Scalable Mobile Web Service Discovery in Peer to Peer Networks

Satish Narayana Srirama [1], Matthias Jarke [1,2], and Hongyan Zhu [1]
[1] RWTH Aachen, Informatik V
Ahornstrasse 55,
52056 Aachen, Germany
Email: {srirama, jarke, zhu}@cs.rwth-aachen.de

Wolfgang Prinz [1,2]
[2] Fraunhofer FIT
Schloss Birlinghoven
53754 Sankt Augustin, Germany
Email: wolfgang.prinz@fit.fraunhofer.de

***Abstract*—Due to the astonishing development in memory and processing capabilities of hand held devices such as smart phones, it is not a dream anymore to enable mobile devices not only as conventional web service requesters but even as providers. The willingness and enthusiasm of service providers place abundant services at the disposal. But this abundance makes the efficiency of service discovery a critical issue. Centralized registries have severe drawbacks in such a scenario due to the dynamic and spontaneous nature of mobile peers. In the quest for a more appropriate approach for mobile web service discovery, we observed P2P to share very similar characteristics with behaviors of peers in mobile network. Hence we tried to find alternate mobile web service discovery mechanisms by using the features of the P2P networks like JXTA modules. The scalability analysis of the approach proves that the discovery can scale to the needs of large cellular networks.**

I. Introduction

In 2008 the number of global wireless subscribers is expected to reach 2 billion. This widespread global popularity of mobile devices among consumers and the generosity of R&D input in mobile technology by producers embrace each other with great enthusiasm. As a result, both hardware and software in mobile computing domain are developed with an astonishing speed in recent years. The next generation devices such as smart phones and PDAs are enabled to conduct tasks almost like personal computers and bring endless possibilities for wireless communication. Meanwhile web services technology is designed to support interoperable machine to machine interaction over a network so that applications could communicate with each other directly in order to exchange data or conduct a task. Recent developments of 3G and 4G technologies have significantly increased the wireless data transmission rates and paved a broad road of possibilities for mobile web service applications [1].

The basic web service architecture is still preserved in mobile web service domain where mobile phones act both as web service clients and providers. Thanks to numerous research around mobile web service clients [2], [3] and availability of many mature software tools [4], [5], we focus our research on mobile web service provisioning. In one of our previous projects, Srirama et al have developed a *Mobile Host* [6] with the capability of providing basic web services from smart phones. The feasibility of mobile web service provisioning is proved by extensive performance analysis of Mobile Host as well as further developed applications.

Due to their popularity a great amount of applications for mobile web services are available and more are under development. They build up on one hand rich resource for mobile clients, on the other hand push discovery mechanism in the spotlight. An effective discovery mechanism is necessary and an efficient one is critical to reduce bottlenecks and proceed the mobile web service provisioning and invocation with success. The frequently used centralized registry UDDI in web services is designed for stable networks and can not cater for the dynamic and spontaneous nature of mobile nodes. Therefore, the search for an appropriate mechanism for the mobile web service discovery is of urgent need.

This paper is a response to this need and call. We are proposing an alternative viewpoint as well as mechanism - peer to peer (P2P) technology to advertise and discover web services in mobile domain. With JXTA, a project aimed to provide services and infrastructure for P2P applications, we developed a mechanism for publishing and discovering mobile web services from smart phones with reasonable performance latencies. This paper intends to elaborate our approach, the scalability evaluation results and share our experience. The rest of the paper is organized as follows:

Section II discusses the concepts of mobile web services, their provisioning and access in P2P networks. Section III discusses the mobile web service discovery approach along with improvements achieved from categorization and advanced matching and filtering of mobile web services. Section IV discusses the scalability evaluation and results of mobile web service discovery approach and section V concludes the paper and proposes future research directions.

II. Mobile Web Services in P2P Networks

A. *Mobile Web Services*

In mobile web services domain, the resource constrained mobile devices are used as both web service clients and providers. Web services have a broad range of service distributions and on the other hand cellular phones have large and swiftly expanding user base. Combining these two domains brings us a new trend and lead to manifold opportunities to mobile operators, wireless equipment vendors, third-party application developers, and end users. By following the basic web services architecture, mobile web services enable communication via open XML web service interfaces and standardized protocols also on the radio link, where today still proprietary, and application- and terminal-specific interfaces are required. To support the mobile web services, there exist many organizations such as OMA [7], LA [8] on the specifications front; some practical data service applications such as over-the-air provisioning (OTA), application handover etc. on the commercial front; and SUN, IBM toolkits [4], [5] on the development front. Thus, though this is early stages, we can safely assume that mobile web services are the road ahead.

