SERVICE DISCOVERY PROTOCOLS USING JINI

ROSHAYATI BINTI YAHYA @ ATAN

A project report submitted in partial fulfilment of the requirements
for the award of the degree of Master of Engineering
(Electrical – Electronics & Telecommunication)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

NOVEMBER 2009

To my beloved daughter, Nuralya Raihanah

ACKNOWLEDGEMENT

First of all, Alhamdulillah because of His bless and giving me a chance to finish this project. Herewith, I want to thanks to my project supervisor, Dr. Sharifah Hafizah binti Syed Ariffin for her ideas, advices, time, and continuously supports during doing the project.

I also want to express my special gratitude to my husband, Mohd Fariz bin Abd Hamid, my father, Yahya @ Atan bin Mohamed, my mother, Aminah binti Abdullah and my siblings, Rosli, Rosnawati, Rosman, Rosniwati, and Rosmizan for their love and support.

Thank you very much too to all my MELA1AJA ex-classmates session 2008/2009, especially Asmida binti Ismail and Zaharah binti Johari that always by my side from the start till the end of this project.

Finally, thank you to all my friends who are either deliberately involve or not in helping me to complete the project.

ABSTRACT

Most of the people nowadays use the internet to find services. However, these will become more complicated to make decision due to the hundred numbers of service providers floating on the Internet. The main purpose of this project is to develop a network protocol that is able to discover any devices or services in a computer network. This service discovery protocol may automatically allow the user find the exact service provider for the requested service. The Jini protocol is used to achieve the aim of this project because it is easily available and modular. The method used for programming in Jini is in Java programming language. To ease implementation, this project used Jini technology version 2.1 starter kit. There are three services generated using three different service providers. They are a public taxi service, an ambulance service, and PLUS services with three other PC as the service providers. The client and service providers are located in three different distances to test the system. From the project analysis, Jini protocol can be implemented to the real world service application. However, the delay performance of the network depending on the network traffic and the distance between the client and service providers.

ABSTRAK

Pada masa kini, ramai pengguna bergantung kepada talain intenet untuk mencari dan menggunakan sesuatu peralatan atau perkhidmatan yang diperlukan. Walau bagaimanapun, ianya boleh menjadi lebih kompleks apabila terdapat beratus-ratus penganjur yang menyediakan peralatan atau perkhidmatan tersebut. Oleh itu, projek ini dijalankan untuk menghasilkan sebuah jaringan protokol yang dapat mengesan kehadiran sesuatu peralatan atau perkhidmatan dalam jaringan komputer selagi ia menggunakan protokol yang sama. Protokol ini secara automatiknya berkebolehan untuk membenarkan pengguna mendapat dan mengguna peralatan atau perkhidmatan yang diperlukan secara tepat. Dalam projek ini, protokol Jini digunakan berbanding protokol yang lain. Ini disebabkan sumber penggunaan protokol Jini adalah terbuka. Metodologi yang digunakan kebanyakannya berasaskan bahasa pengaturcaraan Java. Namun begitu, terdapat pilihan untuk menggunakan kit permulaan dengan teknologi Jini versi 2.1 bagi memudahkan implementasi Jini. Terdapat tiga perkhidmatan yang dibina dalam projek ini iaitu perkhidmatan teksi, ambulan dan PLUS dengan tiga penganjurnya pada komputer yang berbeza. Bagi menguji system ini, satu pengguna diletakkan pada jarak yang berbeza-beza dengan tiga penganjur tersebut. Hasil daripada eksperimen didapati protokol Jini boleh diaplikasi kepada persekitaran untuk mencari serta mengguna perkhidmatan. Walau bagaimanapun, kelewatan pada sistem ini bergantung kepada trafik dalam jaringan telekomunikasi serta jarak di antara pengguna dan penganjur perkhidmatan.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligent
DA	-	Directory Agent
DHCP	-	Dynamic Host Configuration Protocol
ERI	-	Extensible Remote Invocation
GSS	-	Generic Security Service
HTTP	-	Hypertext Transfer Protocol
IETF	-	Internet Engineering Task Force
IP	-	Internet Protocol
JAR	-	Java Archive (Default Java File)
JERI	-	Jini Extensible Remote Invocation
JRMP	-	Java Remote Method Protocol
JSSE	-	Java Secure Socket Extension
JVM	-	Java Virtual Machine
LAN	-	Local Area Network
LUS	-	Lookup Service
MF	-	Manifest (Default Java File)
PLUS	-	Projek Lebuhraya Utara Selatan
RMI	-	Remote Interface
SA	-	Service Agent
SDK	-	Software Development Kit
SDP	-	Service Discovery Protocol
SLM	-	Salutation Manager
SLP	-	Service Location Protocol
SP1	-	Service Provider 1

