MDE-URDS – A Mobile Device Enabled Service Discovery System

Ketaki A. Pradhan, Lahiru S. Gallege and Rajeev R. Raje

Department of Computer and Information Science, IUPUI
{ketpradh, lspileth, rraje}@cs.iupui.edu

***Abstract*—**Service Discovery is an important step during the creation of distributed systems made out of independently developed and deployed quality-aware components. Despite of many prevalent approaches, the task of service discovery that encompasses components running on resource constrained devices does not yet have a satisfactory solution. Here, we present a service discovery framework which is specially designed to include resource constrained devices and uses the Mobile IP and the MANET protocols. A preliminary prototype of this framework, along with associated experiments, is described in this paper.

***Index Terms*— Resource Constrained Devices, Service Discovery, UniFrame, Mobile IP, MANETs.**

I. INTRODUCTION

Service Discovery has been a well studied field in recent times. Its wide spread applicability, before the composition of independently developed and deployed components makes it a critical activity before creating a distributed system. This is evident from many alternatives such as Jini [1], UPnP [2], and SLP [3]. A comprehensive report [4], on first generation of service discovery system indicates the challenge of the incorporation of mobile and resource constrained devices into the service discovery systems. This research aims to address this specific challenge. As an initial step, in this paper, we describe the incorporation of the principles of the Mobile IP [5] and the collaborative approaches from the Mobile Ad-hoc Networks (MANET) domain into the UniFrame Resource Discovery System (URDS) [6], to create a mobile device enabled discovery system called MDE-URDS (Mobile Device Enabled URDS).

II. Related Work

There have been several first-generation service discovery architectures proposed that can be divided into mainly two groups: a) Lookup Services such as Jini, UPnP, CORBA Trader Service [7], UDDI [8] that are registry based discovery services and b) Discovery Services that include specific architecture for resource discovery, such as SLP. Not many of these first generation approaches take into account the need for incorporating mobile devices in the service discovery process and operate in a homogeneous environment. FRODO [9] is a recent approach that includes mobile devices in the service discovery process by categorizing them based on their capabilities of energy (power) and mobility. However, their framework is designed for a particular home environment, without considering the other domains where the mobile entities could migrate to. M-URDS [10] is another extension of the URDS architecture that uses mobile agents to discover new components. It, however, does not consider the incorporation of mobile devices in the URDS framework.

Our work differs from these approaches, as it addresses the key issues of heterogeneity (due to being an extension of the URDS), mobility associated with the resource constrained devices (by incorporating the principles of Mobile IP) and the use of collaborations (by the use of MANET routing protocols) in the service discovery process.

III. OUR SOLUTION – MDE-URDS

The architecture of MDE-URDS is shown in Figure 2.1. The main components of the URDS are the Internet Component Broker (ICB), Headhunters, and Active Registries. The ICB, which is similar to the Object Broker in CORBA, consists of the following entities: a) Domain Security Manager (DSM), which is responsible for maintaining the information about all the entities in the system and provides that information to only authorized entities, such as the Headhunters, b) Query Manager (QM), which is responsible for making queries on behalf of the client of the URDS and getting the results from the Headhunters, c)Adapter Manager (AM) is used for handling heterogeneity by providing the necessary adapter components, d)Link Manager (LM) links several of such ICBs together to create a federation of discovery services, e) Headhunter (HH) is one of the important entities of the URDS, which actively participates in the discovery process of software components and performs the matching of the available components with the input queries. Headhunters also have associated meta-repositories, f) Active Registries are responsible for looking out for new components entering in the URDS and registering them.

For MDE-URDS, we divided the resources, into two categories: resourceful and resource-constrained. We identified a mapping scheme for placing the URDS entities (e.g., HHs) into these two types of resource classes based on about the functionality of each entity and associated empirical evaluations. Table 1 indicates the result of the mapping process.

URDS Component	Resourceful device	Resource-constrained device
Domain Security Manager	Yes	No
Query Manager	Yes	Yes
Link Manager	Yes	No
Adapter Manager	Yes	No
Headhunter	Yes	Yes
Active Registries	Yes	Yes
Clients	Yes	Yes

Table 1: Mapping of URDS components on different devices.

