



**HAL**  
open science

# OMAN - A Management Architecture for P2P Service Overlay Networks

Adriano Fiorese, Paulo Simões, Fernando Boavida

► **To cite this version:**

Adriano Fiorese, Paulo Simões, Fernando Boavida. OMAN - A Management Architecture for P2P Service Overlay Networks. 4th International Conference on Autonomous Infrastructure, Management and Security (AIMS), Jun 2010, Zurich, Switzerland. pp.14-25, 10.1007/978-3-642-13986-4\_3. hal-01056627

**HAL Id: hal-01056627**

**<https://inria.hal.science/hal-01056627>**

Submitted on 20 Aug 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# OMAN - A Management Architecture for P2P Service Overlay Networks

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

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

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

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

**Abstract.** In a world where networking services are increasingly being provided by service overlay networks, management of these services at overlay level is becoming crucially important. This paper presents an architecture for services management in P2P Service Overlay Networks (SON). The architecture, named OMAN, takes into account the formation of the P2P SON comprising several different service providers belonging to several different network domains. It uses P2P mechanisms to provide service search and service self-improvement through the monitoring of the P2P SON overlay and self-configuration. Preliminary results concerning service aggregation and service searching are presented in this paper, giving an insight into the expected benefits and providing proof of concept as well as pointing the overall potential of the OMAN architecture.

**Keywords:** Services Management, P2P, Service Overlays, P2P SON.

## 1 Introduction

With the advent of service overlay networks, new service provisioning business players are appearing. As in any other business investment, service providers weigh costs and profits, and resort to any competitive advantage in order to maximize revenue. A competitive advantage can be achieved through effective service management, exploiting the features of the overlay network used for service composition. Adequate 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 end users Quality of Experience (QoE).

---

\* This work was partially funded by FCT (scholarship contract SFRH/BD/45683/2008).

In this context, we propose OMAN, a service management architecture for P2P service overlay networks, in a multi-provider environment. Current approaches to the management of the Service Overlay Networks (SON) [1–3] lack strategies for adaptation to the flexible needs of new applications and services intended to Future Internet. OMAN explores adaptation opportunities by using service monitoring information and service aggregation at SON level. According to OMAN, nodes that implement some particular software module and that belong to a P2P overlay, which composes a SON, will inform the application or service that executes over it of key SON performance values, which can be used for SON routing and SON resource management, for instance, allowing for the maximization of application or service quality.

Having in mind the stated goal, this paper is organized as follows. Section 2 discusses related work. Subsequently, in section 3, the proposed architecture will be firstly presented in general and secondly described on a layer-by-layer basis. Section 4 presents some preliminary performance results obtained by simulation and further discusses the key aspects of the Service Aggregation functionality. Section 5 presents future work and Section 6 summarizes the conclusions.

## 2 Related Work

The concept of services as applications that can be accessed over the Internet through a software interface is used in this paper as in [4]. The concept of Service Overlay Networks (SON) is not new as well. A SON is an overlay network designed to provide services [1, 3]. The main SON issues are the establishment of the overlay itself and its management.

The bandwidth provisioning over a SON as well as the interactions with the domains involved, and the necessary agreements to do that are studied in [3]. The main contribution of that paper is the study of the bandwidth provisioning problem in a SON using the pipe bandwidth model for static and dynamic provisioning.

Taking advantage of the P2P technology, some of the initial issues related with the establishment of a regular SON can be solved. The scalability through multiple domains as well as the resilience of the P2P overlay networks allow to connect several different service providers over a P2P overlay network to deliver and provision their services. This is the subject of the work presented in [5]. However, this work does not the issue of how to improve or handle enhancements on the services using the properties and information present in the P2P overlay that composes the SON.

Service Overlay Networks composed by Web Services are discussed in [4]. The federation of service providers is the core behind the idea of that paper. The web services are used to compose a SON using SLA as a management tool, and to monitor the intermediaries in order to pursue the accomplishment of the SLA.

The SON composed by peers in a P2P mode is created in an on-demand basis. As in [3], the utilization of SLA is handled by a module of our architecture.

