not only measure cost reduction but also the desirable performance for the services being offered in the context of a P2P SON.

These associated aspects coping with the best peer selection, lead to the need for decoupling the best peer selection process from the service business functions executed by the service or application. In other words, the best peer selection process should focus on several factors regarding the question of interaction between services in a P2P SON. These include the metric used as a parameter to select the best peers and the interaction of the best peer selection module with the other modules/services offered by the OMAN architecture.

The Best Peer Selection Service (BPSS) aims to fulfil the lack of generic and decoupled solutions for best peer selection, particularly in P2P SON. In this context, BPSS is generic since its architecture allows a particular metric to be employed for the selection of the best peer according to the needs of the services involved. To accomplish that, the decoupled BPSS module depends on the AgS in order to search all the peers that provide a particular service.

In the light of this, this chapter offers the specification and evaluation of a Best Peer Selection Service that is suitable for tackling the problem of selecting the best peer in a P2P SON, which is integrated with the general conception of the OMAN architecture. This chapter is structured as follows: Section 5.1 describes the BPSS model in terms of the parts involved and the interactions that occur to reach the best peers. Following this, the BPSS architecture is outlined in Section 5.2. The BPSS architecture is examined to show how the BPSS model is implemented regarding the modules from the OMAN architecture and the interactions between them. Next, Section 5.3 presents and discusses BPSS evaluation and the obtained results. Finally, Section 5.4 summarizes the entire chapter.

5.1 BPSS Model

The BPSS model designates the interacting entities that deal with the best peer selection process. This process includes searching all of the SON peers that provide the desired services. In addition, this model presents and discusses the roles of the parties and entities involved, and the eventually needed information that is exchanged between them in order to accomplish the selection process. Some of the entities also form a part of the AgS model, since both rely on Customers, and SON peers. Figure 5.1 depicts the BPSS model which is represented as a UML-like diagram.

The BPSS relies on the same aspects that AgS is based on. The role of best peer selection requester can be played by SON peers that are either third-party applications or services being executed outside the P2P SON. The decision about who will play this role at a particular moment depends on the application or service. If a SON peer needs to interact with another SON peer at the P2P SON, the best peer selection operation is triggered by a SON peer. On the other hand, when a third-party service or application wants to know which peer is the best peer to interact with, this role is played by an AgS peer through a provided interface. Whatever the situation, the knowledge of the best peer selection metric that is being used, should belong to the requester.

5.1.1 BPSS Model Description

The Service or Service Component object and the Best Peer are the central pieces in the BPSS model. They are part of the interactions with every subject in the model. The service object represents the Service or Service Component that actually executes on the SON peer and is published in the aggregation peers. Aggregation peers are the peers that form the AgS P2P overlay. The Best Peer object is a SON peer resulting from a best peer selection operation.

The Service Provider is another cornerstone piece in the BPSS model. It is the owner of the Service or Service Component as well as the SON peers and the AgS peers (since AgS peers are specialized SON peers). Service Providers are Customers as

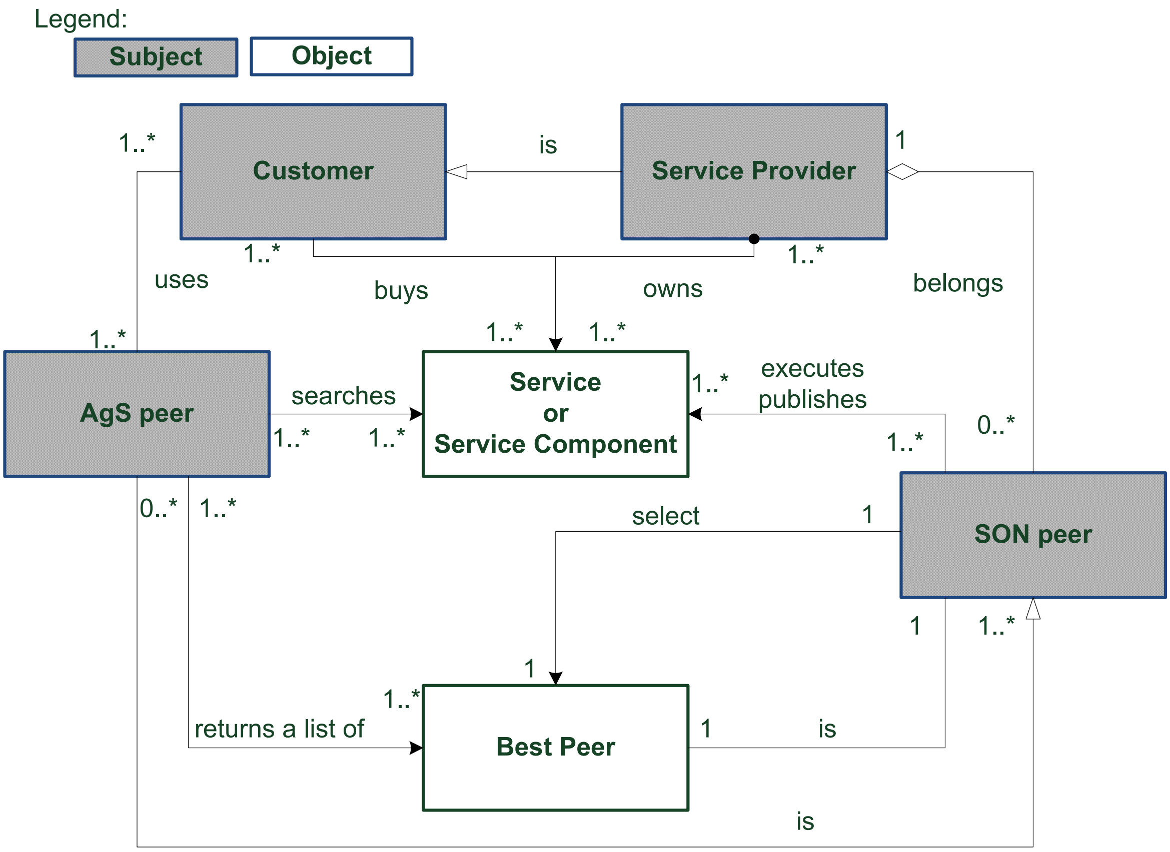


Figure 5.1: BPSS Model

well, so they can act as third-party consumers of service components of other service providers. They can also use third-party services in a service chain, so that they can offer a home user (customer) a complete set of services. However, before doing this, the service provider has to find the best peer with which its SON peer will interact, in order to be able to offer that set of services. Following the BPSS model, the Customer, which might be a third-party Service Provider, uses the AgS peer to search for services by using the AgS module. The AgS peer, in its turn, returns a list of all SON peers that execute the searched services. Ergo, the SON peer then selects the best one according to a metric.

The threefold role of the SON peer is underlined in the BPSS model which distinguishes between the following activities: execute, publish, and select. The first two activities are essentially concerned with handling the services and service components offered in the P2P SON, as already explained in Chapter 4, whereas the third activity comprises the best peer selection on behalf of a particular service. To sum up, all these actions, along with the action of search for services executed by the AgS peers, define the whole behaviour of the cycle of offering and search for services at the foreseen P2P SON. This modelling is implemented by means of a particular architecture that is outlined in Section 5.2.

5.2 BPSS Architecture

The BPSS architecture relies on the roles and interactions embodied in the BPSS Model. It is a highly flexible architecture composed of functionalities that are carried out between the already discussed[1] OMAN layers. The BPSS architecture is built on the third layer of OMAN. This means that the BPSS architecture uses functionalities provided by OMAN at the infrastructural and searching levels (i.e. first and second levels) to accomplish its goals.

The main goal of the BPSS architecture is to provide the best peer, (according to a particular metric), for a particular requester. The best peer will be a SON peer running a pairwise service. However, the requester might be an internal SON peer or an external P2P SON node/service.

Thus, even though BPSS architecture is designed comprising to serve the services and service providers in the P2P SON, external services can also be involved in choosing the best SON peer inside the P2P SON. This is the case when a particular external service wants to choose the best peer in a P2P SON, and employs as its best peer selection metric, the service's lowest price or another QoS parameter. In this case, the services concentration in a P2P SON leads to high odds concerning to find a competitive best peer.

The BPSS is designed as being a cornerstone module, at the OMAN architecture, which interacts with requesters and the AgS. Thus, the BPSS module plays the role

[1]See Chapter 3

of a proxy between applications/services and the lower layers of OMAN. On the one hand, it uses the AgS service, and on the other hand it offers a particular service (it provides best peer) to third-party users.

This module and its interactions are described in greater depth in the next Subsection.

5.2.1 BPSS Architecture Description

With regard to the use of OMAN, service developers can implement an interface with the BPSS module to request and receive best peer information, allowing the service business functions and best peer determination to be split up. As a result, this feature enables not only SON peers to do more than just start best peer selection requisitions. A door is left open to third-party developers to use OMAN to assist their own methods of best peer selection. This decoupling enhances modularity and best peer selection metric independence, thus leading to high flexibility when choosing the particular metric to use for a particular service type. BPSS takes advantage of this aspect by building on the AgS [Fiorese 10a] service when selecting the best peer.

Figure 5.2 illustrates the use of BPSS. SON peers can request best peer information (e.g. select_BP message), regarding a particular service, from the BPSS module. On the reception of a best peer request, the BPSS module asks the AgS service for the list of all the SON peers that have published a service profile for the intended service. After receiving the requested list, the BPSS module calculates the best peer and returns its reference to the best peer requester. The selection of the best peer is carried out by using one of the supported metrics.

In practice, the BPSS module should be implemented in the best peer selection requester. This is particularly suitable because it is responsible for the implementation of the best peer selection rules. Of course, these rules depend on a particular metric. Thus, the knowledge of how to select the best peer remains in the application or service, which is the natural candidate to perform this activity. Moreover, on the basis of the proposed OMAN and BPSS architectures, it is possible to have different services with different ways of allowing them to select their own best peers at the same P2P SON. The bonding that connects them rests on the assumption that the best peer selection will occur among the SON peers that execute the same services.

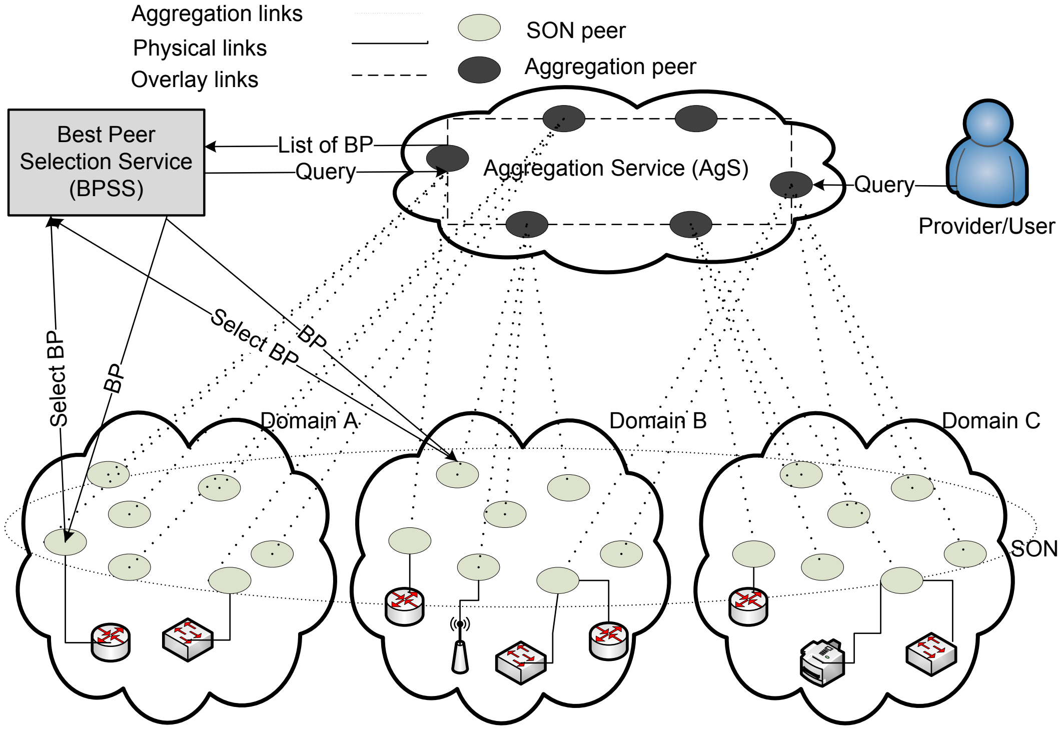


Figure 5.2: BPSS Architecture

It is worth mentioning that with this decoupled approach, it is also possible for an external entity (e.g. a user or a particular application/service or service component from outside the P2P SON) to request a best peer selection, as long as the request is compliant with the BPSS interface and the requested metric is supported.

5.2.2 BPSS Operation

The way BPSS works is based on the Select_BP operation. This operation copes with the procedure for starting the best peer selection. This operation is an event that is triggered on demand by the service running on the SON peer or by third-party services.

Several messages are exchanged between SON and aggregation peers to attain the BPSS goal. One of these messages is the Select_BP message. The other messages are part of the normal search of service that is outlined and explained in Chapter 4.

Figure 5.3 shows the message exchanges that are designed to achieve the best peer selection.