While mobile web service clients are quite common these days, the research with providing web services from smart phones is still sparse. However, a mobile device in the role of a service provider enables, amongst others, entirely new scenarios and end-user services. This paradigm shift from the role of service consumer to the service provider is also a step towards practical realization of various computing paradigms such as pervasive computing, ubiquitous computing, ambient computing and context-aware computing. In the mobile web service provisioning project one such Mobile Host was developed proving the feasibility of concept. Mobile Host is a light weight web service provider built for resource constrained devices like cellular phones. It has been developed as a web service handler built on top of a normal Web server. The SOAP based web service requests sent by HTTP tunneling are diverted and handled by the web service handler component. The Mobile Host was developed in PersonalJava on a SonyEricsson P800 smart phone. The footprint of the fully functional prototype is only 130 KB. Open source kSOAP2 [9] was used for creating and handling the SOAP messages.

The detailed evaluation of this Mobile Host clearly showed that service delivery as well as service administration can be done with reasonable ergonomic quality by normal mobile phone users [10]. As the most important result, it turns out that the total web service processing time at the Mobile Host is only a small fraction of the total request-response time ($<10\%$) and rest all being transmission delay. This makes the performance of the Mobile Host directly proportional to achievable higher data transmission rates. Similar implementations of Mobile Host were also possible with other Java variants like J2ME, for smart phones. We also have developed a J2ME based Mobile Host and its performance was observed to be not so significantly different from that of the PersonalJava version.

Once the Mobile Host was developed and its feasibility analyzed, extensive study was conducted in finding its specific application domains. The study was also aimed at growing Mobile Hosts' application scope; the research mainly focused on mobile community support and pervasive applications. During this study, it was observed that most of the targeted collaborative applications, somehow converged to Peer to Peer (P2P) [11] applications, and P2P offered a large scope for many applications with Mobile Host. Not just the enhanced application scope, the P2P also offers many technical advantages to the Mobile Host, like the better mobile web service discovery and access mechanisms. The approaches are discussed in the following sections.

B. Mobile Web Service Provisioning in P2P Networks

P2P is a set of distributed computing model systems and applications used to perform a critical function in a decentralized manner. P2P takes advantage of resources of individual peers like storage space, processing power, content, which are all critical for smart phones, and achieves scalability, cost sharing and anonymity, thereby enabling ad-hoc communication and collaboration. In order to reap the benefits of P2P, by achieving increased application scope, and targeting efficient utilization of resources of individual mobile peers, we tried to adapt Mobile Host into P2P networks. For this many of the current P2P technologies like Gnutella, Napster and Magi are studied in detail. Most of these technologies are proprietary and are generally targeting specific applications. Only project JXTA [12] offers a language agnostic and platform neutral system for P2P computing.

Combining JXTA and web services is not a completely new concept and at least for standalone systems some effort was already done by the research community. Here we mention some of these approaches. JXTA-SOAP project was started by Kevin Burton. The JXTA-SOAP is a package which allows SOAP communication over the JXTA P2P network. Qu and Nejdl [13] discussed the exposition of existing JXTA services as web services; and also integrating web service enabled content providers into JXTA. Hajamohideen [14] discussed the use of SOAP services as web services in JXTA. The SOAP service is a customization of the peer group service with ability to send and receive SOAP messages via a JXTA transport.

Moreover JXTA community has developed a light version of JXTA for mobile devices, called JXME (JXTA for J2ME). Considering JXME eliminates many of the low level details of the P2P systems like the transportation details. The mobile peers can communicate with each other using the best of the many network interfaces supported by the devices like Bluetooth, WiFi, GPRS etc. Considering these advantages and features of the JXTA, the Mobile Host was adapted into the JXTA network, to check its feasibility in P2P networks. Figure 1 shows the architecture of final deployment scenario of Mobile Hosts in the JXME network [15].