However, this handling is considered in our architecture as a pre-production step since the creation of the P2P SON is intended to be made in a cooperation basis among the service provider competitors.

Some work on QoS-aware SON is presented in [6, 7]. However, these address only service level path establishment and management as well as service components discovery. In [6] this latter issue is pursued by an enhanced CAN P2P overlay network [8, 9]. Our architecture goes beyond this since it proposes to improve the service behaviour while the service is executing.

In [10] a Resilient Overlay Network (RON) is proposed. RONs are designed as overlay networks whose purpose is to recover from path outages in a few milliseconds. To accomplish this, RON nodes monitor the quality of the Internet paths and use this information to decide whether to route the packets directly over the Internet or by way of other RON nodes. This idea has some advantages in routing path resilience; however, services should be optimized according to several other application's criteria beyond path routing as, for instance, the load on the intermediary peers.

### 3 OMAN - Overlay Service Management Architecture

In order to face the service management problem in a P2P SON, it is necessary to handle aspects ranging from the composition of the SON until the interaction aspects between the services and the SON, including how to **take advantage of the information at the P2P overlay level to leverage the services and applications**. OMAN intends to target the latter aspect. Figure 1 shows the proposed OMAN architecture.

The circle end line seen in 1 means an interface for accessing the functionality offered by the module where the line starts, whereas the arrow end line means the module where the line starts is offering some particular information.

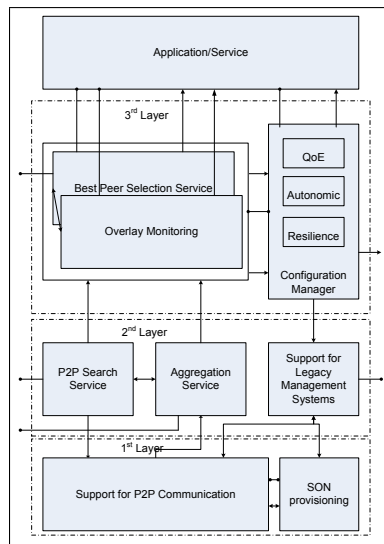
At layer 1 the P2P communication will support the whole architecture. This module should be common to every node participating in the P2P SON. This module will be responsible to manage all the aspects related with the maintenance of the P2P overlay, including the actions of joining and leaving of nodes. This layer is also responsible for handling the aspects of the P2P SON provisioning. It includes the Service Level Agreements (SLA) and administrative management of the service providers in the SON's group.

At layer 2 some basic framework services are provided. This is the case of the external or legacy management systems. Also, service providers can utilize the search functionalities at this level to find overlay management services, particular specific services or infrastructural services, and component services to be used in assembling new services. A particular service designed to cope with the latter kind of searching is the Aggregation Service (AgS) [11]. The results from the AgS can be used by external composition service platforms.

At level 3, specialized services in terms of overlay management and services improvement will take place. The Overlay Monitoring (OM) module collects information about the state of the overlay regarding its resources and execution

conditions. The Best Peers Selection Service (BPSS) is a service that informs a set of best peers to position a service according a particular application metric. In this sense this service helps applications and services on the use of the monitored overlay. The OM and BPSS support the Configuration Manager (CM) service. The CM has sub-modules: Resilience, QoE and Autonomic that cope with the dynamic aspects of the overlay 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 experience information from the users, concerning the services provided by the whole system. The Autonomic sub-module is intended to provide some self-\* capabilities [12] to OMAN architecture, in particular self-configuration capabilities, based on an initial management policy received from the application or service controller (provider) and on the information collected by the OM module.

In the next subsections each module of the architecture will be explained in detail.



**Fig. 1.** The OMAN Architecture

### 3.1 First Layer

This layer comprises the P2P overlay network itself. To accomplish that, every peer involved in the framework that implements this architecture must have the Support for P2P Communication and the support for SON provisioning.

**Support for P2P Communication** This module comprises the algorithms and strategies necessary to implement a P2P overlay network. The choice of which particular P2P overlay will be used depends on the communication provider of the SON. However, preliminary studies will be conducted using CHORD [13] for the validation. Nonetheless, if a particular application that is not aware of the CHORD mechanisms or that wants to implement specific strategies to routing its information then this application or service should implement its own particular routing scheme, including how to choose and how to manage the neighbour's peers.