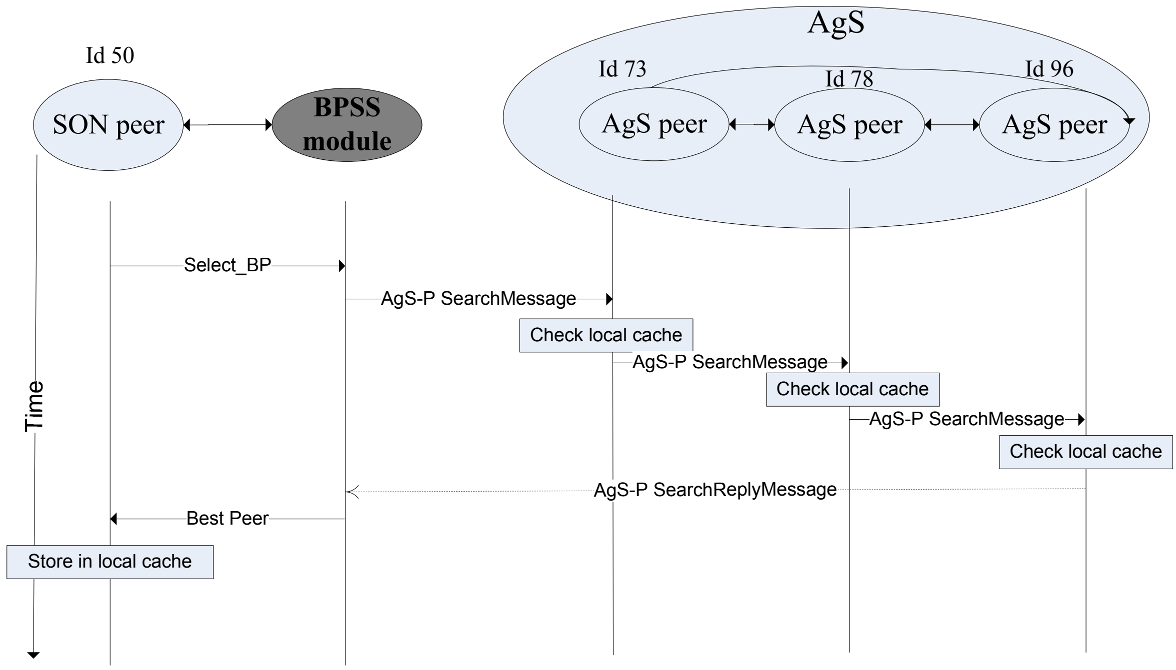


Figure 5.3: Best Peer Selection Operation (Select_BP)

The Select_BP message is very simple and just transports the service name or service keyword as the key for the searching process. Thus, when the BPSS module unfolds the Select_BP message, it uses the payload content as a parameter for the SearchMessage (Section 4.2.3.4).

5.3 BPSS Evaluation

This Section conducts an evaluation of the BPSS [Fiorese 11a, Fiorese 11d]. This evaluation comprises the BPSS analysis in terms of produced results (number of best peers) and also the interaction performance between the requester and the chosen best peer (quality of the choice).

As already explained, several characteristics can be taken into account in order to select a best peer. These characteristics depend on the service or application behaviour. For instance, in file sharing applications, the free-riding problem encourages the adoption of incentive mechanisms as part of the selection scheme. This kind of P2P application takes into account the fairness between uploads and downloads as a metric for the best peer selection [Huang 08].

However, file sharing applications are not the only ones available. OMAN and hence the BPSS are not designed to be compliant with just one type of service. In fact, BPSS is generic enough to support several different kinds of services. This support is based on the rationale that regardless of the kind of service provided, performance is a key issue for all kinds of services. On the basis of this belief and without loss of generality, the BPSS evaluation considers some characteristics associated with performance, as the best peer selection metric. Indeed, with few adjustments, this approach allows BPSS to be used as a first-level filter for all kind of services/applications. This means the best peer selection can be performed in a hierarchical manner, where the performance can be the filter at the very first level. In other words, the best peer can be selected among those which had been chosen with the best performance by another metric.

A particular evaluation framework was used to assess the BPSS. This framework is similar to the one used to evaluate the AgS, since BPSS executes on a P2P SON and depends on the AgS. Thus, several services and service component references running on a large scale multi-domain and made available by multiple service providers, were virtually deployed by simulation experiments. Thus, several simulations using real data were performed in order to select the best peers in scenarios with multiple service providers that were spread by multiple administrative and geographical domains.

Having in mind the stated, this Section is organized as follows: Subsection 5.3.1 provides a detailed description of the metric used for the BPSS evaluation. Subsection 5.3.2 outlines the BPSS evaluation framework and discusses how simulations were performed that take into account simulation set-up and strategies. Finally, the

BPSS evaluation results are analysed in Subsections 5.3.3 and 5.3.4 with the aim of showing evidence to support the answers to the research questions raised earlier about BPSS.

5.3.1 BPSS Metric

As already explained, the BPSS evaluation is carried out by means of a performance criterion to select best peers. This means that the selected best peer should offer optimized interaction with the requesting service/peer, in accordance with this performance criterion.

Several understandings about what performance means can be held when determining the best peer. However, the performance of P2P systems (including P2P SON) is often very sensitive to the delay characteristics of the underlying network. These are influenced, among other factors, by bandwidth, load and also geographic location. In fact, according to [Kaune 09], the geographic location of nodes heavily influences jitter and packet loss. This observation draws attention to the way the node's geographic location is used when developing a delay (performance) prediction model. Thus, the less delay, the better the performance. Bearing this in mind, Kaune et al. [Kaune 09] developed a predictive model for the Internet delay space that takes into account the geographical location of the nodes and the delay between them. Thus, using a rich set of real data, (namely measured end-to-end Round Trip Time (rtt) [mac 11] and measured end-to-end link jitter [pin 10]), these authors mapped the measured end-to-end nodes into a 5-dimensions Euclidean space model of the Internet, by combining these information with Global Network Positioning (GNP) [Lee 06] information.

Therefore, using the coordinates of each peer in this 5-dimensional model, which not only takes networking conditions into account but also peer location, it is possible to calculate the Euclidean distance between peers. As a matter of fact, this distance shows the delay and jitter between two particular geolocated nodes. All in all, two distance values that have the same point of origin, can be compared. This comparison makes it possible to measure which of the two nodes tested is closer to the point of origin. As a result, it can be claimed that, in this 5-dimensional model, the closer a SON peer is to the best peer selection requester, the more performance this peer can deliver to the interaction between them.

For the purposes of BPSS evaluation, the distance metric mentioned above is used to select the best peer for a particular best peer selection requester in accordance with the performance criteria.

The best peer selection quality, i.e. the best peers' behaviour, is also evaluated. In this case, the chosen metric depends on what kind of assessment is performed and performance was also chosen as the criterion for this evaluation. Thus, in this case, the service execution time that considers network data transfer between the best peer requester and the best peer, is used as the assessment metric. Therefore, service execution time allows, for instance, to check if the selected best peer actually behaves better than the selected second-best peer.

The BPSS evaluation was carried out by conducting several experiments using these metrics. There is a description of these experiments, including how they were performed and their results, in the next Subsections.

5.3.2 Evaluation Framework

A simulation study was conducted to evaluate the BPSS. This simulation study comprised the observation of the best peer and the second-best peer selection distributions over well-constrained geographical domains. Within these domains, the behaviour of an ftp-like service application was also assessed, regarding the interaction between a particular SON peer and its selected best and second-best peers.

The simulations performed for the BPSS evaluation only considered the scenario where the best peer selection request is triggered by a SON peer.

The simulator used in this study was the PeerFactSim.KOM [Kovacevic 07] discrete event simulator. This is the same simulator used in the evaluation of AgS in Subsection 4.3.2.

5.3.2.1 Simulations setup

The simulation environment was constituted of SON peers whose network identifiers (IP addresses) were taken from The Cooperative Association for Internet Data Analysis (CAIDA) initiative [cai 10] through its Macroscopic Topology Measurements project and from MaxMind®, Inc. GeoIP free database [max 10], which means the simulated peers belong to real geographical domains. However, it is worth noting, these data were compiled by Kaune et al. [Kaune 09], and integrated into the network layer that is implemented by the simulator used.

The modelling of the scenarios was based on real nodes available in the compiled data set. To accomplish that, peers belonging to specific geographical domains (country) were split between SON and aggregation peers. The number of AgS peers was 10% of the total number of SON peers used in each domain. Thus, if a country had 50 SON peers at the P2P SON - which, for instance, can mean that it comprised 50 service providers - then 5 AgS peers belonging to that country would form a part of the Aggregation Service (AgS). The chosen geographical domains were the following European countries: Portugal, Spain, France, Italy and Germany.

The simulation involved 11 sets of individual simulations. Each set simulated a particular number of SON and AgS peers. The initial set simulated a scenario with a total of 50 SON peers, corresponding to 10 randomly chosen SON peers from each of the domains. The second simulated set comprised a total of 75 SON peers (15 for each country). Thus, with steps of 25 SON peers between each simulated scenario, the last simulated set comprised 300 SON peers. On the other hand, the AgS service started with a total of 5 (1 per country) AgS peers. However, for the AgS service, the increasing step of 1 AgS peer per country only applies when the number of SON peers forming the P2P SON is even. Thus, at the scenarios where the P2P SON is constituted of 75, 125, 175, 225, and 275 SON peers, the total number of AgS peers corresponds to 10% of the previous number of SON peers.

The wide range of simulated P2P SON sizes is designed to cover scenarios with few service providers (e.g. small P2P SON for very specialized services) to scenarios composed of many service providers (e.g. a more competitive scenario).

5.3.2.2 Simulations Strategy

Some aspects of AgS have to be disclosed to simulate the BPSS execution, since AgS underpins BPSS. These aspects are related with the publishing of services because without the published services, SON peers are not able to select best peers comprising any service. The services looking up are another aspect regarding the same reason.

For the sake of simplicity and without loss of generality, a particular SON peer could only publish, at most, seven randomly chosen services or service components (using a uniform distribution) from the service set S={S1,S2,S3,S4,S5,S6,S7}. In addition, each SON peer could only publish its service subset on, at most, 10 distinct randomly-chosen aggregation peers (also following a uniform distribution).

To accomplish the best peer selection, searching service using the AgS is needed. Regarding this aspect, the search concludes when the SearchMessage arrives, after circulating the AgS logical ring, at the search requester AgS peer. At this point, the SearchMessage (see 4.2.3.4 and B.5) should carry the list of all the SON peers that run the searched service.

The BPSS simulations comprised several experiment executions, each of which simulated 50 hours of work and this was repeated 10 times to obtain average values. Every operation (e.g. joining, leaving, publishing, searching, and select) was specified in time. Each simulation executed 100 searching operations, i.e. 100 best peer selection requests, since a searching operation is triggered by a best peer selection request.

The SON peers that execute this selection request, are randomly chosen using a uniform distribution. This takes place when the number of SON peers composing the P2P SON, is greater than the 100 best peer selection operations, which composes each experiment. Otherwise, each SON peer executes the best peer selection operation at least once.

The performed simulations also involved selecting second-best peers. They are the peers whose performance measured by the used metric (distance), put them in second place in a hypothetical ordered list of best peers. The second-best peers are selected from the same unordered list of SON peers provided by the AgS service, by removing the best peer from that list and repeating the measurement process.

The determination of the second-best peer can assist in the validation of the used metric in two ways: 1) by checking if the service behaves better with the selected best peer - this provides a measure of the metrics' consistency; 2) by measuring the average improvement of the best peer over the second-best peer; in this way, an indication of the effectiveness of the metrics can be obtained.

The interaction between the requesting SON peer and the SON peers selected as the best and the second-best peer, was also evaluated. To accomplish that, an ftp-like service was implemented and executed allowing an interaction between the best peer requester and its best peer. The service execution time during this interaction was gathered so that it could be compared with the same interaction process though using the selected second-best peer.

The simulations considered upload and download speeds of 100 and 200Mbps, respectively. A dynamic latency model for packet transmissions was used. The geographic location module developed in [Kaune 09] handled the transmission delays, in accordance with the measured rtt and jitter and the geographical location of the peer. This resulted in near real transmission delays, packet losses and datagram timeouts.

Two sets of simulations were performed. The first set comprised simulations that were designed to obtain results for the best and second-best peer selection distribution over the geographical domains referred to earlier. The second set of simulations copes with the assessment of the quality of the best peer selection. The next Sections include an analysis of the simulation results.

5.3.3 BPSS Evaluation by Geographical Distribution of the Best Peers

These simulations assess the following question:

- What is the distribution of the chosen best and second-best peers over the used geographical domains?

In order to answer this question, the evaluation framework discussed earlier was used in simulations. The set of eleven simulations comprising Portugal, Spain, France, Italy and Germany as geographical domains yielded the best and second-best peer selection distribution results, which are shown in Figure 5.4 and Figure 5.5, respectively. Both results are for requests made by SON peers that are only belonging to the Portuguese geographical domain, though it is obvious that the methodology employed can be applied to any other geographical domain.

Figure 5.4 shows the geographical location distribution of the SON peers selected as best peers. There are eleven 5-bar clusters, each one corresponding to one of the eleven simulated scenarios. In every cluster, each of the five bars represents the number of best peer occurrences in each of the five geographical domains, namely Portugal, Spain, France, Italy, and Germany, respectively.