As shown in figure 1, the *virtual P2P network* also called the *mobile P2P network* is established in the mobile operator network with one of the nodes in operator proprietary network, acting as a JXTA super peer. JXTA network supports different types of peers to be connected to the network. The general peers are called *edge peers*. An edge peer registers itself with a *rendezvous peer* to connect to the JXTA network. Rendezvous peers cache and maintain an index of advertisements published by other peers in the P2P network. Rendezvous peers also participate in forwarding the discovery requests across the P2P network. A *relay peer* maintains route information and routes

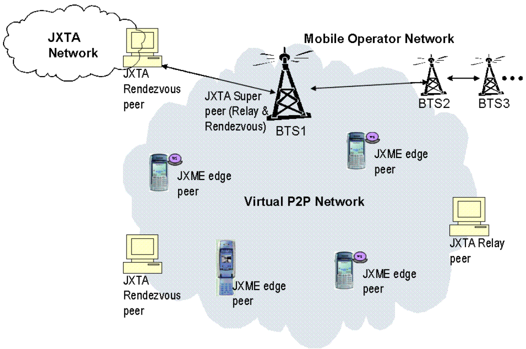

Fig. 1. Virtual mobile P2P network with Mobile Hosts

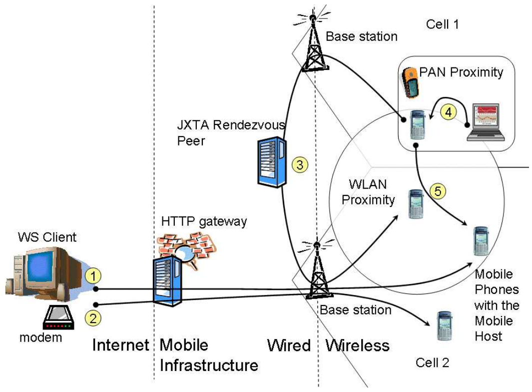

Fig. 2. Mobile web service provisioning and interactions

messages to peers behind the firewalls. A *super peer* has the functionality of both relay and rendezvous peers. In the mobile P2P network, the super peer can exist at Base Transceiver Station (BTS) and can be connected to other base stations, thus extending the JXTA network into the mobile operator network. Any Mobile Host or mobile web service client in the wireless network can connect to the P2P network using the node at base station as the rendezvous peer. The super peer can also relay requests to and from JXTA network, to the smart phones. Standalone systems can also participate in such a network as both rendezvous and relay peers, if the operator network allows such functionality, further extending the mobile P2P network.

C. Mobile Terminal Access in P2P Networks

Once a web service is developed and deployed with the Mobile Host, the mobile terminal, that is registered and connected within the mobile operator network, requires some means of identification and addressing that allows the web service to be accessible also from Internet. Generally, computers and devices in a TCP/IP network are identified using an IP address. The IP address, that is required for the data transfer to and from smart phones (as for any other IP communication client as Web servers, Intranet workstations, etc.), is assigned during the communication configuration phase. Typically, the IP address assigned to mobile devices using GPRS connection is only temporarily available, and is known only within the mobile operator's network, which makes it difficult to use the IP address in the client applications. Very few operators in the market today provide the facility that provides the smart phone with the public IP in GPRS network. The operational setup for accessing the mobile terminal in a GPRS network is shown in figure 2 with the interaction numbered 1. The mobile TCP/IP connection between the web service client and the Mobile Host is deployed on top of a GPRS link into the mobile operator network. From there the traffic is routed through the Internet to/from the web service client. The problem of addressing each mobile node with IP is not a big issue and it could be solved with Mobile IP version 6 (*Mobile IPv6*) [16]. The mobile web service provisioning project also has identified other means of addressing the Mobile Host in HSCSD (High-Speed Circuit Switched Data), shown in figure 2 with number 2. [10]

The need for public IP for each of the participating Mobile Hosts was observed to be the major hindrance for commercial success of the Mobile Host. So, alternative architectures were studied for the addressing of mobile web services. In a JXTA network, each peer is uniquely identified by a static peer ID, which allows the peer to be addressed independent of its physical address like DHCP based IP address. This peer ID will stay forever with the device even though the device supports multiple network interfaces like Ethernet, WiFi or Bluetooth for connecting to the P2P network. Hence, the success of the Mobile Host in the P2P networks makes the addressing of Mobile Host possible with peer ID. Now by using the peer ID, Mobile Host does not have to worry about changing IPs, operator networks, and is always visible to the web service client. Mapping the peer ID to the IP is taken care by the JXTA network, thus eliminating the need for public IP. This Mobile P2P interaction in figure 2 is numbered 3.