**SON provisioning** This module negotiates and enforces the SLA among the service providers that compose the P2P SON. Among other responsibilities this module should guarantee the financial sustainability of the P2P SON with billing policies based on the content shared or bandwidth used, or other parameters the involved choose.

### 3.2 Second Layer

**P2P Search Service (P2PSS)** This module is in fact the first interface to access the generated P2P overlay network. For instance, pure CHORD applications use this module as a service to find out the information they need. For the CHORD overlay, not only the peers' ID but either the application information (e.g. shared files) must be keyed into the flat key space identifier of 128 bits. The ID or key generated is used to find out in which peer in the overlay that information will be stored and by the same principle that information is recovered: just informing its key. This module offers an interface to external applications to use it.

**Aggregation Service (AgS)** The AgS is a higher level search service. It comprises the aggregation of particular services' information in order to improve the search. It works like a second tier for the P2P Search Service, increasing the response time for a specific search. In our validation framework a particular P2P SON use the AgS to search service components to be assembled in new services. The adopted strategy *per si* diminish the time of searching, and using also the replication of the search results the performance gets better, as shown in Section 4.2.

**Support for Legacy Management Systems** This module offers an interface to standard management legacy systems. Third party software operators can use this module to offer a network management service for instance. This module can be adapted and it can be increased depending on the necessity of the P2P SON operators. Basically, this module is responsible for translating the requests from the service into the necessary commands to legacy management systems running on the node (peer) that compose the P2P SON. Also, this module is responsible for exposing the answer received from the legacy system.

### 3.3 Third Layer

**Overlay Monitoring (OM)** The Overlay Monitoring is responsible for offering data related with the P2P SON to the applications and services. This module collects this information through monitoring peers (using pooling or the 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 according its necessity.

The approach used in the Overlay Monitoring module is that follows. Each node belonging to the P2P SON should report the monitored information in a time period interval (or when the information is ready to be reported in the case of publish/subscribe paradigm) to an aggregation node (super-node). These super-nodes, which will be elected according how long they are alive in the SON, exchange the information allowing the application/configuration manager to have easy access to it and how to use it.

Thus, the following information should be monitored.

**Processing Load:** This information depicts how idle or busy is the node's processor. It is closely related with the node's load, although that information also takes into account memory and disc usage per process. Hence, the processing power information is appropriate to applications and services that share processing capabilities. This information is kept as an average value between measurement intervals. This strategy avoids the use of a node that is intensively processing data or that makes intensive use of calculations even taking 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 sharing content applications, especially file sharing ones. This information is also important, to instantiate or to clone the same application or service in other 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 not. 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 a node can communicate with other. So this information is especially valued for routing, content delivery and almost every networked operation an application intends to perform.

**Node Load:** The node load depicts how busy is the node as a whole in the P2P SON. This metric takes into account not only 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 to applications that need some kind of

processing information about other peers in a session. For example, a service that might encrypt or execute a codec and decode some video traffic in the middle of a transmission could use the node load information to choose which nodes can participate on the forwarding route.

**Best Peer Selection Service (BPSS)** The Best Peer Selection Service (BPSS) is a module in our architecture intended to face the peer's location problem. Sometimes the nearest peer can be, or not, the best peer to exchange monitoring or metadata information. Hence, the choice of the best peer is dependent on a particular objective an application has. In compliance with the overlay monitoring, our BPSS module offer the topological metric to the applications. The topological metric is the information about the distances among the peers belonging to the SON. These distances are measured according the link latency among them or using other techniques (landscape mark servers, for instance), and they are dependent on the bandwidth and load existent between nodes. All in all, to use this module the application or service should specify the metrics and the number of best nodes it needs and request them. Depending on the specified metrics, the BPSS will use the Overlay Monitoring (OM) module (so both module should be implemented all together), to answer with a bunch of peers in the P2P SON that correspond that metrics. When the only metric desired is the distance between the node where the service is running and other particular nodes then just the BPSS will be used.