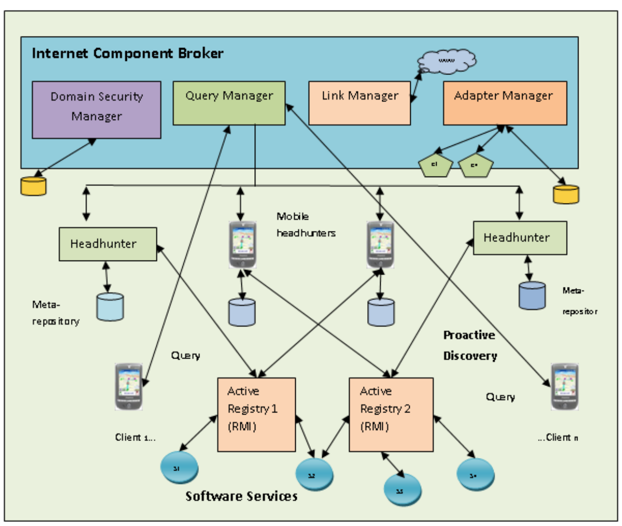

Fig 2.1: Basic MDE-URDS Architecture

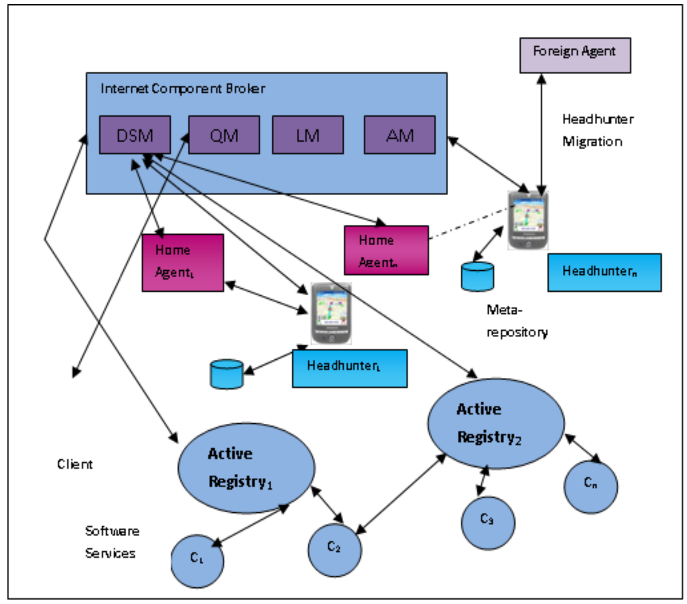

Fig 2.2: MDE-URDS Architecture with Mobile IP
Diagram Key: HH = Headhunter, HA = Home Agent, FA = Foreign Agent,
C1, C2,, …, Cn = Software Components.

To include Mobile IP principles, new entities such as Home Agents and Foreign Agents were added into the basic architecture of MDE-URDS. In the figure 2.2, the components colored blue are the existing components of the MDE-URDS and the ones in the pink color are added for including the Mobile IP principles. As mobility is the key factor here, there must be a mechanism for the mobile entities to move across the network, and still be able to achieve the discovery functions. This is achieved by the use of the principles of Mobile IP, a protocol that provides transparent access to mobile entities moving across different domains with the help of Home Agents and Foreign Agents that help in routing the messages to every mobile entity.

In the context of MDE-URDS, every headhunter has a Home Agent that is responsible for successful query routing to it irrespective of its location. The query propagation to the headhunters on the mobile devices takes place as follows: if the mobile node is in its home world, then the query is sent directly to it, whereas when the mobile node is in any foreign world, then the query is routed from its Home Agent to its Foreign Agent and then to the mobile headhunter. Therefore, we have two scenarios, used in experimentation, for propagating queries to the mobile headhunters: a) when the headhunter is in its home world, b) when the headhunter is connected to an external foreign world and c) when the headhunter is in transit.

2.3. Use of Collaborative Approaches by using MANET

routing protocols: The quality analysis of the results obtained from this prototype of MDE-URDS was studied and it was found that with headhunter using collaborative approaches, the quality of the results obtained improves. In order to achieve this task, protocols from the Mobile Ad-hoc Networks (MANET) domain are used. MANET is a self-configuring mesh network where the individual mobile nodes (sensors) are connected by wireless links and coordinate with its neighbors for a particular task. There are various routing protocols used by the nodes that can be reactive as well as proactive. Amongst them, Ad-hoc On Demand Vector (AODV) Routing [11] and Dynamic Source Routing (DSR) [12] are reactive whereas Optimized Link State Routing (OLSR) [13] is proactive.

Similar to the MANETs, the mobile headhunters were connected in a topology and maintain neighbor information. The collaborative approaches used were as follows: "I do it"(where only one HH is queried), "You do it"(where HH passes the query to its neighbors) and "We do it" (where the results are obtained from the HH and its neighbors). Figure 2.4 shows the two employed approaches of "We do it" and "You do it".