Provisioning of mobile web services in totally decentralized manner is even more challenging. This kind of interaction between peers is also referred as pure P2P. *Pure P2P* is a setup like the classic Gnutella file sharing network. The interactions numbered 4 and 5 in figure 2 represent this pure P2P network idea. In our case of mobile web services, this means that discovery, invocation and integration of web services occur between mobile devices directly without any centralized entities like base stations. We have not studied how to provide mobile web services according to this kind of technical usage scenario, but it promises to have the best cost-effectiveness as long as interactions between clients and providers of mobile web services do not involve proprietary mobile networks. Bluetooth could be a perfect technical solution for establishing such a pure P2P network. This kind of interactions tends to enable personal computing using various devices in Personal Area Network partially or fully based on mobile web services [17].

III. Mobile Web Service Discovery In JXME Network

In a commercial wireless environment with Mobile Hosts, and with each Mobile Host providing some services for Internet, the number of services expected to be published could be quite high. Generally Web services are published by advertising WSDL descriptions in a UDDI registry [18]. But with huge number of services possible with Mobile Hosts, a centralized solution is not a best idea, as they can have bottlenecks and can make single points of failure. Besides, mobile networks are quite dynamic due to the node movement. Devices can join or leave network at any time and can switch from one operator to another operator. This makes the binding information in the WSDL documents, inappropriate. Hence the services are to be republished every time the Mobile Host changes the network.

Dynamic service discovery is one of the most extensively explored research topics in the recent times. Most of these service discovery protocols are based on the announce-listen model like in Jini. In this model periodic multicast mechanism is used for service announcement and discovery. But all these mechanisms assume a service proxy object that acts as the registry and is always available. For dynamic ad hoc networks, assuming the existence of devices that are stable and powerful enough to play the role of the central service registries is inappropriate. Hence services distributed in the ad hoc networks must be discovered without a centralized registry and should be able to support spontaneous peer to peer connectivity. [19] proposes a distributed peer to peer Web service registry solution based on lightweight Web service profiles. They have developed VISR (View based Integration of Web Service Registries) as a peer to peer architecture for distributed Web service registry. Similarly Konark service discovery protocol [20] was designed for discovery and delivery of device independent services in ad hoc networks.

Considering these developments and our need for distributed registry and dynamic discovery, we have studied alternative means of mobile Web service discovery and realized a discovery mechanism in JXTA/JXME network [21].

A. Mobile Web Service Discovery Approach

In JXTA the decentralization is achieved with the advertisements. All resources like peers, peer groups and the services provided by peers in JXTA network are described using Advertisements. Advertisements are language-neutral metadata structures represented as XML documents. Peers discover each other, the resources available in the network and the services provided by peers and peer groups, by searching for their corresponding advertisements. Peers may cache any of the discovered advertisements locally. Every advertisement exists with a lifetime that specifies the availability of that resource. Lifetimes gives the opportunity to control out of date resources without the need for any centralized control mechanism. To extend the life time of an advertisement, the advertisements are to be republished.

Thus to achieve alternate discovery mechanism for mobile web services, the services deployed on Mobile Host in the JXTA network are to be published as JXTA advertisements, so that they can be sensed as JXTA services among other peers. JXTA specifies '*Modules*' as a generic abstraction that allows peers to describe and instantiate any type of implementation of behavior representing any piece of "code" in the JXTA world. So the mobile web services are published as JXTA modules in the P2P network. The module abstraction includes a *module class*, *module specification*, and *module implementation*. The module class is primarily used to advertise the existence of behavior. Each module class contains one or more module specifications, which contain all the information necessary to access or invoke the module. The module implementation is the implementation of a given specification. There might be more than one implementation for a given specification across different platforms. Figure 3 shows the mapping between JXTA modules and web services. The collection of module abstractions represent the UDDI in a sense of publishing and finding service description and WSDL in a sense of defining transport binding to the service.

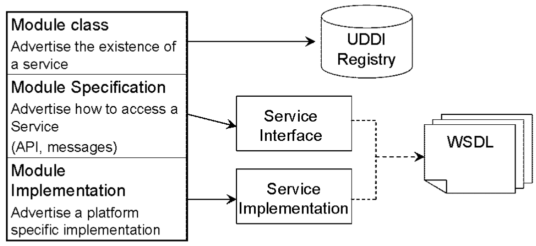

Fig. 3. Mapping between JXTA modules and web services

To publish the mobile web services in the JXTA network, a standard *Module Class Advertisement (MCA)* is published into the P2P network, declaring the availability of a set of web service definitions, in that peer group. Once new web services are developed for the Mobile Host, the WSDL descriptions of these services are incorporated into the `<parm>` element of the *Module Specification Advertisements (MSA)*, and are published into the P2P network. Thus MCA declares the existence of the web service and MSA provides metadata of the service. The MSAs are published into JXME network with an approximate life time that specifies the amount of time the Mobile Host wants to provide the service. The MSAs are cached at rendezvous peers or any other peers, with sufficient resource capabilities. Once the life time expires the MSAs are automatically deleted from the P2P network, thus avoiding the stale advertisements. The MSA can be published into the network by a service developer or even by the Mobile Host.

