International Journal of Recent Trends in Engineering & Research (IJRTER)
Conference on Electronics, Information and Communication Systems (CELICS'18)
Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

Chord4S: An Efficient Service Discovery Approach in P2P Networks

V.Sunil Anandh
Ap, Department of CSE
Annai Vailankanni College of Engineering

Abstract— Service-Oriented Computing (SOC) is emerging as a paradigm for developing distributed applications. Centralized service discovery approach that uses UDDI registries that lead to bottleneck and vulnerability of failures problem. Distributed service discovery approach named chord that uses Dynamic Hash Table for service discovery that leads to less data availability. To address these problems an efficient service discovery approach called chord4S is introduced which provides efficient service discovery in peer to peer network environments .Data availability is improved by using this approach. In chord4S data availability is improved by distributing published descriptions of functionally equivalent services to different successor nodes.Chord4S provides better reliability and flexibility. By using this approach we can retrieve a list of matched service descriptions by forwarding queries. After retrieving a service we can open the location of the file and can access the specific folder with security. Also the accuracy of this process is also improved.

Keywords— Chord4S, Query Forwarding, Search Process, Identifier circle*,* Service identifier

1. INTRODUCTION:

Peer-to-peer (P2P) is an alternative network model to that provided by traditional client-server architecture. P2P networks use a decentralized model in which each machine, referred to as a peer, functions as a client with its own layer of server functionality. A peer plays the role of a client and a server at the same time.

The peer can initiate requests to other peers, and at the same time respond to incoming requests from other peers on the network. It differs from the traditional client-server model where a client can only send requests to a server and then wait for the server's response.

Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance without requiring the intermediation or support of a global centralized server or authority.

Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

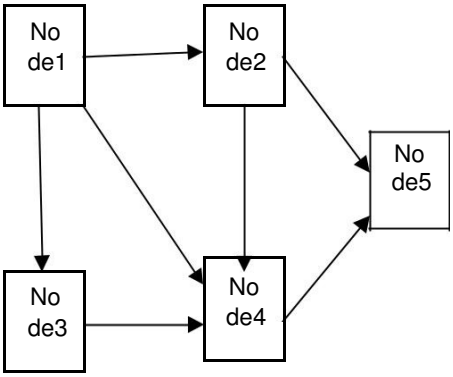


Figure 1: Peer to Peer Network

Service-Oriented Computing (SOC) is emerging as a paradigm for developing applications. Traditional service discovery approach that uses UDDI registries [1]. One or more UDDI nodes may be combined to form a UDDI Registry. The nodes in a UDDI registry collectively manage a particular set of UDDI data. This data is distinguished by the visible behavior associated with the entities contained in it. A UDDI Registry has these defining characteristics.

1. A registry is comprised of one or more UDDI nodes.
2. The nodes of a registry collectively manage a well-defined set of UDDI data. Typically, this is supported by the use of UDDI replication between the nodes in the registry which reside on different systems.
3. A registry must make a policy decision for each policy decision point. It may choose to delegate policy decisions to nodes.

A major issue with peer to peer applications is that they are not able to efficiently locate the nodes containing a particular data item. To solve this chord protocol has been developed which is a distributed lookup protocol. Its primary function is to map a given key onto a node. Chord is a scalable protocol, and it provides efficient service discovery .it retrieves the service from centralized registries and send it to the service consumer.

Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

(A) **CENTRALIZED SERVICE DISCOVERY:**

A centralized topology is probably the most efficient in terms of configuration, because clients only have to maintain the location of one registry [1]. Also, since there is only one centralized location with a complete view of the service network state, query response control is not a problem.

To maintain an up-to-date view of the service network in a centralized registry, removal of old advertisements should be done, since we have to deal with dynamic conditions where services may disappear abruptly. Should the number of service advertisements grow very large, storage may be a constraint as well, since one node must host all advertisements. Further, by delegating service selection to the central registry, query evaluation may only have to be carried out once. The opportunity to allow service selection support in registries is important to relieve constrained clients.