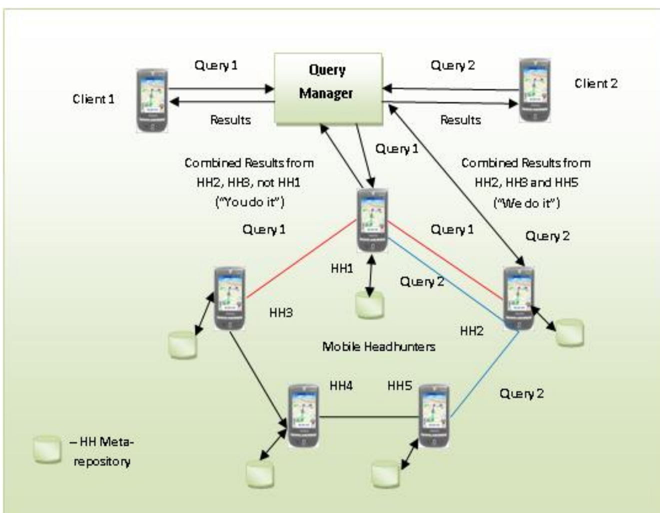

Fig. 2.4: Collaborative approaches of "We do it" and "You do it".

IV. EXPERIMENTATION

A prototype of MDE-URDS was created that contained Pharos Traveler GPS PDAs as the resource-constrained devices running Windows Mobile 5 operating system and twelve desktop machines that use Windows XP operating system. The desktop machines are connected by a LAN and wireless connectivity is provided by the different wireless networks on campus and also a private access point for a constrained experimental setup. The entire MDE-URDS implementation is created using Java RMI, and we have used J9 JVM [14] for running the java classes on the Pharos PDAs. We also divided the experimental setup into two "worlds" – a home world and a foreign world and ran several experiments, querying different headhunters for components, moving seamlessly across different worlds (home and foreign), sending queries simultaneously to them. We had deployed four headhunters on mobile devices, four stationary headhunters for comparing the response times, four active registries that search for new components.

For our setup, we have limited the number of components to a total of 15, and every headhunter has its own meta-repository that updates from the active registries. The meta-repositories have overlapping components. We calculated the average query response time for each mentioned scenario. The query structure is composed of the Headhunter name, its IP address, and the type of the component to be retrieved from the headhunter repository. Thus, the matching semantics used in the following experiments is restricted only to type matching of components. These queries are sent out randomly to the headhunters with proper sequence numbers to provide duplicate filtering and all the queries are independent of each other. In Figures 4.1-4.5, every point on the graph is the time value obtained from a particular headhunter.

First, we established a base case situation with headhunters deployed only on resourceful devices, used for comparison purposes. The average response time for this case, as shown in Figure 4.1, varies between 30 ms to 43 ms.

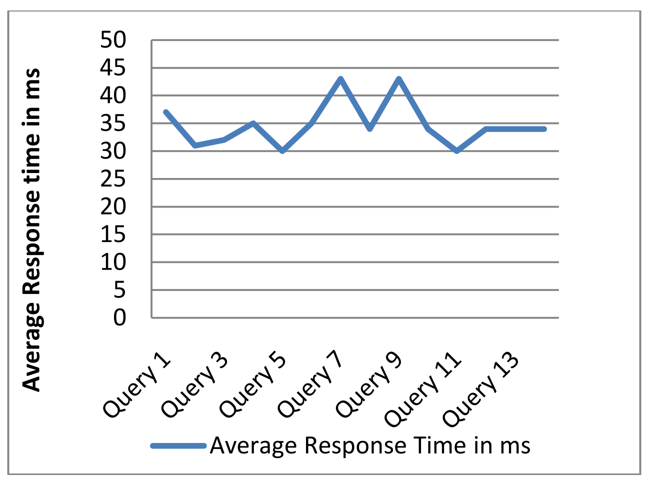

Fig. 4.1: Average response time when headhunters are on resourceful devices.

A .Scenario 1: Headhunters in home world

Figure 4.2 shows the average response time for random queries from headhunters deployed on PDAs which are in their home world. The average response time here varies between 630 ms and 830 ms. These values are high compared to the base case, because of the added time spent in communication with these wireless devices.

B. Scenario 2: Headhunters in foreign world

Figure 4.3 indicates the average response time, when the Headhunters are in the foreign world. Here, the average response time ranges between 1043 ms to 1350 ms. The increase is due to the rerouting of the queries to the Foreign Agent across different domains.