One can expect the highest number of selected best peers to be in the domain of the requester (Portugal, in the case of these simulations), due to considerations of geographical distance. However, the obtained results clearly show the effects of two key aspects of the OMAN architecture and of its AgS and BPSS services: on the one hand, in some cases, search of services performed by the AgS service found that the desired services were not available at any of the SON peers of the requester's domain; on the other hand, the metric used by the BPSS service - based on the Internet model proposed in [Kaune 09], which takes into account both the geographical location and delay and jitter - led to the fact that the closest peer, in terms of the 5-dimension Euclidean distance, lay in a different geographical domain.

Similar results based on the same explanations are obtained regarding the second-best peers, which are shown in Figure 5.5.

Despite this, even with the mentioned constraints regarding the statistical availability (or unavailability) of the desired services in the requester's domain, the average results of all the simulations show that the highest number of best peers was

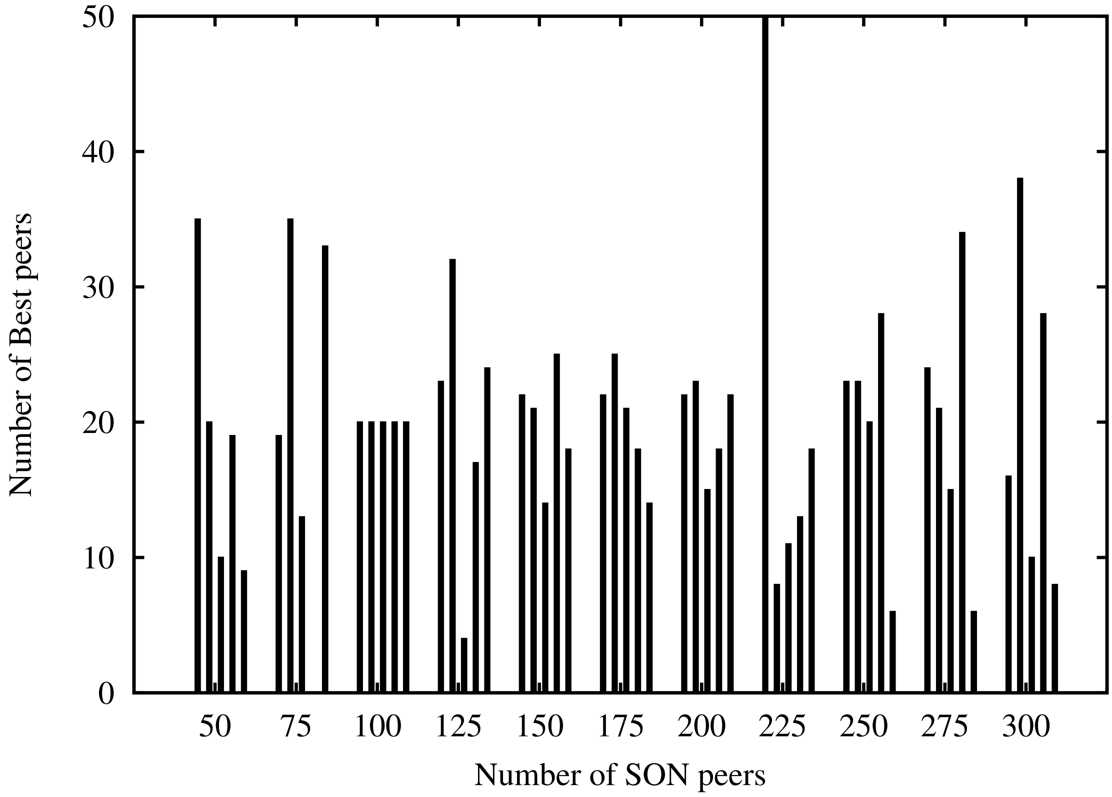


Figure 5.4: Clustering Best Peer Occurrences by Domain and by Number of SON peers

selected in the same domain of the requester. This can be seen in Figure 5.6 and Figure 5.7. It is worth mentioning the results rely on a confidence interval of 95% for the mean number of best peer selections in each geographical domain.

When the best peers are added to the second-best peers in the domain, the SON peers in the requester's domain (Portugal) were selected as best peers for 27% of the time, followed by Spain (22.5%), Italy (22%), Germany (14.5%) and France (13%). This means that almost half of the best peer selections resulted in peers belonging to the same geographical domain or to the neighboring geographical domain. This suggests the consistency of the used metric and the success of OMAN's AgS and BPSS services.

It is also worth mentioning that the BPSS overhead is negligible. The time taken by the selection process for both the best and second-best peer is around 2ms. On the

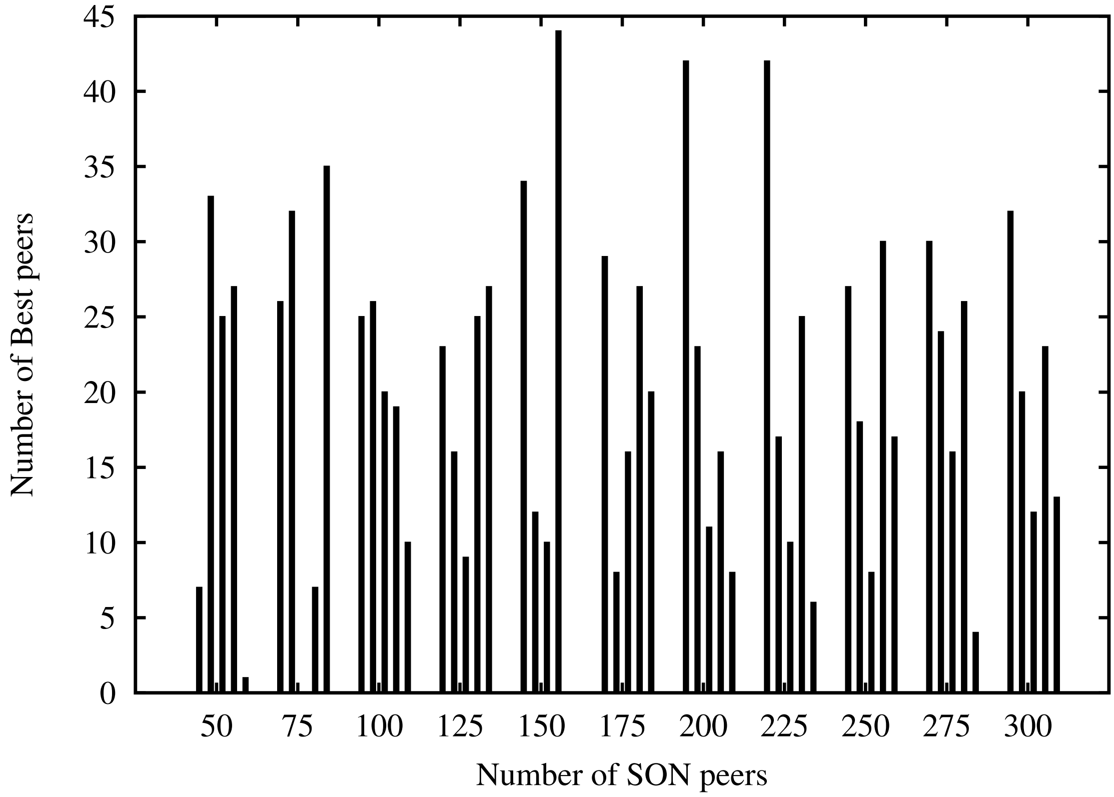


Figure 5.5: Clustering Second-Best Peer Occurrences by Domain and by Number of SON peers

other hand, the overhead associated with the maintenance of the P2P SON and with the AgS service is approximately 164ms. These values strongly suggest the efficiency, feasibility, and practicality of the proposed BPSS service, in conjunction with AgS and the other modules of the OMAN architecture.

5.3.3.1 Summary

Section 5.3.3 comprised the BPSS evaluation analysing the distribution of the best and second-best peer selection over five distinct geographical domains. A particular distance metric was used to carry this out. The used metric, proposed in [Kaune 09], combines measured delay and jitter data with geographical location data.

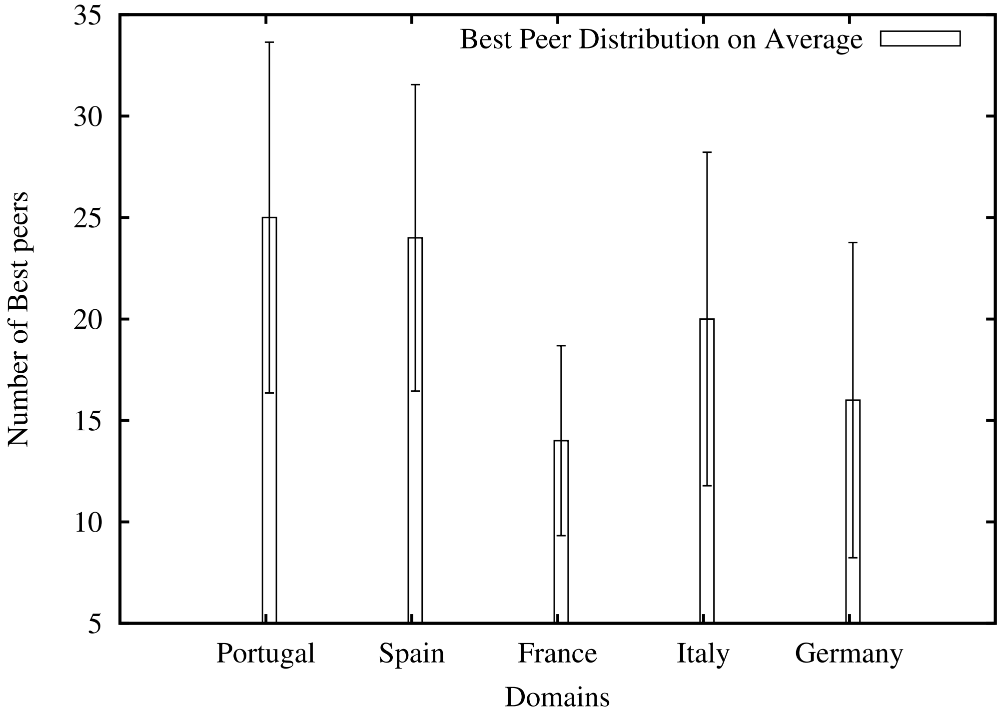


Figure 5.6: Best Peer Distribution by Geographical Domain

The obtained results showed that best peer selection performs well over distinct geographical domains. As expected, on average the best and second-best peer distribution leads to higher rates in the same geographical domain than the best peer selection requester. In other words, for requesters belonging to Portugal, 27% of the requisitions selected best and second-best peers belonging to Portugal.

5.3.4 BPSS Evaluation through Performance of the Best Peers

This evaluation measures and analyses service performance for the selected best and second-best peer. The measuring and comparisons to be made are a methodological means of answering the following questions:

- Does the best peer selection reflect an improvement in the performance of the interaction of an observed service?

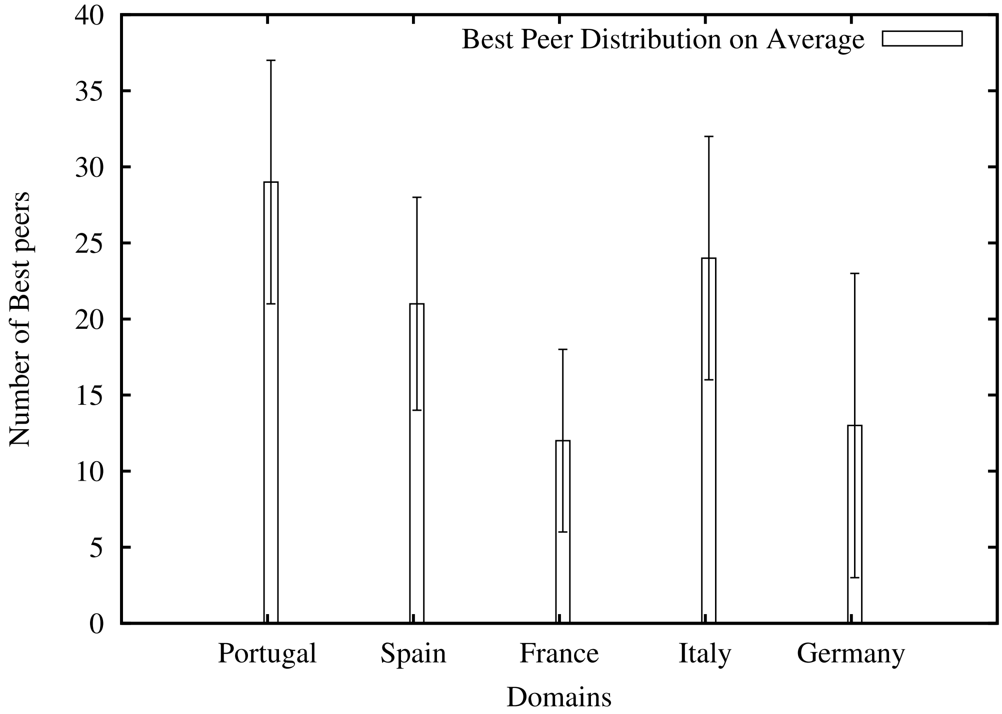


Figure 5.7: Second-Best Peer Distribution by Geographical Domain

- What is the difference between the best and the second-best peer in terms of the performance of the service interaction?

Although the BPSS chooses best peers in accordance with the distance metric, it still lacks confirmation of advantages for the services that are executing on the P2P SON. These advantages can be measured by taking the service execution performance as a metric. As already explained, the considered performance testing is based on the service execution time regarding network data transfer (interaction) between a particular best peer requester and its chosen best and second-best peers.