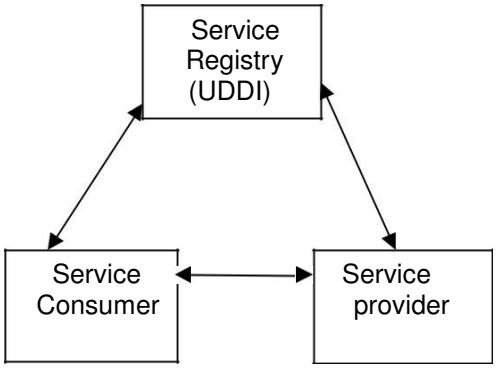


Figure 2: Centralized Service Discovery

However, a completely centralized solution has problems related to robustness, since we now have a single point of failure.

(B) **DECENTRALIZED SERVICE DISCOVERY:**

Decentralized service discovery is considered as a promising approach to addressing the problems caused by centralized infrastructures. In particular, some preliminary research has been conducted to utilize P2P computing for service discovery.

Distributed approaches are mainly based on P2P systems. The semantic of a service is described using WSDL- S. The WSDL-S most important information is hashed to a Chord network and the others to a Skip Graph that accelerates and facilitates data querying while preserving the information structure.

Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

Decentralized service discovery that uses distributed hash table (DHT) is a class of a decentralized distributed system that provides a lookup service similar to a hash table pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key. Responsibility for maintaining the mapping from keys to values is distributed among the nodes, in such a way that a change in the set of participants causes a minimal amount of disruption.

2. RELATED WORK:

In this section, we will see the some of the related works to the intrusion detection system using different approaches:
The centralized client/server model has been adopted for service discovery since SOC emerged. UDDI has been recognized as the most popular discovery mechanism for web services. At present, several software vendors have included UDDI support as a key feature of their software products to provide comprehensive solution for application and service integration challenges.

Romp thong and Senivongse propose a federation of UDDI registries to enlarge the search space for service queries. Although a UDDI Federation Agent is added as an extension to a standard UDDI registry to forward queries to other federating nodes, the authors did not provide any experimental evaluation.

Wu et al.describe an interoperable model of distributed UDDI which divides UDDI servers into three types: root server, super domain server, and normal server. The authors adopt the philosophy of Domain Name System (DNS). Super domain servers, managed by a root server, are used to maintain normal servers. Since the model imitates DNS, it is still exposed to the same threats that DNS faces, e.g., Distributed Denial of Service (DDoS) attack.

Web Services Dynamic Discovery (WS-Discovery) a multicast discovery protocol to locate services on a local network is developed by BEP Systems, Canon, Intel, Microsoft and Web Methods. In WS-Discovery, a client sends a request to the corresponding multicast group to locate a target service. A proxy-specific protocol is also defined and can be switched on if a discovery proxy is available on the network. WS- Discovery is becoming popular and is already being used by some software vendors, such as the People near Me contact location system in Microsoft's Windows Vista operating system.

M-Chord utilizes idistance to transform the metric search problem into the interval search problem in one dimension. It provides Chord with the ability to perform metric-based similarity search. We observe that none of the above has addressed the issue of data availability in open and volatile SOC environments. Node failures would lead to severe data loss when the above approaches are adopted to facilitate service discovery because descriptions of functionally

DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

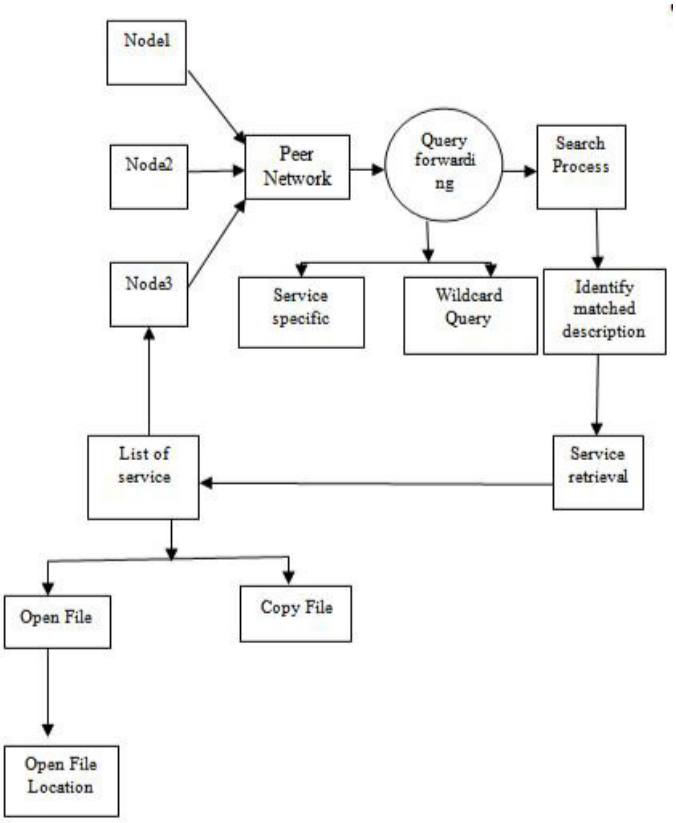
equivalent services would be stored at the same successor nodes.

3. PROPOSED SYSTEM:

Chord4S, a Chord-based decentralized service discovery approach that supports service description distribution and discovery in a P2P manner.Chord4S takes advantages of the basic principles of Chord for nodes organization, data distribution and query routing.

Aim:

The main aim of designing Chord4S is to largely improve the availability of service descriptions in volatile environments by distributing descriptions of functionally equivalent services to different successor nodes. In case one node fails, a service consumer is still able to find functionally equivalent services that are stored at other successor nodes. Another two features of Chord4S are to support service discovery with wildcard(s) and QoS awareness. Furthermore, Chord4S extends Chord's original routing protocol to support discovery of multiple functionally-equivalent services at different successor nodes with one query, which is necessary for negotiation of a service level agreement (SLA) and selection of optimal service providers. By using this Chord4s approach we can retrieve an efficient service description. Moreover we can open and copy the file in specific location with authentication. Also we can find the path of service which we retrieved by service consumer



Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

Figure 3: Proposed System Architecture

A number of nodes are connected to form a .peer to peer networks. With this network a user search a file by forwarding a query and they can login into the system and can submit the query from node. For searching there are two types of query can submit.

(a)Service Specific Query

Service specific query contains the complete description of service. It means that a user want to search document file means specify the type of service. To initiate a service-specific query, the service consumer needs to fill in all the layers that compose the query with explicit service information. Then each of those layers will be hashed and the results will be connected to generate the function bits of the target service identifier.

Since the objective of using service-specific queries is usually to look up a group of functionally equivalent services provided by different service providers, the provider bits of the query will be stuffed with 0s instead of being explicitly specified.

(b)Query with wildcard

Sometimes service consumers need to search for categories of services. For example, an amplifier service that amplifies the audio of an RM movie file can be composed using three component services which correspond to three specific steps: audio extraction, audio amplification, and video/audio combination.

Process Description:

After submit the query the query will forward to identifier circle and search process will take place. In the identifier circle the functionally equivalent services are store in different successor nodes. The query searches what are all the matched service descriptions available in the nodes. After finding a matched service in a node, it will able to route the query to next successor node in the network.

The search process completes until the query reaches to all nodes. After finding all the matched service description it will retrieve the query to the service consumer. A list of matched service description is displayed in the form. After that user will copy the file and place it in specified location. the user can also open the file and can save it.

The service discovery process in Chord4S is different from traditional approaches based on the original Chord. The main difference is that successor nodes may need to forward the query message based on their own routing information.

DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

The service identifiers are the identifications of the services as the basis for routing query messages. The QoS specification specifies the quality of the service that the service provider can offer. The syntax specification describes the syntax of the service, e.g. the names and data types of the input and output parameters. This paper only focuses on the former two concepts, i.e. service identifier and QoS specification.