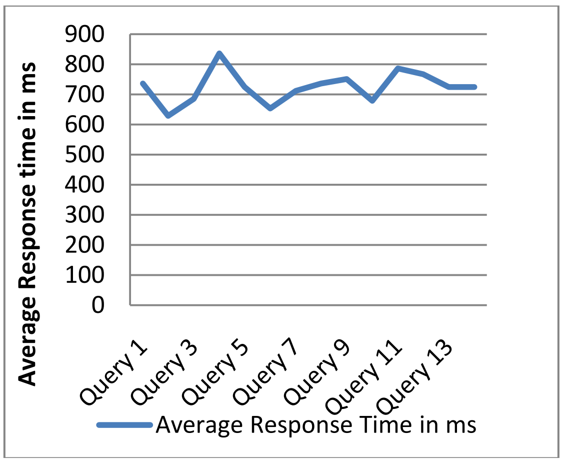

Fig. 4.2: Average response time when headhunters are in the home domain

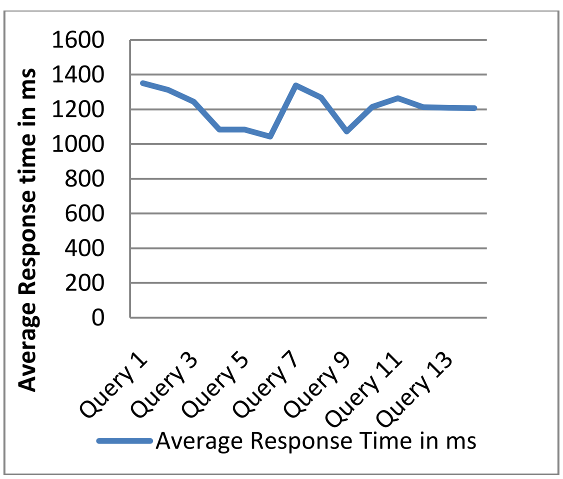

Fig. 4.3: Average Response Time when headhunters are in the foreign domain.

C. Scenario 3: Headhunters in transit

In this case, the response time varies according to whether there is a wait for the HH (in case of quality-aware application) or is passed to another HH (time-aware application).

We can conclude from these experiments that the average response time is higher using the resource-constrained devices as compared to resourceful devices. This is because of the additional time spent in communicating with these devices using the pair of Home Agent and Foreign Agent.

Results obtained from the Collaborative Approaches: The results obtained are evaluated in terms of response time and quality. Figures 4.4 and 4.5 indicate the response time obtained from the collaborative HHs which is high as now more HHs are queried. Table 2 and 3 indicate the quality of results obtained from collaborative HHs. It is seen that the recall of the results obtained is high in this case than from a single HH.

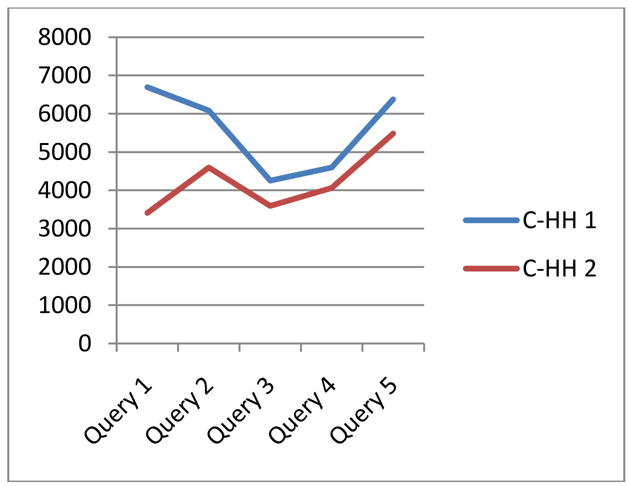

Fig. 4.4: We do it approach using AODV protocol.

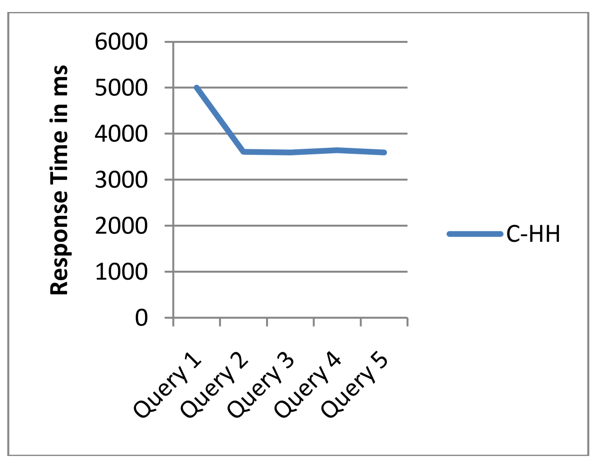

Fig. 4.5: Collaborative results for 'You do it' approach.