The average service execution time for each simulated scenario was calculated through measurements that considered the difference between the simulation time at the end and at the beginning of the service interactions. Equation 5.1 shows the formula for the average service execution time (T_A).

$$T_A = \frac{\sum_{s=1}^{n} T_e - T_b}{n} \tag{5.1}$$

Therefore, after selecting the best and the second-best peers, an ftp-like service was run between these peers and the requesting peer to check if the choice was made in a consistent way. This kind of service was chosen because it fits a class of services that intensely use the communication channel, such as P2P video streaming applications, for instance. The service required the transmission of 10MB of data during its execution. The network parameters are explained in Section 5.3.2.2.

The ftp-like service was immediately triggered after the best and second-best peer selection. The TCP protocol is used for sending datagrams. Best peer and second-best peer interactions were executed concurrently. The execution time for each of the interactions, i.e. with the best peer and with the second-best peer, was measured distinctly. The average values within a 95% confidence interval are plotted in Figure 5.8.

Figure 5.8 shows that the service execution time for interactions with the chosen best peer is lower than the service execution time of second-best peer interactions most of the time. Taking the arithmetic mean of the execution time of all the experiments for best peer and second-best peers, service execution time with the best peer is approximately 5% lower than with the second-best peer. Additionally, Figure 5.8 shows similar uphill and downhill tendencies for both plotted service execution times, the exceptions being the 125 and 200 SON peer sizes. This suggests there is a high correlation between the best and second-best chosen peers.

On the other hand, it should be emphasized that the values presented in Figure 5.8 are also an indicator of the effectiveness of the used distance metric, and that the BPSS service and the underlying P2P SON, for that matter, work equally well with other metrics. In fact, BPSS can be used by the cooperating P2P SON service providers in order to fine-tune any performance metric and adapt it to their overall goals.

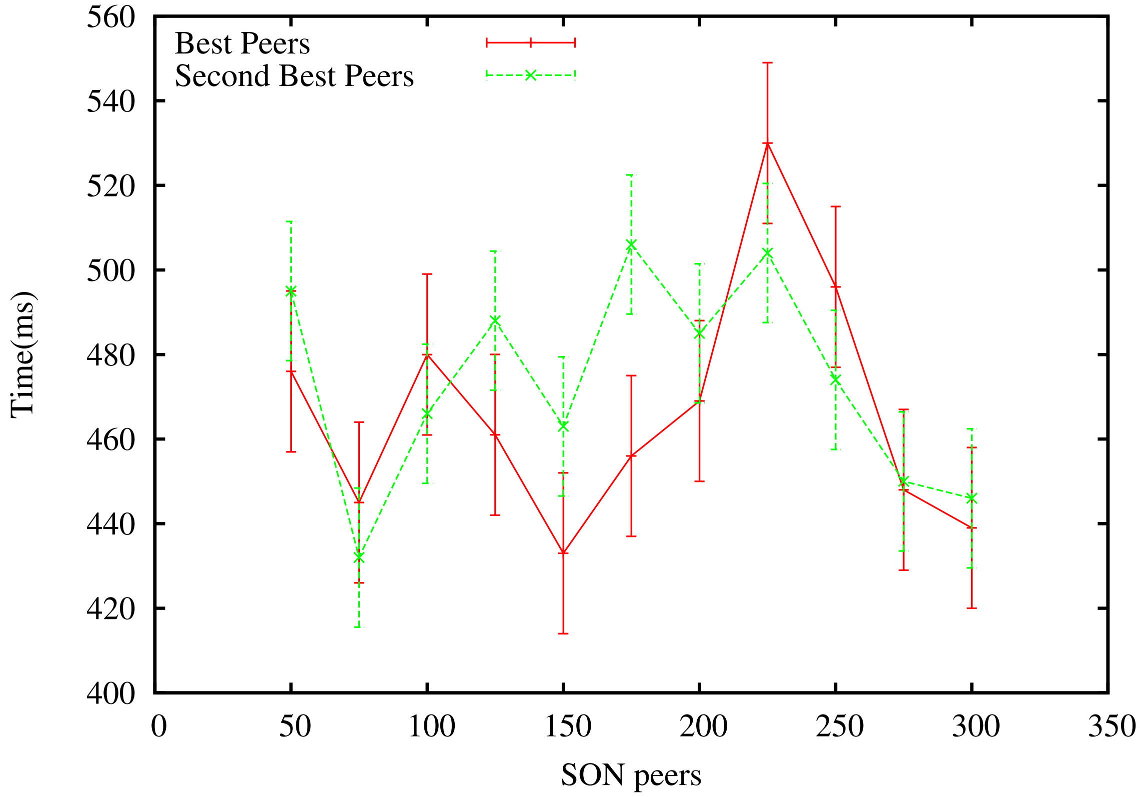


Figure 5.8: Service Execution Performance

5.3.4.1 Summary

Section 5.3.4 comprised to assess service performance by testing the BPSS best and second-best selected peers. To accomplish that, an ftp-like service was run between the best peer requester (ftp-like service requester) and the chosen best and second-best peers. The ftp-like service execution transferred 10MB of data using TCP protocol and a connection with 100 and 200Mbps for upload and download, respectively. The same simulated environment was employed to select best and second-best peers and to perform these experiments. The service execution time was used as a metric for these experiments. This metric is based on the time taken to transfer 10MB of data from the best and second-best peer to the best peer requester.

Several experiments were conducted to find out the average service execution times. These experiments were carried out in two environments: 1) service requester and best peers, and 2) service requester and second-best peers. Thus, when the results for both these categories are averaged, the numbers show that the ftp-like service execution time for the selected best peers is 5% lower than for the second-best peers.

Thus, these results reveal, (as expected), that the interactions between requesters and selected best peers show an improvement. This finding is important because it gives credibility to using the distance metric to choose the best and second-best peers. Moreover, these results also provide evidence that the performance is a good metric for assessing services in an environment like P2P SON.

5.4 Summary

In this Chapter, there has been a study and evaluation of the Best Peer Selection Service (BPSS). The BPSS model and its architecture were presented and discussed. An attempt has been made to show how BPSS, which was integrated in OMAN, can employ a decoupled method to select best peers independently of the service/application. Initially, BPSS allows a consumer (e.g, service provider, final user, etc.) to select the best peer to interact with. In order to accomplish this, BPSS needs to know the service or application best-peer selection requirements, i.e. the best peer selection metric. Furthermore, as well as the selection metric, BPSS uses the underlying OMAN services to carry out the selection. These characteristics allow the service/application business rules to be decoupled from the best peer selection.

In its first stage, BPSS uses the AgS only to return the SON peers that are running the requested service. After this, BPSS performs the best peer selection by using the adopted metric.

Nonetheless, for the sake of conducting an evaluation of BPSS, performance has been chosen as the metric for best peer selection. This metric was chosen regarding the wide range of services that take this parameter into account in their interactions. Thus, this Chapter considers performance based on the distance metric described in [Kaune 09] as the best peer selection metric. Moreover, it also considers service execution time as a performance metric to assess the best and second-best peer selection results.

In addition, this Chapter has provided an analysis of the BPSS model and architecture by examining its composition, usage and advantages. Furthermore, it has attempted to validate BPSS by carrying out several experiments. This validation has been put forward in two axes: 1) Best and second-best peer distribution over several particular geographical domains; 2) Performance of the selected best and second-best peers.

The experiments (and hence the results) comprising best and second-best peer distribution took place in five distinct real European geographic domains. These domains were: Portugal; Spain; France; Italy; and Germany, although the best peer requesters were constrained to Portugal. This "constraining" was designed to act as a reference point for the results. Thus, experiments were conducted to find the best and second-best peer distribution in these domains. Real data has been used in the experiments and the results show that the best peer distribution is as follows: Portugal (27%) of the best peer selections, followed by Spain (22.5%), Italy (22%), Germany (14.5%), and France (13%).

The selected best and second-best peers performance was also evaluated. An ftp-like service (executing in both the requester and best/second-best peers) was used to perform these experiments. Thus, after selecting the best and second-best peer, the aforementioned service was started. The service sends 10MB of data by means of the TCP/IP protocol suit. The performance was assessed concerning the service execution time. This metric corresponds to the time taken to transmit all the data from the best or second-best peer to the best peer requester. The simulated transmission process takes into account the virtually real network conditions. These conditions are related with the distance metric used to select the best peers. Thus, this relation allows the quality of the selections to be assessed. The results show that on average the service execution time between selected best peers and the best peer requester performs 5% better than with the selected second-best peers. This finding confirms the expectations about the quality of selections.

Chapter 6

Conclusions and Future Work

This thesis addresses the problem of providing a P2P SON environment coping with the services offering market on the Internet. In particular, it faces issues regarding the searching services and best service provider selection in P2P SON environments. This final Chapter outlines the main conclusions of this work and mentions some still open issues that could be the subject of further studies. Therefore, Section 6.1 summarizes this thesis. Section 6.2 describes the main contributions of this work. Finally, Section 6.3 points out some issues that need to be explored in future work.

6.1 Synthesis of the Thesis

This thesis has sought to outline a proposal for a multi-provider environment for services offering on the Internet. This environment is modelled by the OMAN architecture and mainly implemented by AgS and BPSS modules. It can support providers that offer their services in an on-demand fashion beyond their administrative boundaries and on a cost-sharing basis. OMAN provides two particularly useful modules/services for tackling issues regarding searching the offered services; and for selecting the best peer provider of a service, in case there are several providers available for a particular service. These two services are the AgS and BPSS, respectively.

AgS optimizes the search of services by using an aggregation mechanism that takes into account the replication of already found service references by exploring the dense distribution of service publishing information in the AgS overlay (AgS was discussed in detail in Chapter 4). As well as providing benefits for both customers and service providers, it concentrates service exposure in a P2P SON by allowing customers to carry out quick service searches and service providers to reduce operational costs since they do not need to build or lease their own SON. This approach is particularly well suited to small service providers, which focus on market niches, as can be seen in the evaluation results for the AgS, which are given in Chapter 4.

This is the case for the BPSS, which is presented in Chapter 5. BPSS allows the selection of the best peer (best service provider) for which a requested service is being run. Thus, if the best peer requester is regarded as a peer representing a customer or a third party service provider, BPSS brings about an improvement in the performance of the interaction between the requesting best service provider and the selected one, when they are exchanging data in order to run the service. The best peer selection is carried out with the support of the AgS, which for this purpose returns all the potential best service providers that make the desired service available. In its turn, BPSS uses a performance-like metric to compare all of them in order to choose one. The one that shows the best values for that metric is chosen as the best. The performance-like metric takes into account underlay network performance parameters (e.g. delay, jitter, rtt) and the geolocation of the nodes. A simulated evaluation of BPSS is also presented in Chapter 5.

The use of AgS and BPSS were simulated in several scenarios and situations. These simulated evaluations were discussed in the appropriate sections of the Chapters 4 and 5. The examination of these simulation results shows that the approaches adopted by OMAN and its AgS and BPSS modules achieve the proposed objectives: the construction of a multi-provider environment for services offering based on P2P SON.

6.2 Main Contributions

This Section describes the main research contributions of this thesis.

An architecture to provide an environment based on P2P SON for services offering in the Internet

This thesis has outlined a modular architecture named OMAN that is designed to cope with the construction of a multi-provider environment for services offering. OMAN accomplishes that by providing a modular approach in three layers which when combined are responsible for provisioning the mentioned environment. The three layers of OMAN consists of: a) the cornerstone first layer that the environment for services offering relies on; b) the basic and optimized mechanisms for services searching, which include AgS; c) the best service provider selection mechanism; among others. This architecture helps service providers reduce their operational costs by sharing them with other service provider partners and gives customers the opportunity to find other services in accordance with their particular requesting parameters, and apart from those on the portfolio of their Internet access provider.

Flexible representation of services

A service profile that represents services has been made available by OMAN architecture. A service profile is an XML file that is supplied by service providers with the necessary information about a particular service. Service profiles are published in order to offer services and they bind business rules of service providers to customers' requests. They can flexibly represent several different kinds of services that can be offered and bought in the P2P SON environment. Although the flexible representation allows service providers to hide crucial information about a particular service, service providers must offer enough information to attend the requesting parameters of potential customers.

An optimized service searching mechanism

The AgS provides a dynamic means of addressing the problem of searching and selecting service or service components in P2P SON environments. An AgS model has been set out and discussed together with the developed AgS architecture. The search algorithm is also examined and the utilization of Aggregation Service Protocol (AgS-P) is highlighted. The use of AgS results in a particular service provider reference (SON peer). This reference is based on the service that is being searched. It is obtained by comparing the requester's service requirements with the service parameters that are published to select the peers that are most suitable for providing the service. The adoption of this approach means that the service or service component can be found, regardless of the class of service, since the parameters are compared individually during the search process.

Proposal of a service searching protocol for P2P SON

The development of AgS gave rise to the need for optimized communication between peers in order to perform an optimized service searching and for this reason the AgS-P was developed. AgS-P is used to exchange data information between AgS peers and between SON and AgS peers by employing underlying Internet transport protocols to send and receive particular application messages.

A best service provider selection mechanism

The BPSS adopts a decoupled approach to select best service providers (best peers) independently of the service/application. BPSS allows a consumer (e.g. service provider, final user, etc.) to select the best peer to interact with by using a best peer selection metric. Furthermore, besides the selection metric, BPSS relies on the underlying AgS service to carry out the selection. These characteristics allow the service/application business rules within the application to be decoupled from the best peer selection process.