4. MODULES:

- Query Forwarding
- Search Process
- Service Retrieval
- Open file location
- Security

4.1 Query Forwarding

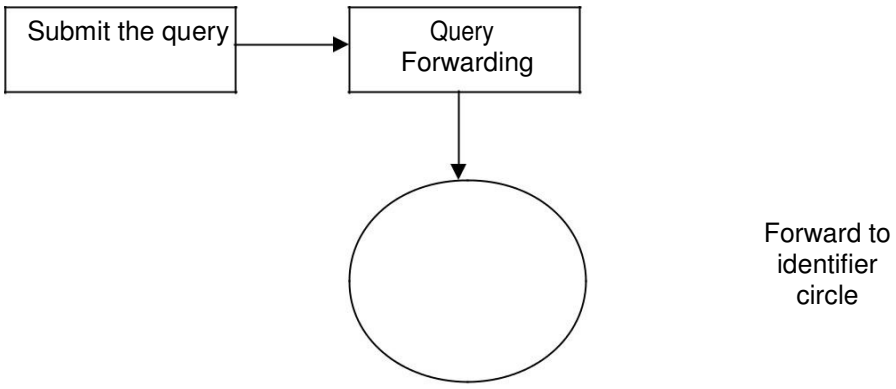
Chord4S supports two types of query: service-specific queries and queries with wildcard(s). When a user wants to search a file means they can forward a query to the network. The query will forward to the identifier circle.

Service specific query

In Chord4S, for a service consumer to find multiple functionally equivalent services with one query, the query must be routed across the corresponding virtual segment of the
identifier circle until sufficient services required by the service consumer have been found. In a system that allows four-layered function bits descriptions, "Multimedia.Video.Encoder.AVI2RM"
is a typical example of service-specific query.

Query with wildcard

The process of forwarding queries with wildcard(s) is similar to that of forwarding service- specific queries. Instead of all the function bits, only the bits generated from explicit service information will be taken into consideration. The service consumer needs to find the component services from categories of services.



DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

Figure 4: Query forwarding

4.2. Search Process

After submitting the query, it will forward to the identifier circle and search what are all the matched service descriptions are available in the identifier circle. In the idenfier circle the service identifier of each description is available for each service with its description. The search process searches the functionally equivalent services in the identifier circle. After searching a service in a node and the query will forward to next successor node and so on. This process continues until all the nodes are reached.

After a matched service description is found, a query will still be passed along the circle until sufficient service providers are found. The Query is forwarded by using flooding technique. After searching in all nodes the set of matched service description is retrieved to the service consumer

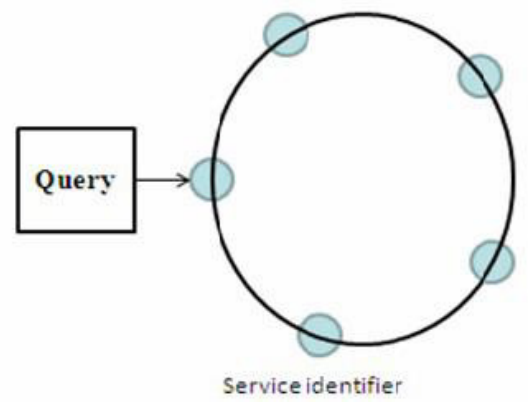
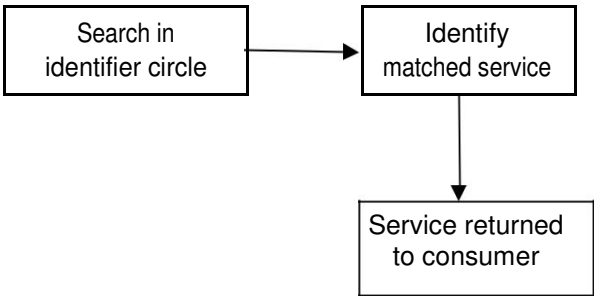


Figure 5: Search Process **4.3 Service Retrieval**

In this module after finding the matched service descriptions the requested service will be retrieved and send back to the user who is requesting the service. In the service description the functionally equivalent services with filename, file type, service identifier.

After retrieving a user can copy the file and place it in specific location by using user login, Also a user can directly open the file by using user login and make use of this file.



Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

Figure 6: Service Retrieval **4.4 Open File Location**

After retrieving the files the service consumer can open the file and its location. Along with retrieved matched description the path of the file is also retrieved. By using this path we can access the location of the file service. Also the retrieval of data is much faster and we can open the location of original file.

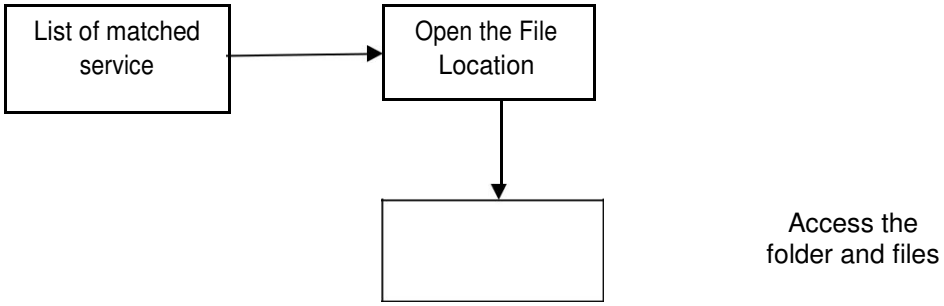


Figure 7: Open file location

4.5 Security

For opening the specific location authentication system is used. Only authorized persons can open the particular file location. By using this we can avoid the unauthorized user's access the files and folders.

5. ALGORITHM: Search algorithm

For searching the matched service description a message identifier and a finger table is used. By using this the functionally equivalent services are retrieved efficiently. The pseudo code that implements the service discovery process is used for search process. When looking for matched successor nodes, node n checks the entries of its finger table to find the node with the smallest identifier in the target virtual segment which is about to take responsibility of keeping routing the query. Thus the algorithm always makes progress until sufficient service descriptions have been found. Function closest_preceding_finger is used to request n to find the node known by n that most closely precedes message.id. However, the implementation is different from Chord as a result of our novel hierarchical structure for service descriptions. The matched service descriptions are retrieved to the service consumer.

Designing efficient search algorithms is a key challenge in unstructured peer-to-peer networks. Flooding and random walk (RW) are two typical search algorithms. Flooding

DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

searches aggressively and covers the most nodes. However, it generates a large amount of query messages and, thus, does not scale. On the contrary, RW searches conservatively. It only generates a fixed amount of query messages at each hop but would take longer search time. We propose the dynamic search (DS) algorithm, which is a generalization of flooding and RW. DS takes advantage of various contexts under which each previous search algorithm performs well. It resembles flooding for short-term search and RW for long-term search. Moreover, DS could be further combined with knowledge-based search mechanisms to improve the search performance. We analyze the performance of DS based on some performance metrics including the success rate, search time, query hits, query messages, query efficiency, and search efficiency. Numerical results show that DS provides a good tradeoff between search performance and cost. On average, DS performs about 25 times better than flooding and 58 times better than RW in power-law graphs, and about 186 times better than flooding and 120 times better than RW in bimodal topologies.

6. EXPERIMENTAL EVALUATION:

To evaluate the performance of Chord4S, a Chord simulator is extended to support Chord4S topology control, data distribution and routing protocol. As the configuration and operation of the underlying overlay network is based on Chord, Chord4S inherits good scalability with low communication cost.

Data Availability

A unique feature, and also a main design goal of Chord4S is the high data availability in volatile environments. To set up the volatile environments, we randomly select a fraction of nodes participating in the network to fail in each experiment, increasing from 5 to 60 percent by steps of 5 percent. Then the remaining nodes were randomly selected to send queries for services.

Routing Performance

To evaluate the routing performance of Chord4S, the average number of hops needed for a query of a certain number of functionally equivalent services was measured. The number of required services per query is randomly picked from the interval. Note that when only one service is required, the discovery process equals to that of Chord and the routing completes when the first matched service is found.