B. Advanced Matching/Filtering of Services

Once published to the mobile P2P network, the services can later be discovered by using the keyword based search provided by JXTA. The module specification advertisements carrying the web service descriptions can be searched by `name` and `description` parameters. This basic search returns a large number of resulted services, returning every service that matches the keyword. Since the discovery client in mobile Web services scenario is a smart phone, the result set should be quite small so that the user can scroll through the list and can select the intended services.

Moreover we would like to extend the search criteria to the WSDL level. This means that search parameters would not be restricted to module specification advertisement details. The search will also extend by looking up the WSDL tags and information. The main idea behind this approach is that people usually express their opinion by using frequently used words and the frequency of a keyword in WSDL description is also relevant. A similar approach is taken in UDDI Explorer tool, developed for searching standalone web services [22]. Hence the JXTA search resulted services are ordered according to their relevancy using *Apache Lucene* tool [23]. Lucene is an open source project that provides a Java based high-performance, full-featured text search engine library. Lucene allows adding indexing and searching capabilities to user applications. Lucene can index and make searchable any data that can be converted to a textual format. Using the tool and its index mechanism the search results were ordered/filtered and the advanced matched services were returned to the discovery client.

C. Categorization of Mobile Web Services

In order to make mobile web service discovery process more efficient, we tried to adapt categorization feature in UDDI standards. The ability to attribute metadata to services registered in UDDI and then the ability to run queries based on the metadata is absolutely central and critical to the purpose of UDDI at both design time and run time. To locate data with a UDDI registry would prove to be very difficult without categorization. Especially for the discovery of previously unknown businesses, services, binding or service types. It is indispensable that the corresponding UDDI registration data is marked with a set of categories that can universally be searched on. All four main UDDI data structure types provide a structure to support attaching categories to data. By providing a placeholder *categoryBag* for attaching categories to these data structures, any number of categories can be used for a variety of purpose. A categoryBag consists of one or more *keyedReference*. Each keyedReference contains attributes to identify category information. When the relationship between single categories is needed to describe a more complex entity, *keyedReferenceGroup*, which contains a set of keyedReference, is used to serve for the purpose.

With the intent to borrow categorization to our P2P-based mobile web service discovery mechanism, we also need a placeholder similar to categoryBag in UDDI. We called it *categoryPack*, which contains information of peer groups by way of keyedReferenceGroup. The categoryPack is incorporated into MCA. When an MSA is to be published in the network, it should search for the most appropriate MCA to match by the category information provided by keyedReferenceGroup in MCA, and should attach itself to the respective MCA.

For further achieving categorization we resorted to peer groups. With the reference to some popular industry categorization standards North American Industry Classification System (NAICS) and United Nations Standard Products and Services Code (UNSPSC), we designed a hierarchical categorization structure for mobile web services as in Figure 4. It