Customers can send a particular message to a peer that executes the BPSS service requesting a best peer reference regarding a particular service. The details, such as algorithms employed, are shown as well as the description of the BPSS model and architecture. The validation of the best service provider choice (best peer) was also undertaken. A requesting peer used an ftp-like service parameter to select the best peer to interact with. This approach has resorted to carrying out an evaluation of the whole cycle of peer selection.

Simulation and validation of the optimized service searching and the selection of the best service provider mechanisms

The proposed approaches for AgS and BPSS were evaluated by means of a P2P network simulator. This evaluation involved several simulations to cope with the following: performance and scalability; analysis of the performance regarding data replication; and search efficiency *versus* AgS dimensioning factors. In addition, several simulation rounds were performed comprising geographical best peer distribution, and reflecting the performance of the chosen best peers.

Simulation results demonstrated that AgS and BPSS have good efficiency for searching services and selecting best peers, respectively. This efficiency also demonstrates the great potential of OMAN as an architecture for handling offerings of services in a Peer-to-Peer (P2P) Service Overlay Networks (SON).

6.3 Future Work

Apart from its research contributions, this study has raised some issues that require further research and work. A series of recommendations are made here to improve the OMAN modules and the overall usage of the OMAN in a P2P SON environment.

Thorough implementation of OMAN architecture

Despite the observed great potential of the overall OMAN architecture, some peripheral modules were not implemented in the simulations, such as Overlay Monitoring (OM) and Configuration Manager (CM). These OMAN third layer modules can provide mechanisms to improve the performance of P2P SON. Since this thesis rests on the assumption that there is a freely expressed desire for a self-organized service providers consortium, the simulation of the remainder of the undeveloped OMAN modules can help to evaluate the influence of these modules on the relationship between the service providers in this kind of P2P SON environment.

Interfacing for third-party legacy systems

Although the OMAN architecture by itself foresees the need for partners systems to provide full coverage and usage of an environment for services offering, the Support for Management Systems module does not specify a particular interface for data exchange. Further effort is needed to make OMAN compliant with external systems, such as billing, Service Level Agreement (SLA) enforcement at the P2P SON provisioning, security, etc.

Service profile enhancement

This thesis is based on the assumption that all the providers agree to a standard representation of services' profiles, such as employing the same representation for parameter values. However, a more extensive study of the best means to represent this information needs to be carried out. For instance, ontology-based semantic annotations could be used as a means of creating flexible, and yet standardized representations.

Consistency mechanism

The services offering environment provided by OMAN relies on the service's profile publishing. This publishing occurs when the service profiles are sent by SON peers and stored in local caches of AgS peers. AgS and BPSS make use of this fact to play their roles. In any case, the service providers can stop offering a particular service whenever they need. However, currently, both AgS and BPSS lack a mechanism for checking if their results are consistent, which means determining whether the service provider is still offering the requested service. Further work is planned to address this issue. Considering scenarios with a high number of SON peers involved, a promising approach to tackle this problem is to extend the AgS-P. In this way, the checking could be performed on demand during the searching process. This approach can avoid underlying network overload through a periodically polling solution, at the same time as providing a simple and elegant way of maintaining consistency between the results and the service providers' offerings.

References

[Aberer 05a] Karl Aberer, Luc Onana Alima, Ali Ghodsi, Sarunas Girdzijauskas, Seif Haridi & Manfred Hauswirth. *The Essence of P2P: A Reference Architecture for Overlay Networks.* In Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing, P2P '05, pages 11–20, Washington, DC, USA, 2005. IEEE Computer Society.

[Aberer 05b] Karl Aberer, Anwitaman Datta & Manfred Hauswirth. *P-Grid: Dynamics of Self-Organizeing Process in Structured Peer-to-Peer Systems.* In Ralf Steinmetz & Klaus Wherle, editeurs, Peer-to-Peer Systems and Applications, volume 3485 of *Lecture Notes on Computer Science*, pages 137–153. Springer, 2005.

[Adami 09] Davide Adami, Christian Callegari, Stefano Giordano, Gianfranco Nencioni & Michele Pagano. *Design and performance evaluation of service overlay networks topologies.* In Proceedings of the 12th international conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems, SPECTS'09, pages 296–303, Piscataway, NJ, USA, 2009. IEEE Press.

[Afsarmanesh 05] Hamideh Afsarmanesh & Luis Camarinha-Matos. *A Framework for Management of Virtual Organization Breeding Environments.* In Luis Camarinha-Matos, Hamideh

Afsarmanesh & Angel Ortiz, editeurs, Collaborative Networks and Their Breeding Environments, volume 186 of *IFIP International Federation for Information Processing*, pages 35–48. Springer Boston, 2005.

[Aggarwal 08] V. Aggarwal, O. Akonjang & A. Feldmann. *Improving user and ISP experience through ISP-aided P2P locality.* In INFOCOM Workshops 2008, IEEE, pages 1–6, 2008.

[aka 12] *Akamai.* http://www.akamai.com, 2012.

[Andersen 01] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek & R. Morris. *The case for resilient overlay networks.* In Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on, pages 152–157, may 2001.

[Auer 08] Lukas Auer, Christine Strauss, Natalia Kryvinska & Peter 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, pages 535–539, Barcelona, Spain, March 2008. IEEE Computer Society.

[Banaei-Kashani 04] Farnoush Banaei-Kashani, Ching chien Chen & Cyrus Shahabi. *WSPDS: web services peer-to-peer discovery service.* In Proceedings of the International Conference on Internet Computing, pages 733–743, 2004.

[Belqasmi 08] F. Belqasmi, R. Glitho & R. Dssouli. *Ambient network composition.* IEEE Network, vol. 22, no. 4, pages 6–12, August 2008.

[Bernstein 03] Daniel S Bernstein, Zhengzhu Feng, Brian Neil Levine & Shlomo Zilberstein. *Adaptive Peer Selection.* In In Proceedings of the 2nd International workshop on Peer-toPeer Sustems (IPTPS03), 2003.

[Beverly Yang 03] B. Beverly Yang & H. Garcia-Molina. *Designing a super-peer network*. In 19th International Conference on Data Engineering, 2003. Proceedings, pages 49– 60. IEEE, March 2003.

[Bianchini 09] Devis Bianchini, Valeria De Antonellis & Michele Melchiori. *Service-based semantic search in P2P systems*. In Web Services, European Conference on, volume 0, pages 7–16, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[Binzenhöfer 06] Andreas Binzenhöfer, Kurt Tutschku, Björn Graben, Markus Fiedler & Patrik Arlos. *A P2P-Based Framework for Distributed Network Management*. In Wireless Systems and Network Architectures in Next Generation Internet, volume Volume 3883/2006 of *Lecture Notes in Computer Science*, pages 198–210. Springer Berlin / Heidelberg, 2006.

[Brown 09] Alan Brown, Mario Kolberg & John Buford. *Chameleon: an adaptable 2-tier variable hop overlay*. In Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference, pages 770–775, Las Vegas, NV, USA, 2009. IEEE Press.

[cai 10] *The Cooperative Association for Internet Data Analysis (CAIDA)*. http://www.caida.org/, 2010.

[Camarinha-Matos 05] Luis Camarinha-Matos, Hamideh Afsarmanesh & Martin Ollus. Virtual organizations: Systems and practices. Springer, 2005.

[Capone 08] A. Capone, J. Elias & F. Martignon. *Models and Algorithms for the Design of Service Overlay Networks*. IEEE Transactions on Network and Service Management, vol. 5, no. 3, pages 143–156, September 2008.

[Choffnes 08] David R. Choffnes & Fabián E. Bustamante. *Taming the torrent: a practical approach to reducing cross-isp traffic* *in peer-to-peer systems.* In ACM SIGCOMM Computer Communication Review, volume 38, pages 363–374, New York, NY, USA, August 2008. ACM.

[Chu 00] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan & Hui Zhang. *A Case for End System Multicast.* In Proceedings of ACM Sigmetrics, pages 1–12, 2000.

[Clarke 01] Ian Clarke, Oskar Sandberg, Brandon Wiley & Theodore Hong. *Freenet: a distributed anonymous information storage and retrieval system.* In Designing Privacy Enhancing Technologies, volume 2009/2001 of *Lecture Notes in Computer Science*, pages 46–66. Springer, 2001.

[Crespo 05] Arturo Crespo & Hector Garcia-Molina. *Semantic Overlay Networks for P2P Systems.* In Gianluca Moro, Sonia Bergamaschi & Karl Aberer, editeurs, Agents and Peer-to-Peer Computing, volume 3601, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[Curbera 02] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi & S. Weerawarana. *Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI.* IEEE Internet Computing, vol. 6, no. 2, pages 86 –93, April 2002.

[Dabek 03] Frank Dabek, Ben Y. Zhao, Peter Druschel, John Kubiatowicz & Ion Stoica. *Towards a Common API for Structured Peer-to-Peer Overlays.* In Peer-to-Peer Systems, pages 33–44, 2003.

[Ding 09] Jianguo Ding, Ilangko Balasingham & Pascal Bouvry. *Management of Overlay Networks: A Survey.* In Mobile Ubiquitous Computing, Systems, Services and Technologies, International Conference on, volume 0, pages 249–255, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[Domingue 11] John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, Dave Lambert, Frances Cleary, Petros Daras, Srdjan Krco, Henning Müller, Man-Sze Li, Hans Schaffers, Volkmar Lotz, Federico Alvarez, Burkhard Stiller, Stamatis Karnouskos, Susanna Avessta & Michael Nilsson, editeurs. The future internet: Future internet assembly 2011: Achievements and technological promises, volume 6656 of *Lecture Notes on Computer Science*. Springer, 1 edition, 2011.

[Dong 04] Xin Dong, Alon Halevy, Jayant Madhavan, Ema Nemes & Jun Zhang. *Similarity search for web services.* In Proceedings of the Thirtieth international conference on Very large data bases - Volume 30, pages 372–383, Toronto, Canada, 2004. VLDB Endowment.

[dos Santos 08] Carlos Raniery Paula dos Santos, Luis Fernando Dalla Santa, Clarissa Cassales Marquezan, Sérgio Luis Cechin, Ewerton Monteiro Salvador, Lisandro Zambenedetti Granville, Maria Janilce Bosquiroli Almeida & Liane Margarida Rockenbach 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, pages 2057–2062, Fortaleza, Ceara, Brazil, 2008. ACM.

[Duan 03] Zhenhai Duan, Zhi-Li Zhang & Yiwei Thomas Hou. *Service overlay networks: SLAs, QoS, and bandwidth provisioning.* IEEE/ACM Transactions on Networking, vol. 11, no. 6, pages 870–883, December 2003.

[Dustdar 05] Schahram Dustdar & Wolfgang Schreiner. *A Survey on web services composition.* International Journal of Web and Grid Services, vol. 1, no. 1, pages 1–30, August 2005.

[Edwards 06] W. Keith Edwards. *Discovery Systems in Ubiquitous Computing*. IEEE Pervasive Computing, vol. 5, no. 2, pages 70–77, April 2006.

[Eriksson 94] Hans Eriksson. *MBONE: the multicast backbone*. Commun. ACM, vol. 37, no. 8, pages 54–60, August 1994.

[Fan 06] J. Fan & M. H. Ammar. *Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies*. In INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pages 1–12, 2006.

[Fiorese 09] Adriano Fiorese, Paulo Simões & Fernando Boavida. *Um Serviço de Agregação Baseado em P2P para a gestão de redes e serviços*. In Actas da 9^a Conferência sobre Redes de Computadores (CRC 2009), Oeiras, Portugal, November 2009.

[Fiorese 10a] Adriano Fiorese, Paulo Simões & F. Boavida. *An aggregation scheme for the optimisation of service search in Peer-to-Peer overlays*. In Network and Service Management (CNSM), 2010 International Conference on, pages 481–486, Niagara Falls, Canada, October 2010.

[Fiorese 10b] Adriano Fiorese, Paulo Simões & Fernando Boavida. *Avaliação de Desempenho da Busca de Serviços usando Agregação em P2P Service Overlay Networks*. In Actas da 10^a Conferência Sobre Redes de Computadores (CRC 2010), pages 193–195, University of Minho, Braga, Portugal, November 2010.

[Fiorese 10c] Adriano Fiorese, Paulo Simões & Fernando Boavida. *OMAN – A Management Architecture for P2P Service Overlay Networks*. In Mechanisms for Autonomous Management of Networks and Services, volume 6155 of *Lecture Notes in Computer Science*, pages 14–25, Zurich, Switzerland, June 2010. Stiller, Burkhard; De Turck, Filip.

[Fiorese 10d] Adriano Fiorese, Paulo Simões & Fernando Boavida. *Service Searching based on P2P Aggregation.* In Proceedings of the International Conference on Information Networking 2010 (ICOIN 2010), Busan - South Korea, 2010.

[Fiorese 11a] Adriano Fiorese, Paulo Simões & Fernando Boavida. *An approach to peer selection in service overlays.* In Proceedings of the 7 *th* International Conference on Network and Service Management (CNSM 2011), pages 1–4. IEEE, October 2011.

[Fiorese 11b] Adriano Fiorese, Paulo Simões & F. Boavida. *On the Dimensioning of an Aggregation Service for P2P Service Overlay Networks.* In Proceedings of the 5^{th} International Conference on Autonomous Infrastructure, Management and Security (AIMS 2011), pages 136–147, Nancy, France, June 2011.