**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 receiving the configurations to be applied to the overlay; create groups of peers to execute particular management operations; control the authorizations for applying configurations and with the help of the BPSS module it is responsible for applying the configuration on the appropriated service, device or peer. It also can offers information about the result of the configuration or also delegates the operation of configuring to another peer. The CM autonomously will use the information collected by the OM module to adjust the storing, sharing, and other capabilities.

To control the underlay, CM will use the Support for Legacy Management Systems. The decisions the CM should make to control the underlay are taken depending on the overall policy of the P2P SON (based on SLAs agreed between the operators or providers in the application or service level), and autonomously by the metrics monitored by the Overlay Monitoring (OM) module. The Autonomous sub-module will take care of this aspect.

The information monitored by the OM module also influences the Resilience sub-module. This last module is responsible to keep the P2P SON working when disrupting abnormal conditions occurs. An example of its working appears when a node in the CHORD overlay does not respond to the requirements and its application and service must be cloned and started again in another node. The



task of cloning and starting the “new” application is part of the responsibilities of this module. The choice of the best peer to start the “new” application can be done using the BPSS module. So the Resilience sub-module is dependent on the implementation of the BPSS module.

With the regular necessity of enhancing applications and services, the user’s general opinion about them is even more required. The Quality of Experience (QoE) is the concept 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 analysing the initial management policy and automatically adapt the P2P SON to reach the statements is executed by the sub-module we call Autonomic. The self-configuration is one of a series of self-\* [12] characteristics intended to comprise the self-management concept. At a first glance the P2P SON adaptation based on an initial policy with the integration of the information collected by the Overlay Monitoring module can also support the Resilience sub-module that executes some self-healing functions.

To accomplish 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 need to implement the other modules of the third layer of our architecture.

## 4 Preliminary Results

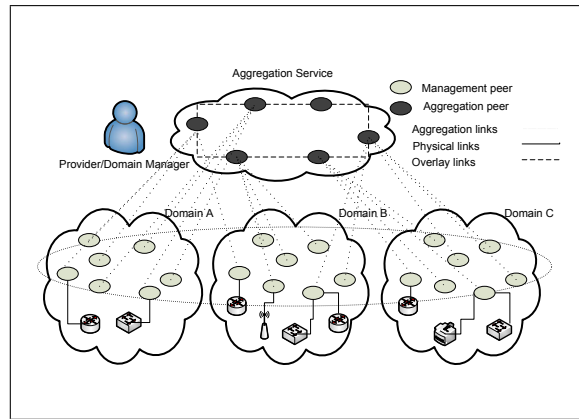
This section presents some preliminary results obtained by simulation, concerning the AgS, which is in the layer 2 of OMAN. The AgS was chosen as proof of concept to illustrate part of the OMAN contribution. For that, it was chosen to be presented. Then, firstly, the AgS will be explained in more detail and finally assessment results will be shown and discussed.

### 4.1 Aggregation Service (AgS)

The proposed AgS is a P2P tier over the Support for P2P Communication. As a P2P tier, it consists of super-peers elected among the peers that belong to the service providers that constitute the P2P SON. These super-peers are also called aggregation peers and their purpose is to aggregate the offerings of services and service components in another overlay tier to improve search.

Unlike the P2P Search Service (see Fig. 1, 2<sup>nd</sup> layer), whose functionality is provided by the P2P overlay network itself to execute generic searches, the AgS is designed to improve the search process of aggregated services and component services that will be offered to end users or to third party service providers to make a service composition. For this to work, the peers that execute these services and component services, which are called management peers, should publish the interfaces of these services and component services in the aggregation peers. Once the interfaces are published they can be searched. Figure 2 illustrates the AgS concept.

Thus, for instance, the AgS in the P2P SON can be utilized in a scenario where a provider of a composed service wants to search service components to be assembled into a new service. This provider uses the AgS to search the management peers where the required service components are available. Taking Fig. 2 as reference, 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 management peer reference for this service can be recovered. Further access to the service interfaces (e.g. contracting, monitoring, and life cycle management) is provided by the management peer that represents that service component. Interaction between the service contractor and the management peer is conducted outside the AgS.