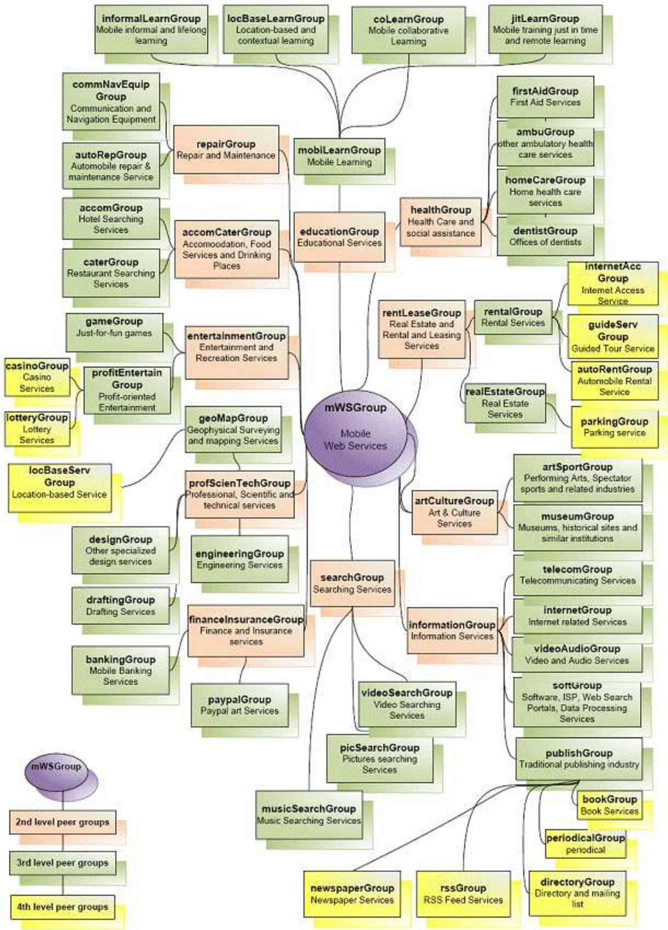

Fig. 4. Mobile web services category hierarchy

consists of four levels and fifty-seven peer groups altogether with mobile Web Service Group (*mWSGroup*) on the top level. The group structure is a first draft to realize the idea of categorization. To construct a complete categorization hierarchy for all possible mobile web services is neither realizable for the present stand of our research nor meaningful for our need. The application of the hierarchical categorization structure is implemented by a shell command in JXTA platform. With this self-defined shell command the whole categorization structure is published into JXTA network.

With the peer group hierarchy in place, a potential service requester first joins his group of interest and then issues the query for the discovery. When a super peer that is in the peer group receives the request, it looks through the cache and sends the request to the still available peers in the group. Each peer that receives the request from its super peer conducts the search process in its local cache by searching for the MCAs according to the keyword given by the mobile requestor. An MCA maps the general description of web services under the same group. Since more than one web services could possess the same general description but different detailed description and content, one MCA could possibly match more than one MSA. After the discovery process the matched MSAs are provided as the search results. If the number of found MSAs is small, then the searching results could already be accepted and sent back to the service requester. Otherwise the Lucene based Advanced Matching/Filtering of Service is performed on the result set and the most relevant services are returned to the mobile client.

IV. Scalability Evaluation of Mobile Web Service Discovery

Scalability is an acknowledged desirable property for network performance. A system is described as scalable if it is able to accommodate an increasing number of elements or objects and/or to process growing volume of work gracefully. According to the factors taken into consideration there are different types of scalability, which are structural, space, time and load scalability. In order to determine the most appropriate types of scalability test for mobile web service discovery we reviewed the related projects. The broadest research to measure the scalability of JXTA is JXTA Benchmarking project, whose goal is to construct a test harness which can be set up by people relatively unfamiliar with JXTA, operated more or less automatically (scripted), which reports on network performance measures of interest to the JXTA community. Two other projects further implement JXTA Benchmarking project to some extent regarding performance evaluation of typical peer operations and sequences [24] and the performance of peerview and discovery protocols by large-scale, multi-site experiments [25] respectively.

With the above mentioned three projects as reference, we conducted the scalability test on the basis of peer operations in the process of discovery under different topologies. In our test startup benchmark is set as main metric for pre-discovery stage and Round-Time Trip (RTT) benchmark is set as main metric for discovery stage. Our benchmark suit included four network topologies where the numbers of participating rendezvous peers (RDV) are 0, 1, 2 and 3 respectively. The topologies are shown in figure 5. We considered discovery mechanism both with and without categorization.

The hardware setup for the evaluation included a Sony Ericsson P990i smart phone and a pool of eight computers. The phone has a memory of 60 MB and 3G technology capability with data transfer speeds up to 384 kbps for Internet. The computers are equipped with Pentium IV, 3.2GHz and RAM of 1GB or 2GB. The network environment is campus 100 Mbps LAN at RWTH Aachen, Germany. JXTA-JXSE version 2.4.1 is used to execute the test.

For obtaining the performance results, large no of MSAs (in the range 10-100) are published into the group categorization hierarchy. All the experiments are conducted in all the 4 different topologies, under multiple number of matching services for the keywords. In the case of single peer topology with no RDV, non-categorization discovery is comparatively easier to conduct. Since the only variable is the number of MSAs to be published. The discovery latency is measured in each case of different number of MSAs in the local cache of the single peer. In categorization discovery the measurement