[Fiorese 11c] Adriano Fiorese, Paulo Simões & F. Boavida. *Performance Evaluation of Service Searching using Aggregation in Peer-to-Peer Service Overlay Networks.* In Proceedings of the 12^{th} IFIP/IEEE International Symposium on Integrated Network Management(IM 2011), pages 652–655, Dublin, Ireland, May 2011.

[Fiorese 11d] Adriano Fiorese, Paulo Simões & Fernando Boavida. *Selecćão de pares em redes de sobreposićão usando critérios de localizaćão geográfica.* In Actas da 11^a Conferência sobre Redes de Computadores (CRC 2011), Coimbra - PT, November 2011.

[Fiorese 12] Adriano Fiorese, Paulo Simões & Fernando Boavida. *Peer Selection in P2P Service Overlays Using Geographical Location Criteria.* In Beniamino Murgante, Osvaldo Gervasi, Sanjay Misra, Nadia Nedjah, Ana Rocha, David Taniar & Bernady Apduhan, editeurs, Computational Science and Its Applications ICCSA 2012, volume 7334 of *Lecture Notes* *in Computer Science*, pages 234–248. Springer Berlin / Heidelberg, 2012.

[Foster 08] Ian Foster, Yong Zhao, Ioan Raicu & Shiyong Lu. *Cloud Computing and Grid Computing 360-degree compared.* In Grid Computing Environments Workshop, 2008. GCE '08, pages 1–10, Austin, TX, USA, December 2008.

[Fu 04] Xiang Fu, Tevfik Bultan & Jianwen Su. *Analysis of interacting BPEL web services.* In Proceedings of the 13th international conference on World Wide Web, pages 621–630, New York, NY, USA, 2004. ACM.

[Gamma 94] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. Design patterns: Elements of reusable object-oriented software. Addison-Wesley, 1994.

[Goel 07] Sanjay Goel, Shashishekara S. Talya & Michael Sobolewski. *Service-based P2P overlay network for collaborative problem solving.* Decision Support Systems, vol. 43, no. 2, pages 547–568, March 2007.

[Granville 05] Lizandro Zanbenedetti Granville, Diego Moreira da Rosa, André Panisson, Cristina Melchiors, Maria Janilce Bosquiroli Almeida & Liane Margarida Rockenbach Tarouco. *Managing computer networks using peer-to-peer technologies.* Communications Magazine, IEEE, vol. 43, no. 10, pages 62–68, 2005.

[Gutierrez 08] P.A.A. Gutierrez, I. Miloucheva, D. Wagner, C. Niephaus, A. Flizikowski, N. Van Wambeke, F. Armando, C. Chassot & S.P. Romano. *NETQOS Policy Management Architecture for Flexible QoS Provisioning in Future Internet.* In Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST '08. The Second International Conference on, pages 53–58, sep. 2008.

[Haase 07] Peter Haase, Ronny Siebes & Frank Harmelen. *Expertise-based peer selection in Peer-to-Peer networks.* Knowledge and Information Systems, vol. 15, no. 1, pages 75–107, January 2007.

[Habib 06] A. Habib & J. Chuang. *Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming.* Multimedia, IEEE Transactions on, vol. 8, no. 3, pages 610–621, 2006.

[Hariri 06] Salim Hariri, Bithika Khargharia, Houping Chen, Jingmei Yang, Yeliang Zhang, Manish Parashar & Hua Liu. *The Autonomic Computing Paradigm.* Cluster Computing, vol. 9, no. 1, pages 5–17, 2006.

[Huang 08] Kun Huang, Li'e Wang, Dafang Zhang & Yongwei Liu. *Optimizing the BitTorrent performance using an adaptive peer selection strategy.* Future Generation Computer Systems, vol. 24, pages 621–630, July 2008.

[Janert 09] Philipp K. Janert. Gnuplot in action: Understanding data with graphs. Manning Publications Co., 2009.

[Jannotti 00] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, James W. O'Toole, Jr, M. Frans & Kaashoek James. *Overcast: Reliable Multicasting with an Overlay Network.* pages 197–212, 2000.

[Jesi 07] G. P Jesi, A. Montresor & O. Babaoglu. *Proximity-Aware Superpeer Overlay Topologies.* IEEE Transactions on Network and Service Management, vol. 4, no. 2, pages 74–83, September 2007.

[Jin 06] Hai Jin, Xiaomin Ning & Hanhua Chen. *Efficient search for peer-to-peer information retrieval using semantic small world.* In Proceedings of the 15th international conference on World

Wide Web, WWW '06, pages 1003–1004, New York, NY, USA, 2006. ACM.

[Kaune 09] S. Kaune, K. Pussep, C. Leng, A. Kovacevic, G. Tyson & R. Steinmetz. *Modelling the Internet Delay Space Based on Geographical Locations.* In Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on, pages 301–310, 2009.

[Kim 11] Sung-Su Kim, John Strassner & James Won-Ki Hong. *Semantic overlay network for peer-to-peer hybrid information search and retrieval.* In Integrated Network Management'11, pages 430–437, 2011.

[Kovacevic 07] Aleksandra Kovacevic, Sebastian Kaune, Nicolas Liebau, Ralf Steinmetz & Patrick Mukherjee. *Benchmarking platform for peer-to-peer systems.* it - Information Technology, vol. 49, no. 5, pages 312–319, September 2007.

[Lam 07] Ngok Lam, Zbigniew Dziong & Lorne G. Mason. *Network capacity allocation in service overlay networks.* In Proceedings of the 20th international teletraffic conference on Managing traffic performance in converged networks, ITC20'07, pages 224–235, Berlin, Heidelberg, 2007. Springer-Verlag.

[Lavinal 09] Emmanuel Lavinal, Noëmie Simoni, Meng Song & Bertrand Mathieu. *A next-generation service overlay architecture.* Annals of Telecommunications, vol. 64, no. 3, pages 175–185, April 2009. Publisher: Springer Paris.

[Lee 06] Sanghwan Lee, Zhi-Li Zhang, Sambit Sahu & Debanjan Saha. *On suitability of Euclidean embedding of internet hosts.* In ACM SIGMETRICS Performance Evaluation Review, volume 34, pages 157–168, New York, NY, USA, June 2006. ACM.

[Liang 07] Jin Liang, Xiaohui Gu & K. Nahrstedt. *Self-Configuring Information Management for Large-Scale Service Overlays.* In INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pages 472–480, 2007.

[Lua 05] Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma & Steven Lim. *A survey and comparison of peer-to-peer overlay network schemes.* IEEE Communications Surveys & Tutorials, vol. 7, no. 2, pages 72–93, 2005.

[mac 11] *Macroscopic Topology Measurements.* http://www.caida.org/projects/macroscopic/, 2011.

[Marquezan 07] Clarissa Cassale Marquezan, Carlos Raniery Paula dos Santos, Jéferson Campos Nobre, Maria Janilce Bosquiroli Almeida, Liane Margarida Rockenbach Tarouco & Lisandro Zambenedetti 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.

[Mathieu 07] B. Mathieu, Meng Song, A. Galis, L. Cheng, K. Jean, R. Ocampo, M. Brunner, M. Stiemerling & M. Cassini. *Self-Management of Context-Aware Overlay Ambient Networks.* In 10^{th} IFIP/IEEE International Symposium on Integrated Network Management, 2007. IM '07, pages 749–752. IEEE, May 2007.

[max 10] *MaxMind, Inc.* http://www.maxmind.com/, 2010.

[Maymounkov 02] Petar Maymounkov & David Mazieres. *Kademlia: A peer-to-peer information system based on the xor metric.* In First International Workshop on Peer-to-Peer Systems (IPTPS '02), volume 2429, pages 53–65. Springer Berlin/Heidelberg, 2002.

[Meshkova 08] Elena Meshkova, Janne Riihijärvi, Marina Petrova & Petri Mähönen. *A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks.* Computer Networks, vol. 52, pages 2097–2128, August 2008.

[Michel 06] Sebastian Michel, Matthias Bender, Nikos Ntarmos, Peter Triantafillou, Gerhard Weikum & Christian Zimmer. *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, pages 172–181, Arlington, Virginia, USA, 2006. ACM.

[Michiardi 07] P. Michiardi, P. Marrow, R. Tateson & F. Saffre. *Aggregation Dynamics in Service Overlay Networks.* In First International Conference on Self-Adaptive and Self-Organizing Systems, 2007. SASO '07, pages 129–140. IEEE, July 2007.

[Mischke 03] Jan Mischke & Burkhard Stiller. *Peer-to-peer overlay network management through AGILE.* In IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003, pages 337–350. IEEE, March 2003.

[Munir 09] Saad Ahmed Munir, Xie Dongliang, Chen Canfeng & Jian Ma. *Virtual overlay for service classification & discovery in wireless sensor networks.* In Proceedings of the 11th international conference on Advanced Communication Technology - Volume 2, ICACT'09, pages 991–996, Piscataway, NJ, USA, 2009. IEEE Press.

[Peltz 03] Chris Peltz. *Web services orchestration and choreography.* Computer, vol. 36, no. 10, pages 46–52, October 2003.

[pin 10] *The PingER Project.* http://www-iepm.slac.stanford.edu/pinger/, 2010.

[Pouyllau 10] H. Pouyllau & R. Douville. *End-to-end QoS negotiation in network federations.* In Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, pages 173 –176, april 2010.

[Ratnasamy 01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp & Scott Shenker. *A scalable content-addressable network.* In Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pages 161–172, San Diego, California, United States, 2001. ACM.

[Ripeanu 02] Matei Ripeanu, Adriana Iamnitchi & Ian Foster. *Mapping the Gnutella network.* IEEE Internet Computing, vol. 6, no. 1, pages 50–57, January 2002.

[Rowstron 01] Antony Rowstron & Peter Druschel. *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.* In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pages 350, 329, 2001.

[Rumbaugh 05] J. Rumbaugh, I. Jacobson & G. Booch. The unified modeling language reference manual. Addison Wesley Professional, 2nd edition, 2005.

[Schmidt 04] Cristina Schmidt & Manish Parashar. *A peer-to-peer approach to web service discovery.* World Wide Web, vol. 7, no. 2, pages 211–229, June 2004.

[Semret 00] N. Semret, R.R.-F. Liao, A.T. Campbell & A.A. Lazar. *Pricing, provisioning and peering: dynamic markets for differentiated Internet services and implications for network interconnections.* Selected Areas in Communications, IEEE Journal on, vol. 18, no. 12, pages 2499–2513, December 2000.

[Shen 09] Haiying Shen. *A P2P-based intelligent resource discovery mechanism in Internet-based distributed systems.* Journal of Parallel and Distributed Computing, vol. 69, no. 2, pages 197–209, February 2009.

[Simon 05] Csaba Simon, Róbert Szabó, Péter Kersch, Balázs Kovács, Alex Galis & Lawrence Cheng. *Peer-to-Peer Management in Ambient Networks.* In Proceedings of 14th IST Mobile & Wireless Communications Summit, Dresden, Germany, June 2005.

[Steinmetz 05] Ralf Steinmetz & Klaus Wehrle, editeurs. Peer-to-Peer systems and applications, volume 3485 of *Lecture Notes in Computer Science*. Springer, 2005.

[Stoica 01] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek & Hari 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, pages 149–160, San Diego, California, United States, 2001. ACM.

[Tran 10] Con Tran & Zbigniew Dziong. *Service overlay network capacity adaptation for profit maximization.* IEEE Transactions on Network and Service Management, vol. 7, no. 2, pages 72–82, June 2010.

[Tselentis 09] G. Tselentis, J. Domingue, A. Galis, A. Gavras & D. Hausheer. Towards the future internet: A european research perspective. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.

[Verma 05] Kunal Verma, Kaarthik Sivashanmugam, Amit Sheth, Abhijit Patil, Swapna Oundhakar & John 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, pages 17–39, January 2005.

[Vieira 04] S.L. Vieira & J. Liebeherr. *Topology design for service overlay networks with bandwidth guarantees.* In Quality of Service, 2004. IWQOS 2004. Twelfth IEEE International Workshop on, pages 211 – 220, June 2004.

[Vu 10] Quang Hieu Vu, Mihai Lupu & Beng Chin Ooi. Peer-to-Peer computing: Principles and applications. Springer, 2010.

[Wang 08] Qiang Wang, Rui Li, Lei Chen, Jie Lian & M. Tamer Özsu. *Speed up semantic search in p2p networks.* In Proceedings of the 17th ACM conference on Information and knowledge management, CIKM '08, pages 1341–1342, New York, NY, USA, 2008. ACM.

[Weerawarana 05] Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Tony Storey & Donald F Ferguson. Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging, and more. Prentice Hall, Upper Saddle River, NJ, USA, 1 edition, 2005.

[Williams 10] Thomas Williams, Colin Kelley, Russell Lang, Dave Kotz, John Campbell, Gershon Elber, Alexander Woo & many others. *gnuplot site*, June 2010.

[Wu 05] Chuan Wu & Baochun Li. *Optimal peer selection for minimum-delay peer-to-peer streaming with rateless codes.* In Proceedings of the ACM workshop on Advances in peer-to-peer multimedia streaming, pages 69–78, New York, NY, USA, 2005. ACM.