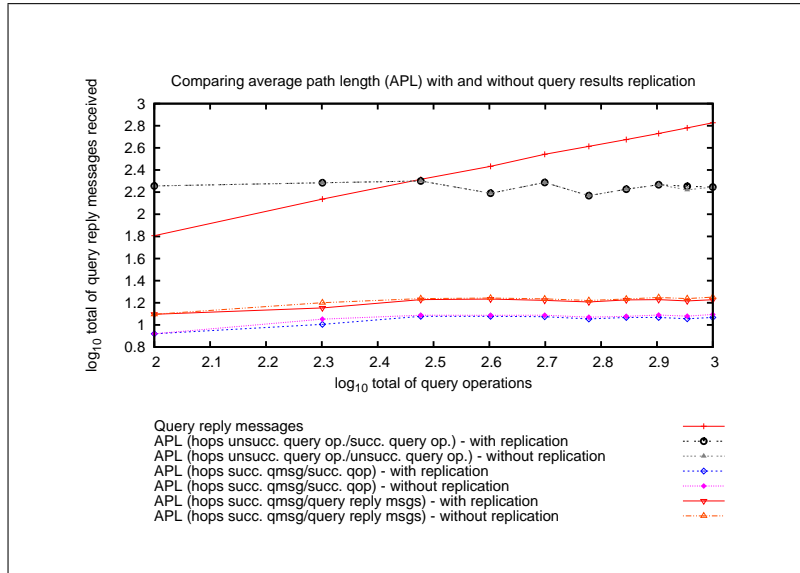


**Fig. 2.** The Aggregation Service (AgS)

## 4.2 Aggregation Service Evaluation

The AgS was evaluated through simulation. The simulated scenario was a virtual ring arrangement with 1.000 aggregation peers and 10.000 management peers. Each management peer could randomly publish, using a uniform distribution, up to 7 different service references in a randomly chosen aggregation peer. The simulations used the hop-count number of the Search Message (sent between aggregation peers) as a metric to calculate the average path length for the searches. The searching operations ranged from 100 to 1000 ones in order to observe the scalability and the average path length behaviour in each of the scenarios belonging to the same simulated environment. Two scenarios were assessed: 1) a scenario where the search result was cached (replicated) on the aggregation peer who started the search; and 2) a scenario without the replication of the search

results. Preliminary results show high scalability as well as the resilience of the proposed AgS, through the replication of the search results. Figure 3 shows the most significant results.



**Fig. 3.** AgS average path length with and without replication of search results

In Figure 3 is possible to see that there is a difference between the lines representing both scenarios, and that the average path length on the scenario whose search results are replicate is lower. The confidence interval for mean is of 95%. Thus, for the scenario without replication the average number of hops (peers) to find information is 14,1805 and the confidence interval is 1,004258. For the scenario with replication the mean is 13,634 with confidence interval of 0,950478.

The results show that replication of search results improves the search performance and also enhances the resilience, since the aggregation peer that originally received the service reference publication message can go down and that service reference can still be reached through a previous replicated search result. It is worth saying that over time this strategy can lead to a situation where a good number of search operations are successfully performed in the peer’s local cache, avoiding any communication overhead on the AgS P2P overlay. As expected, the average path length in unsuccessful query operations is greater than in successful operations and it is not altered by replication or not, as it can be seen in the lines (hops unsucc. query op.) in Figure 3.

Another finding is that the average path length depends on the number of active aggregation peers. Although the number of simulated aggregation peers

was 1,000, the AgS P2P overlay consisted, on average, of 177 active aggregation peers with standard deviation of 16. This is the number of hops undertaken by unsuccessful query messages that travelled through the entire overlay and did not find the desired information. This low number of active aggregation peers can explain the low average path length that was shown in the results.

## 5 Future Work

In order to answer questions regarding the efficiency of the OMAN architecture some work is needed. First of all, the integration of AgS in a full implementation of the architecture in a simulated scenario or test-bed can provide valuable information about overall usability, about the capacity to handle a large number of services and about the interaction with and dependencies on the supporting P2P overlay.