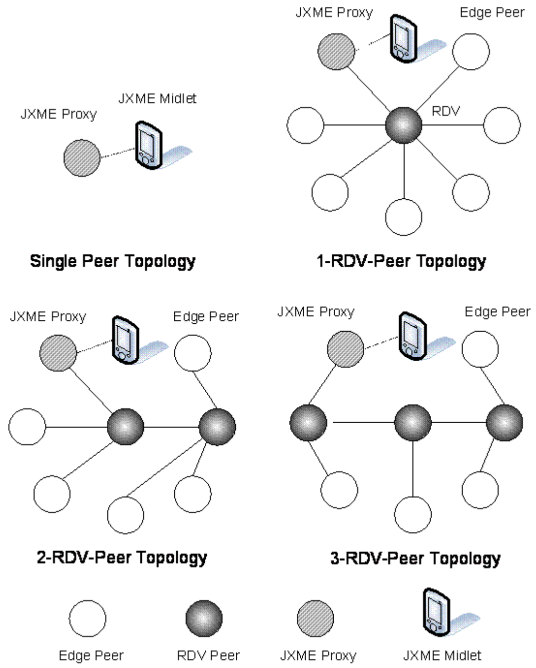

Fig. 5. Scalability test topologies

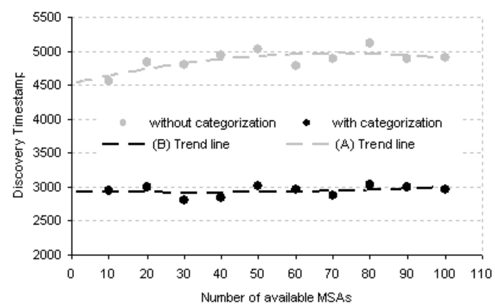

Fig. 6. Comparison of timestamps for topology with 2 RDV

is more complicated. Since four levels of peer groups are built in the category hierarchy and with the concern that the discovery latency may be rather diversified on different levels of category hierarchy, we measured the performance from each level of hierarchy. With number of MSAs to be published as the variable, discovery latency is measured from groups of each of the four levels.

Figure 6 shows the results for the topology with 2 rendezvous peers. From these results we can observe that It costs about 41% less latency for a peer to find an MSA with discovery mechanism with categorization than non- categorization discovery mechanism. The trend line of non-categorization discovery mechanism shows a light linear form, while that of discovery mechanism with categorization keeps rather constant trend line. The mean of categorization discovery is 2872 ms, in comparison to 4866 ms.

From the scalability results we also have observed that the percentage difference for categorization and non-categorization discovery mechanisms in case of topology with no RDV is approximately 50%. Moreover, one or two more RDV on the route of discovery bring no dramatic effect to the discovery time and trend. The results from Two-RDV and Three-RDV topology are very much alike to One-RDV topology. The non-categorization discovery approach keeps the tendency of mild linear growth, while categorization discovery approach leads to almost constant discovery time hardly affected by the growing number of available MSAs in peer groups.

These results indicate that the use of categorization in discovery process obviously improved the performance by 40% to 50% in average. With the addition of more RDV peers, the discovery time of non-categorization version grows lightly, which could lead to undesirable scalable property in large-scale network. In contrast the discovery time with categorization mechanism does not grow much, if at all, with the addition of more RDV peers. Although it is just a small-scale test result, we could still conclude that the design of mobile web service discovery mechanism with categorization could very probably bring scalable performance even under the large-scale networks.

V. Conclusion

The paper addressed the concept of mobile web service provisioning and discovery in P2P networks. First it explained the details of Mobile Host's entry into P2P network and later described the technical advantages associated with it for the mobile web services, in terms of alternative addressing and discovery mechanisms. The mobile web service discovery approach makes use of the JXTA modules feature for publishing and discovering the services. We presume the approach clearly solves the problem of discovering huge number of services possible in mobile networks. The scalability evaluation results further elevate the point and prove that the discovery approach can scale to the needs of large cellular networks. The performance latencies are also reasonable and are about 3 seconds for discovery with categorization.

Acknowledgment

This work is supported by the Research Cluster Ultra High-Speed Mobile Information and Communication (UMIC) at RWTH Aachen University (http://www.umic.rwth-aachen.de/). The authors would also like to thank R. Levenshteyn and M. Gerdes of Ericsson Research for their help and support.

References

[1] 4GPress, "World's First 2.5Gbps Packet Transmission in 4G Field Experiment," 4G Press, 2005. [Online]. Available: http://www.4g.co.uk/PR2006/2056.htm

[2] N. Balani, "Deliver web services to mobile apps," IBM DeveloperWorks, January 2003. [Online]. Available: http://www-128.ibm.com/developerworks/wireless/edu/wi-dw-wiwsvs-i.html

[3] J. Ellis and M. Young, "J2ME Web Services 1.0 - Final Draft (JSR 172)," Sun Microsystems, Inc., Tech. Rep. 11, October 2003.