Evaluation of Quality:

Query	Quality of Results from single HH		Quality of Results with collaboration	
	Precision	Recall	Precision	Recall
1	0	0	0.5	0.4
2	0	0	0.6	0.833
3	0.5	0.2	0.3	0.4
4	1	0.125	0.75	0.375
5	1	0.166	0.727	0.66

Table 2: Precision and Recall for 'We do it' approach.

Evaluation of quality:

Query	Quality of Results from single HH		Quality of Results with collaboration	
	Precision	Recall	Precision	Recal l
1	0	0	0.5	0.4
2	0	0	0.75	0.5
3	0.5	0.2	0.5	0.4
4	1	0.125	0.66	0.25
5	1	0.16	0.66	0.5

Table 3: Precision and Recall comparison for 'You do it' approach.

V. CONCLUSION AND FUTURE WORK

In this paper, we have described the results obtained from incorporating resource-constrained mobile devices in a service discovery system. An empirical validation of the MDE-URDS prototype indicates an increase in the average response time due to the re-routing of queries and results carried out by the Home and Foreign Agents when compared with the URDS containing resourceful devices. The collaborative approaches used indicate that the quality of results is better when a collaborative approach is used, with the tradeoff of high response time. A direction for our future work is to incorporate uncertainty of the context in the discovery process.

ACKNOWLEGMENTS

The authors would like to thank my colleague Alfredo Moreno, for his help and support in the implementation of the system.

REFERENCES

1] Sun Microsystems, “Jini™ Architectural Overview”,http://pages.cs.wisc.edu/~lhl/cs740/papers/jini-overview.ps, 1999, accessed on September 25th, 2009.

[2] Universal plug and Play (UPnP), http://www.upnp.org/, accessed on September 25th, 2009.

[3] Guttman E., Perkins C., Veizades J., and Day M., “Service Location Protocol”, Version 2. IETF RFC 2608, 1999.

[4] Dabrowski C., Mills K., and Quirolgico, S., “A Model-based Analysis of First-Generation Service Discovery Systems”, NIST, ational Institute of Standards and Technology, Special Publication 500-260, October 2005.

[5] Perkins, C., “Mobile IP”, IEEE Communications Magazine, May 1997-Volume 35, Number 5.

[6] Siram, Nanditha N., Raje, R., Olson, A., Bryant, B., Burt, C., and Auguston, M., “An Architecture for the UniFrame Resource Discovery Service”, Springer- Verlag Lecture Notes in Computer Science, Vol. 2596, 2003, pp. 20-35.

[7] Common Object Broker Architecture (CORBA), http://www.corba.org/, accessed on 25th September, 2009.

[8] UDDI Technical White Paper, 2000. http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, accessed on 25th September, 2009.

[9] Sundramoorthy V., “FRODO High-level and Detailed Specifications”, PhD Thesis, University of Twente, Enschede, the Netherlands, 2006.

[10] Gandhamaneni, J., “UniFrame Mobile Agent Based Resource Discovery Service (MURDS)”, M.S. Project TR-CIS-1122-03, Department of Computer & Information Science, IUPUI, June, 28, 2004.

[11]Perkins C., Royer M. "An implementation study of the AODV routing protocol." In Proceedings of the IEEE Wireless Communications and Networking. Chicago,IL, 2000.

[12] Johnson D., Maltz D.A. and Broch J. "The dynamic source routing protocol for mobile ad hoc networks." Mobile Adhoc Network (MANET) Working Group. IETF, 1998.

[13] Clausen T., Jaqcuet P.,Laouiti A., Minet P., Muhlethaler P., Qayyum A.,Viennot L. "Optimized link state routing protocol." IETF Draft, 2003.

[14] IBM’s J9 JVM, http://brainmurmurs.com/products/timeout/docs/wm2003/j9Install.htm, accessed on 25th September, 2009.

Biography

Ketaki A. Pradhan is a Master of Science student in the Department of Computer and Information Science, IUPUI.
Lahiru S. Gallege is a student in the Department of Computer and Information Science, IUPUI and is currently pursuing his PhD.
Dr. Rajeev R. Raje is a Professor and the Associate Chair of the Department of Computer and Information Science, IUPUI.