Also, simulations on the Best Peer Selection Service (BPSS) - a key component to services enhancement - should provide information about how to organize, in terms of topology, a real P2P SON. Last but not least, this kind of work also can contribute to assess what we claim as the necessary cooperation among service providers in order to keep the P2P SON communications linking costs under control.

The Overlay Monitoring (OM) implementation and evaluation is also of great importance. Together with the BPSS module, the OM module can play a key role in service performance improvement, leveraging the end user quality of experience and the efficiency of service provision.

The integration of the BPSS, OM and the Configuration Manager in a test-bed whose purpose will be to assess the behaviour of services running on a P2P SON is another important topic for future work.

## 6 Conclusion

This paper presented OMAN, an architecture for service management in P2P overlay networks. OMAN addresses the problem of managing and providing service composition in large-scale environments involving multiple administrative domains and several service providers organized in a P2P SON.

OMAN is based on 3 layers. The first layer comprises the support for P2P communication, where a P2P network involving service providers will take place, as well as an administrative module whose responsibility is to compose the service overlay network and to establish the necessary inter-provider agreements. The second layer comprises a searching service, whose main purpose is to offer basic and advanced searching features to the applications and services as well as to service providers; additionally, this layer provides support for legacy management services. The third layer provides value-added monitoring and adaptation services, based on information collected at the P2P SON level. This modular construction allows specific delivery strategies over the P2P SON for the service

providers, allowing different peers to play different roles according the service providers policies.

This paper also presented and discussed some preliminary results about the AgS module of the OMAN architecture. In this context, two scenarios were simulated: first, a scenario using the caching (replication) of the search results at the aggregation peer where the query was initiated; second, a scenario without replication of the search results. The obtained results confirm the good scalability, resilience and performance of the AgS paradigm, and point to the potential of the overall OMAN architecture.

## References

1. Fan, J., Ammar, M.H.: Dynamic topology configuration in service overlay networks: A study of reconfiguration policies. In: INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings. (2006) 1–12
2. Liang, J., Gu, X., Nahrstedt, K.: Self-Configuring information management for Large-Scale service overlays. In: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. (2007) 472–480
3. Duan, Z., Zhang, Z., Hou, Y.T.: Service overlay networks: SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Trans. Netw.* **11**(6) (2003) 870–883
4. Machiraju, V., Sahai, A., Moorsel, A.V.: Web services management network - an overlay network for federated service management. In: IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003. (2003) 351–364
5. Zhou, S., Hogan, M., Ardon, S., Portman, M., Hu, T., Wongrujira, K., Seneviratne, A.: ALASA: when service overlay networks meet Peer-to-Peer networks. In: Communications, 2005 Asia-Pacific Conference on. (2005) 1053–1057
6. Lavinal, E., Simoni, N., Song, M., Mathieu, B.: A next-generation service overlay architecture. *Annals of Telecommunications* **64**(3) (April 2009) 175–185
7. Adam, C., Stadler, R., Tang, C., Steinder, M., Spreitzer, M.: A service middleware that scales in system size and applications. In: Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on. (2007) 70–79
8. Ratnasamy, S., Francis, P., Shenker, S., Karp, R., Handley, M.: A scalable Content-Addressable network. In Proceedings of ACM SIGCOMM (2001) 161–172
9. Crowcroft, J., Pias, M., Sharma, R., Lim, S., Lua, K.: A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE* (2005) 72–93
10. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient overlay networks. In: Proceedings of the eighteenth ACM symposium on Operating systems principles, Banff, Alberta, Canada, ACM (2001) 131–145
11. Fiorese, A., oes, P.S., Boavida, F.: Service searching based on P2P aggregation. In: Proceedings of ICOIN 2010, Busan - South Korea (2010)
12. Hariri, S., Khargharia, B., Chen, H., Yang, J., Zhang, Y., Parashar, M., Liu, H.: The autonomic computing paradigm. *Cluster Computing* **9**(1) (2006) 5–17
13. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, San Diego, California, United States, ACM (2001) 149–160