[Yang 01] Beverly Yang & Hector Garcia-Molina. *Comparing Hybrid Peer-to-Peer Systems.* In Proceedings of the 27th International Conference on Very Large Data Bases, VLDB'01, pages

561–570, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[Youseff 08] L Youseff, M Butrico & D Da Silva. *Toward a unified ontology of cloud computing.* In Grid Computing Environments Workshop, 2008. GCE '08, pages 1–10, 2008.

[Zhao 04] BY Zhao, Ling Huang, J Stribling, SC Rhea, AD Joseph & JD Kubiatowicz. *Tapestry: a resilient global-scale overlay for service deployment.* Selected Areas in Communications, IEEE Journal on, vol. 22, no. 1, pages 53, 41, 2004.

[Zhou 05] S. Zhou, M. Hogan, S. Ardon, M. Portman, T. Hu, K. Wongrujira & A. Seneviratne. *ALASA: When service overlay networks meet peer-to-peer networks.* In Communications, 2005 Asia-Pacific Conference on, pages 1053–1057, Perth, WA, USA, 2005.

[Zhuang 09] Hong Zhuang, Yinlong Xu, Yuchong Hu & Xiaobin Lin. *A Peer Selection Mechanism in P2P Networks Based on the Collaboration of ISPs and P2P Systems.* In Information Science and Engineering, International Conference on, volume 0, pages 1573–1576, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

Appendix A

Service Profile

Service Profiles offer the main structure used by providers to make their services available. They contain administrative, business and technical information about services or service components and are designed as XML documents to guarantee portability and extensibility of the services and service components information, and facilitate data manipulation.

This appendix outlines in detail the elements depicted in the XML representation of Service Profiles described in Section 3.1.1.1. Figure A.1 shows the root element of the XSD. Rectangles with plus signs represent complex elements that contain child elements. For the sake of organization, if a figure contains a complex element that is not expanded, this element will be described in subsequent charts. Following this, there is a detailed description of the entire XSD structure.

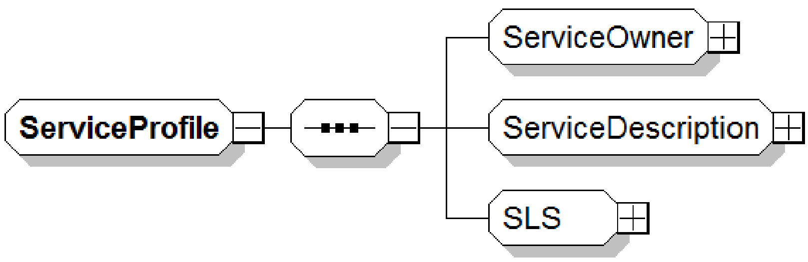


Figure A.1: XSD - Root Service Profile

- *ServiceOwner*: Encompasses the identification and the contact information of the provider of the service element;
- *ServiceDescription*: Contains a general description of the service or service component, such as the name, type and release date;
- *SLS*: Contains technical information about the service element;

A.1 Service Owner

The ServiceOwner element depicted in Figure A.2 shows the identification and contact information of the service providers that make the service or service component available. It contains the following sub-elements:

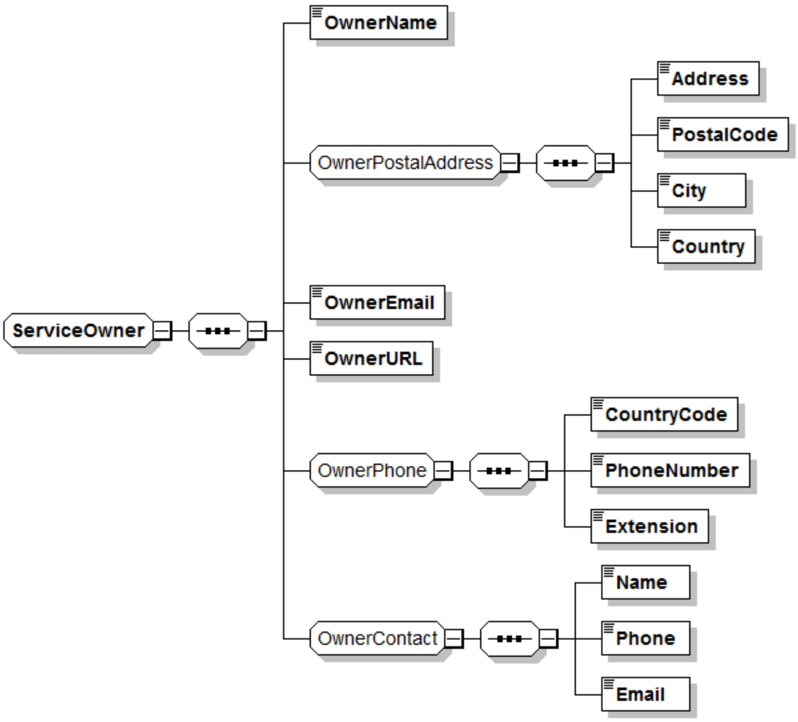


Figure A.2: XSD - Service Owner

- *OwnerName:* The service provider name;

- *OwnerPostalAddress:* The service provider physical address information;
- *OwnerEmail:* The help desk service provider contact e-email;
- *OwnerURL:* The service provider web site;
- *OwnerPhone:* The service provider contact phone number; and
- *OwnerContact:* Name, phone number and email of the person to contact at the service provider's office in order to deal with administrative and technical issues.

A.2 Service Description

Figure A.3 displays general information about the service description element. The ServiceDescription element contains the following sub-elements:

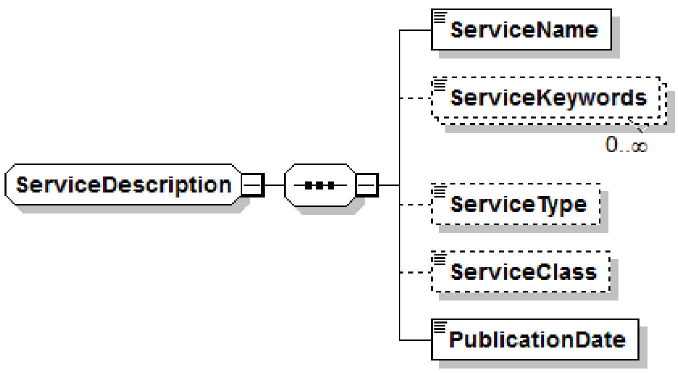


Figure A.3: XSD - Service Description

- *ServiceName:* The name of the service or service component that is executing on the SON peers. This element is used in several operations;
- *Servicekeywords:* Optional element. Along with the ServiceName element, this element defines the keys AgS can use to search services;
- *ServiceType:* The type of the service being offered, e.g. infrastructure, component, etc. Its use is optional;

- *ServiceClass:* Specifies the class (level of quality) of the service; and
- *PublicationDate:* Specifies when the service or service component has been made available at the SON peer.

A.3 SLS

The SLS element (Figure A.4) contains technical information about the service element. It contains the following sub-elements:

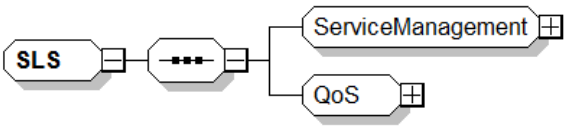


Figure A.4: XSD - Service Level Specification

- *ServiceManagement:* Comprises the interfaces and management functions designed to handle the administrative contacts with the service; and
- *QoS:* Comprises some QoS parameters that services can eventually take advantage of in the context of OMAN.

A.3.1 Service Management

This element is part of the service technical specifications concerning the APIs/interfaces or functions necessary to interact with the service administratively. This interaction is necessary to allow service consumers to connect to the service in order to negotiate and establish SLAs or negotiate the composition process, depending on the case. This element (Figure A.5) contains the following sub-elements:

- *ManagementAPI:* Concerns the functions a service customer can use to communicate with the service to handle administrative functionalities, e.g. celebrate SLAs, retrieve particular information from the service, etc., and

- *CompositionManagement:* OMAN handles services and also *service components*. In order to be used, these service components have to be composed. This optional element (only service components can make it available) contains sub-elements regarding information related to an eventual composition process.

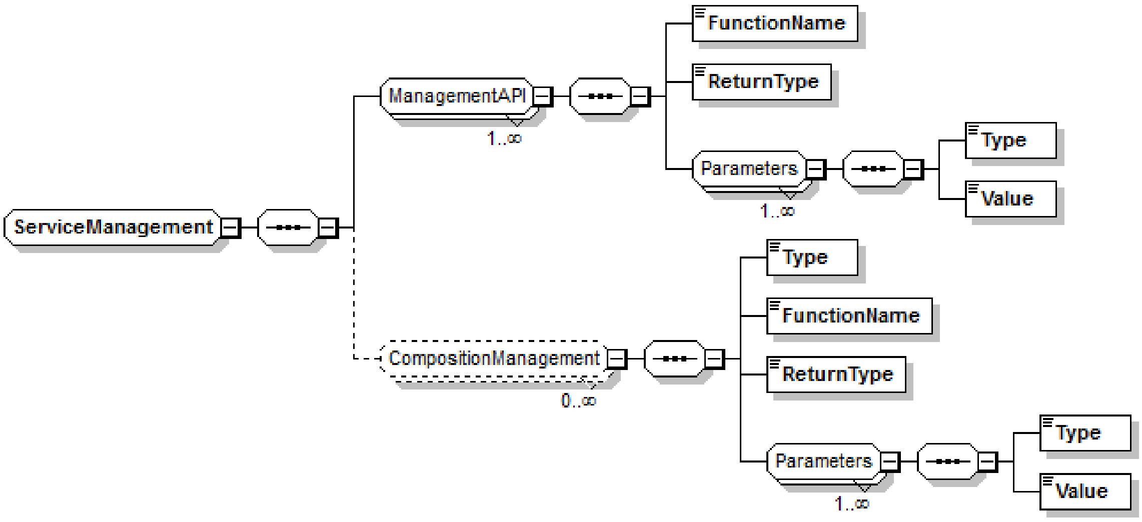


Figure A.5: XSD - Service Management

A.3.1.1 *Management API*

This sub-element, shown in Figure A.5, contains the following sub-elements:

- *FunctionName:* The management function name;

- *ReturnType:* The type, e.g. int, float, etc., returned to the caller (e.g. service consumer) by the management function; and

- *Parameters:* This is an optional element and it either specifies none or several parameters (type and value) that the FunctionName expects as real arguments.

A.3.1.2 *Composition Management*

This sub-element, shown in Figure A.5, contains the same sub-elements as in the *ManagementAPI* sub-element, except for the element Type which can be regarded as the composition type, for instance: orchestration or choreography.

A.3.2 QoS

For the purposes of this thesis, the QoS sub-element is small. Indeed, it appears in the Service Profile as a reminder that with a few adaptations, AgS can perform service searches based on QoS parameters as well, instead of just using the service name or keywords. Thus, as can be seen in Figure A.6, there are a few sub-elements that are a part of the QoS element. These are as follows:

- *Price:* This is an example of a QoS parameter that can be used by the OMAN architecture, especially by the AgS service; and
- *OtherParameters:* This is an optional sub-element that can contain several other QoS parameters, depending on the service or service component.

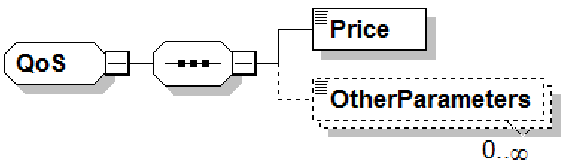


Figure A.6: XSD - QoS

Appendix B

AgS-P Specification

Aggregation Service Protocol (AgS-P) supports four message types for exchanging control and data information between AgS peers and one message type to exchange data information between SON and AgS peers. The first 64 bit-words of all AgS-P messages represent the common header, following by a body consisting of a variable number of objects of static or variable length. The common header and other objects are both encapsulated in a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) messages, and transported by one of these services. The rest of this appendix defines the formats of the common header and each AgS-P message type and object.

B.1 *Common Header*

All the data objects for the AgS-P must contain this common header as the first 64 bits of the object. Figure B.1 introduces the common header format used by all the AgS-P messages.

The fields in the common header are as follows:

- *Msg-Type* is composed of 8 bits, where the first 3 bits mean:

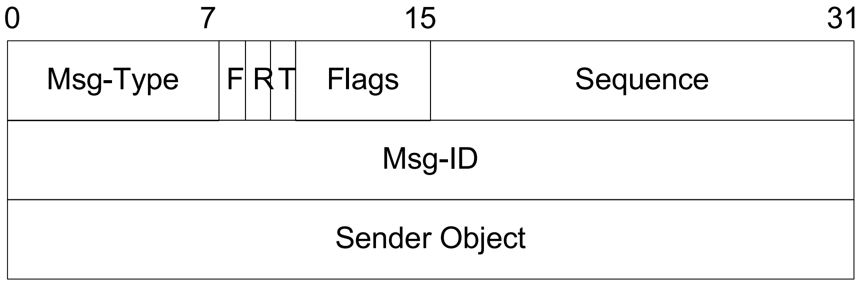


Figure B.1: AgS-P Common Header

000 = JoinMessage;

001 = PublishMessage;

010 = SearchMessage;

011 = SearchReplyMessage;

111 = LeaveMessage;

- *F bit*: When set means the AgS-P message is fragmented.
- *R bit*: When set means the AgS-P message is being retransmitted.
- *T bit*: This bit represents the transport service that will carry the AgS-P message. It can assume two values: 0 = TCP and 1 = UDP.
- *Flags*: Represent flags to specific messages. It is formed of 5 bits and is defined as a part of the individual specification of AgS-P messages.
- *Sequence*: Represents a sequential number associated with each Msg-ID. It is composed of 8 bits and is used when there is AgS-P message fragmentation or retransmission, to prevent flooding and to control the assembly of fragmented messages.
- *Msg-ID:* This field is composed of a 32 bit-word. It represents the message identification.
- *Sender Object*: This field represents the Internet Protocol (IP) address of the message sender. It is a field of 32 bits for IP version 4 or 128 bits for IP version

6. This field is used as a pointer for the successor and predecessor peers of an AgS peer.

The set of appropriated flags depends on the particular message being processed. Any bit that is not defined as a flag for a particular message, must be set to zero on sending and must be ignored on receiving.