[4] Sun Microsystems, "Sun Java Wireless Toolkit for CLDC," Sun Developer Network, 2007. [Online]. Available: http://java.sun.com/products/sjwtoolkit/

[5] IBM Corporation, "IBM WebSphere Studio Device Developer," IBM developerWorks, 2007. [Online]. Available: http://www-306.ibm.com/software/wireless/wsdd/

[6] S. N. Srirama, M. Jarke, and W. Prinz, "Mobile web service provisioning," in *AICT-ICIW '06: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*. IEEE Computer Society, 2006, p. 120.

[7] OMA, "Mobile Web Services Requirements - Version 1.1," Open Mobile Alliance Group, Tech. Rep., March 2006.

[8] J. Tourzan and Y. Koga, "Liberty ID-WSF Web Services Framework Overview - Version: 2.0," Liberty Alliance Project, Tech. Rep., 2006. [Online]. Available: http://www.projectliberty.org/liberty/content/download/889/6243/file/liberty-idwsf-overview-v2.0.pdf

[9] kSOAP2, "kSOAP2 - An efficient, lean, Java SOAP library for constrained devices," SourceForge.net, 2007. [Online]. Available: http://sourceforge.net/projects/ksoap2

[10] S. Srirama, M. Jarke, and W. Prinz, "Mobile Host: A feasibility analysis of mobile Web Service provisioning," in *4th International Workshop on Ubiquitous Mobile Information and Collaboration Systems, UMICS 2006, a CAiSE'06 workshop*, June 2006, pp. 942–953.

[11] C. Shirky, K. Truelov, R. Dornfest, and L. Gonze, *P2P Networking Overview*. O'Reilly, October 2001.

[12] B. J. Wilson, *JXTA*. New Riders Publishing, June 2002.

[13] C. Qu and W. Nejdl, "Interacting the edutella/jxta peer-to-peer network with web services," in *Proceedings of the 2004 International Symposium on Applications and the Internet (SAINT'04)*, 2004.

[14] S. H. Hajamohideen, "A model for web service discovery and invocation in jxta," Master's thesis, Technical University Hamburg-Harburg, March 2003. [Online]. Available: http://www.ti5.tu-harburg.de/publication/2003/Thesis/haja03/haja03.pdf

[15] S. Srirama, "Publishing and discovery of mobile web services in peer to peer networks," in *Proceedings of First International Workshop on Mobile Services and Personalized Environments (MSPE'06)*, vol. P-102. Lecture Notes in Informatics, GI, November 2006, pp. 15–28.

[16] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6 (RFC 3775)," Network Working Group, Tech. Rep., June 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt

[17] A. Naumenko, S. Srirama, V. Terziyan, and M. Jarke, "Semantics-based access control for mobile web services," *International Journal on Semantic Web and Information systems, Special Issue on Mobile services and Ontologies (Submitted for Publication)*, 2007.

[18] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to web services architecture," *IBM Systems Journal: New Developments in Web Services and E-commerce*, vol. 41(2), pp. 178–198, 2002. [Online]. Available: http://researchweb.watson.ibm.com/journal/sj/412/gottschalk.html

[19] S. Dustdar and M. Treiber, "Integration of transient web services into a virtual peer to peer web service registry," *Distributed and Parallel Databases*, vol. 20, pp. 91–115, 2006.

[20] C. Lee, A. Helal, N. Desai, V. Verma, and B. Arslan, "Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services," *IEEE transactions on systems, man, and cybernetics - part a: systems and humans*, vol. 33, no. 6, pp. 682–696, November 2003.

[21] S. N. Srirama, M. Jarke, and W. Prinz, "Mobile web service discovery in peer to peer networks," in *5th International Workshop on Ubiquitous Mobile Information and Collaboration Systems, UMICS 2007, a CAiSE'07 workshop*, June 2007, pp. 531–542.

[22] T. Dao, "Uddi explorer: Tool for searching web services," The Code Project, December 2005.

[23] D. Cutting, "Apache Lucene," 2007. [Online]. Available: http://lucene.apache.org/

[24] E. Halepovic and R. Deters, "The costs of using jxta," in *The Third IEEE International Conference on Peer-to-Peer Computing*, 2003.

[25] G. Antoniu, L. Cudennec, M. Jan, and M. Duigou, "Performance scalability of the jxta p2p framework," in *IEEE Parallel and Distributed Processing Symposium, 2007*, 2007.