7. CONCLUSION:

Service discovery is a critical component in service-oriented computing. Over recent years, peer-to-peer-based service discovery has attracted researchers' attention after the deficiencies of centralized service discovery are identified. This paper has proposed Chord4S, a peer- to-peer-based approach for decentralized service discovery. To improve data availability, Chord4S distributes the descriptions of

DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

functionally equivalent services. Chord4S is scalable, reliable, and robust due to the enhanced peer-to-peer architecture. Experimental results demonstrate that Chord4S can achieve high data availability and efficient query of multiple functionally equivalent services in different successor nodes.

8. REFERENCES:

[1] L. Clement, A. Hately, C. von Riegen, and T. Rogers,"UDDIVersion3.0.2,"OASIS,http://www.uddi.org/pubs /uddi_v3.htm,2004.F.

[2] Emekci, O.D. Sahin, D. Agrawal, and A.E. Abbadi, A Peer-to-Peer Framework for Web Service Discovery with Ranking," Proc.IEEE Int'l Conf. Web Services (ICWS '04), pp. 192-199, 2004.

[3] D. Novak and P. Zezula, "M-Chord: A Scalable Distributed similarity Search Structure," Proc. First Int'l Conf. Scalable Information Systems, 2006.

[4] B. Sapkota, D. Roman, S.R. Kruk, and D. Fensel, "Distributed WebService Discovery Architecture," Proc. Advanced Int'l Conf.Telecomm. and Int'l Conf. Internet and Web Applications and Services. 136, 2006.

[5] C. Schmidt and M. Parashar, "A Peer-to-Peer Approach to Web Service Discovery," World Wide Web, vol. 7, no. 2, pp. 211-229, 2004.

[6] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (SIGCOMM '01), pp. 149-160, 2001.

[7] P. Rompothong and T. Senivongse, "A Query Federation of UDDI Registries," Proc. First Int'l Symp. Information and Comm. Technologies, pp. 561-566, 2003.

[8] S. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to- Peer Internet Telephony Protocol," Proc. IEEE INFOCOM, pp. 1-11, 2006.

[9] H.V. Jagadish, B.C. Ooi, K.-L.Y. Tan, and R. Cui Zhang, "iDistance: An Adaptive B+-Tree Based Indexing Method for Nearest Neighbor Search.

[10] S. Sanghan and M.M. Hasan, "Intelligent P2P VoIP through Extension of Existing Protocols," Proc.Ninth Int'l Conf. Advanced Comm. Technology (ICACT '07), pp. 1597-1601, and 2007.

[11] B.Y. Zhao, L. Huang, J. Stribling, A.D. Joseph, and J. Kubiatowicz, "Exploiting Routing Redundancy via Structured Peer-to-Peer Overlays," Proc. IEEE 11th Int'l Conf. Network Protocols (ICNP '03), pp. 246-257,2003

Special Issue; March - 2018 [ISSN: 2455-1457]
DOI : 10.23883/IJRTER.CONF.02180328.009.E1RN6

[12] Foster, H., Uchitel, S., Magee, J., Kramer, J. "Modelbased Verification of Web Service Compositions", Proc of 18th IEEE International Conference on Automated Software Engineering (ASE'03), Montreal, Canada, 2003,
IEEE Computer Society, pp. 152-163.

[13] Gopalakrishnan, V., Silaghi, B. D., Bhattacharjee, B.,Keleher, P. J. "Adaptive Replication in Peer-to-Peer Systems",Proc of 24th International Conference on Distributed Computing Systems (ICDCS'04), Hachioji, Tokyo, Japan, 2004, IEEE Computer Society, pp. 360-369.

[14] Li, Y., Zou, F., Wu, Z., Ma, F. "PWSD: A Scalable Web Service Discovery Architecture Based on Peer-to-Peer Overlay Network", Proc of 6th Asia-Pacific Web Conference
on Advanced Web Technologies and Applications (APWeb'04), Hangzhou, China, 2004, pp. 291-300.

[15] Narayanan, S., McIlraith, S. A. "Simulation, Verification and Automated Composition of Web Services", Proc of 11th International Conference on World Wide Web (WWW'02), Honolulu, Hawaii, USA, 2002, ACM, pp. 77-88.