B.2 *JoinMessage*

The JoinMessage is very simple. It only uses the AgS-P Common Header to carry the Sender Object as the data field that will be used by the AgS peer receiver to update its internal successor or predecessor fields. The sender AgS peer carries this out by setting the Flags field leftmost bit in accordance with the recipient of the JoinMessage. When it is sending the JoinMessage to its new successor, the bit is set to 0, which means that the Sender Object at the Common Header must be set as the receiver's predecessor. Otherwise, when the JoinMessage is being sent to an AgS peer predecessor, the bit is set to 1, which means the Sender Object must be set as the receiver's successor.

Figure B.2 shows a part of the AgS-P Common Header and highlights the J bit (Flags leftmost bit).

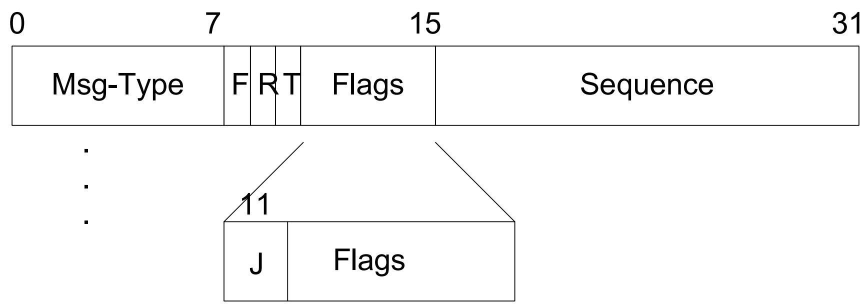


Figure B.2: AgS-P Join Message Flag

B.3 *LeaveMessage*

Like the JoinMessage, the LeaveMessage is very simple. It is based on the same principle of JoinMessage, i.e. the sender gives data to update internal fields of the receiver. Thus, the LeaveMessage uses the same J bit to inform the receiver which internal field to update (successor or predecessor). However, instead of using the Sender Object in the Common Header, the receiver must use an appropriated new field. AgS-P calls this field the Updater Object. The Updater Object is, in fact, the successor or predecessor link object at the leaving peer. Hence,, for instance, when a leaving AgS peer sends a LeaveMessage to its current successor, it must set J bit to 0 and the Updater Object with the IP address of its current predecessor. On the other hand, the second LeaveMessage that a leaving AgS peer sends, it must have J bit set to 1 and the Updater Object with the IP address of its current successor.

Figure B.3 shows the entire LeaveMessage, including the Common Header. Figure B.3 also shows that the Updater Object has the same properties as the Sender Object, i.e. static 32 bit-word representing an IPv4 address or four 32 bit-word representing an IPv6 address.

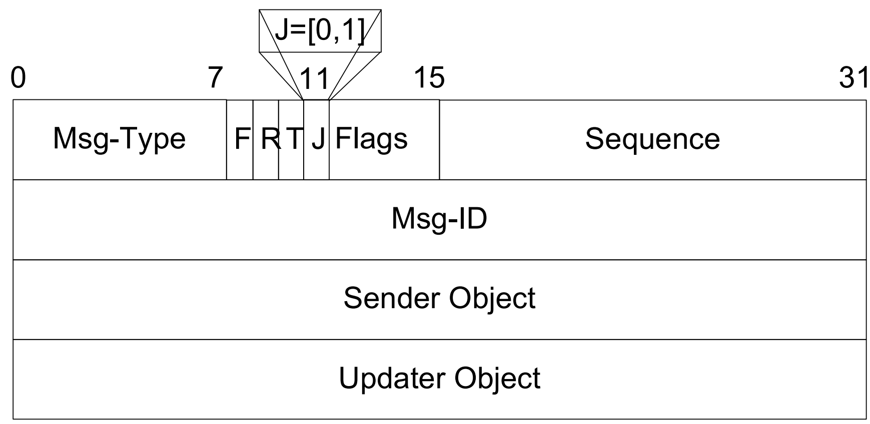


Figure B.3: AgS-P Leave Message

B.4 *PublishMessage*

PublishMessage is responsible for delivering the Service Profile of a service or service component to an AgS peer. Regarding this responsibility the Sender Object at the Common Header is the address of a SON peer. This address will be associated with the sent Service Profile in the local cache of the AgS peer receiver. This data must be present at the PublishMessage.

Figure B.4 shows the format of a PublishMessage, without including the common header, since this message does not have message-specific flags, where:

- *Service Index*: Composed of 8 bits, this is a sequential number for the publishing tries for that Service Profile Object. This field is used to detect redundant publishing messages for a particular Sender Object.
- *Keyword Size*: Composed of 24 bit-word, this represents the size in bytes of a keyword associated with the Service Profile. The keyword is stored at the local cache of the AgS peer receiver as well as the Sender Object and the Service Profile.
- *Service Profile Object Size*: Steady length field composed of 32 bit-word representing the size of the Service Profile Object.
- *Keyword*: Variable length field corresponding to the keyword associated with the Service Profile Object.
- *Service Profile Object*: Variable length field carrying the XML file that represents a Service Profile.

B.5 *SearchMessage*

As mentioned in Section 4.2.3.4, SearchMessage supports two working modes. On the one hand, in the regular mode, the forwarding process stops when the SearchMessage

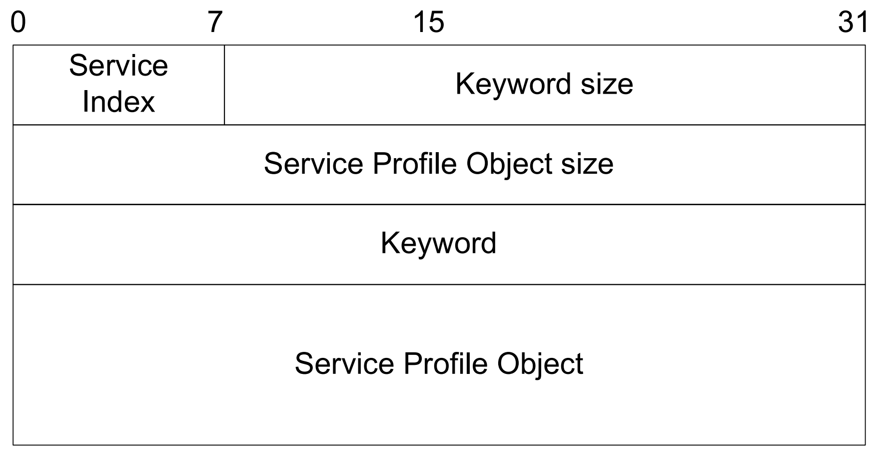


Figure B.4: AgS-P Publish Message

finds first occurrence of the searched service. On the other hand, SearchMessage collects all the occurrences. In order to work in the second mode, SearchMessage must be able to carry the result list, since the message forwarding will only stop when the requesting AgS peer is reached. Furthermore, as the keyword used to match the service or service component must be carried as well as the search result, SearchMessage does not have a fixed length. The SearchMessage fields are as follows:

- *Reserved*: Composed of 8 bits, it is initially set at zero in decimal notation. This field is reserved for future use.
- *Keyword size*: Composed of steady 24 bit-word, represents the size in bytes of a keyword used to search a service or service component.
- *Keyword*: Variable length field, it carries the keyword that has to be compared when searching a particular service or service component.
- *Result List size*: Composed of steady 32 bit-word, it represents the size in bytes of the Result List Object.
- *Result List Object*: Variable length field, it stores the SON peer references that published the searched service.

Figure B.5 shows the SearchMessage, with the AgS-P Common Header. The field Flags in the Common Header supports the M bit to allow the mode of collection of the SearchMessage. This M bit is the 12^{th} bit on the first 32 bit-word on the Common Header, and the 2^{nd} leftmost bit on the Flags field on the Common Header. Thus, when this M bit is set to 0, the SearchMessage will be forwarded until the first service match. Otherwise, it plays the role of collector.

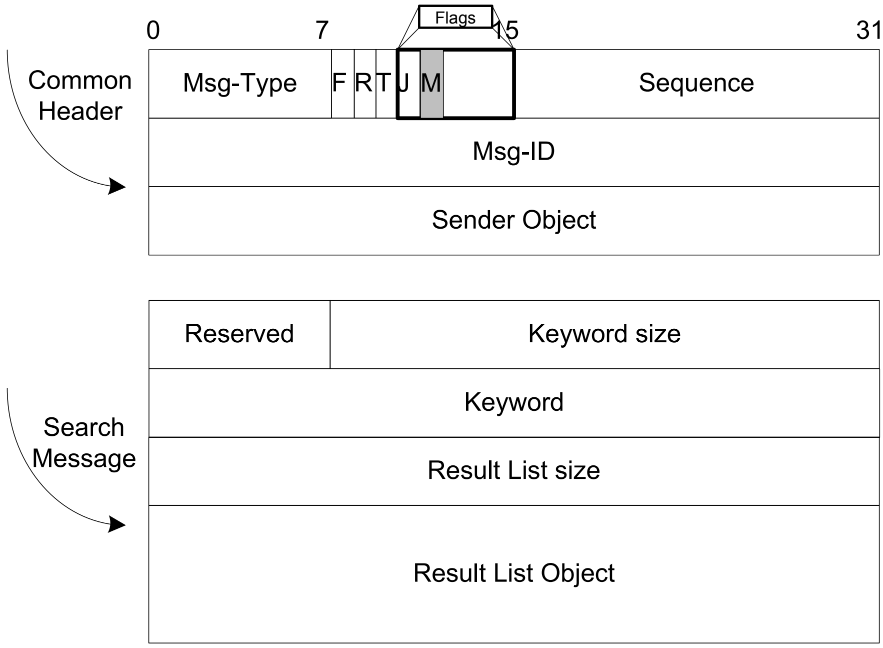


Figure B.5: AgS-P Search Message

B.6 *SearchReplyMessage*

The SearchReplyMessage is the counterpart of the SearchMessage when it is operating in regular mode. In this case, when the SearchMessage finds the first occurrence of the searched service, a SearchReplyMessage is created and filled with information regarding the service profile for the searched service and the address(es) of the SON peer(s) that had published that service profile. Following this, the SearchReplyMessage is sent back to the requesting AgS peer, which is in the Sender Object at the AgS-P Common Header.

The fields on SearchReplyMessage are as follows:

- *Service Profile size*: This field is a steady 16 bit-word representing the size in bytes of the Service Profile object. It is used for the recently found service replication/caching. Thus, if this feature is not used, this field can remain at zero, which means the last field on the SearchReplyMessage will not be used either.
- *Number of SON peers*: The result of a search is a number of references for peers that currently run the searched service. This means that references to this/these peer(s) must be carried in a SearchReplyMessage. This steady 16 bit-word field states how many SON peers will be present in this response message. This is necessary since several SON peers can publish the same service at the same AgS peers. Thus, a search can find several SON peer references for the searched service.
- *SON peer List*: This variable length field encapsulates the address(es) of SON peer(s). However, each individual SON peer object is composed of an IPv4 or IPv6 address.
- *Service Profile Object*: This variable length field is the same as in the PublishMessage. It carries the XML file that represents a Service Profile.

Figure B.6 shows the SearchReplyMessage format without the Common Header.

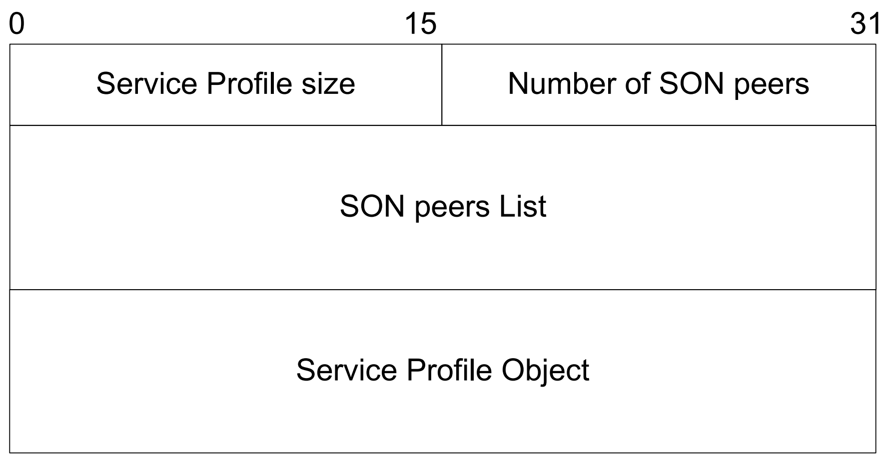


Figure B.6: AgS-P Search Reply Message

Index