SP2	-	Service Provider 2
SP3	-	Service Provider 3
SSDP	-	Simple Service Discovery Protocol
SSL	-	Secure Socket Layer
TCP/IP	-	Transmission Control Protocol/Internet Protocol
UA	-	User Agent
UDP	-	User Datagram Protocol
UPnP	-	Universal Plug and Play
XML	-	Extensible Markup Language (Default Java File)

CHAPTER 1

INTRODUCTION

1.1 Introduction

Service Discovery Protocols (SDPs) are network protocols which allow automatic detection of devices and services offered by these devices on a computer network [1]. In other words, service discovery is an action of finding a service provider for the requested service. The user may access and use the service when the location of the demanded service (typically the address of the service provider) is retrieved. The service discovery protocols also provide features to spontaneously lookup services with a low communication overhead.

There are some elements participate in the service discovery protocol. Each service discovery protocol consists of at least two participating elements which are client and server. The client also called as a user is an entity that interested in finding and using a service, while server is the entity that offers the service. However, it is common to find third participating entity within service discovery protocol. This is due to protocol may use service repositories to facilitate service mappings. The third entity is named as directory which act as a node in the network that hosts partially or

entirely the service description information [1]. The directory also known as broker, central, resolver, and server depends on situation.

The most popular service discovery protocols that addressing the service discovery are Service Location Protocol (SLP), Jini, Bluetooth Service Discovery Protocol (Bluetooth SDP), Salutation, and Universal Plug and Play (UPnP).

1.1.1 Service Location Protocol (SLP)

The Service Location Protocol (SLP) is developed by the IETF (Internet Engineering Task Force) working group. The aim of SLP is to be a vendor-independent standard. SLP is designed for TCP/IP networks and is scalable up to large enterprise networks. The SLP architecture consists of three main components which are User Agents (UA), Service Agents (SA), and Directory Agents (DA). The UA is on behalf of the client or user that perform service discovery. The SA on behalf of service advertises the location and characteristics of services, while DA acts as data storage. It collects service addresses and information received from SA in their database and respond to the service requestor which is UA.

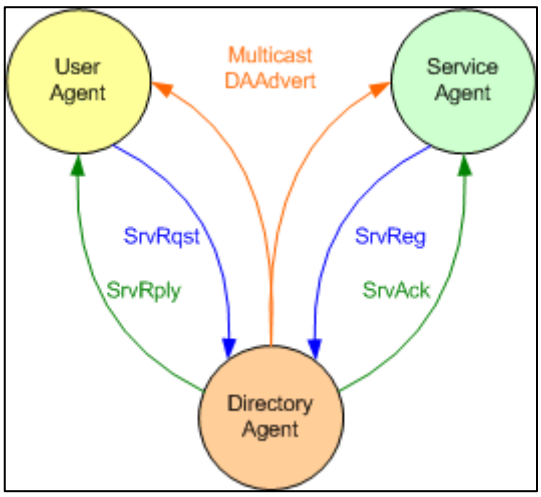

Figure 1.1 : Service Location Protocol with Directory Agent

Figure 1.1 shows the interactions between three agents (user agent, service agent, and directory agent). It starts whenever a new service connects to the networks, the SA contacts the DA to advertise its existence. This process named as *Service Registration* process. After sometimes, whenever a user demand and request for a certain service, the UA make queries for the available services in the network from the DA. The process of querying for a service is called as *Service Request*. When the user receives the address and characteristics of the demanded service, it may finally utilize the service.

However, a client of DA (UA or SA) must discover the existence of the DA before they are able to contact the DA. There are three methods for DA discovery which are static, active, and passive discovery. In static discovery, SLP agents obtain the address of DAs through the Dynamic Host Configuration Protocol (DHCP). DHCP servers distribute the addresses of DAs to hosts that request them. In active discovery, UAs and SAs send service requests to the SLP multicast group address (239.255.255.253). A DA listening on this address will eventually receive a service request and directly respond to the requesting agent via unicast. With the passive discovery, DAs are periodically sending out multicast advertisements for their services. UAs and SAs then learn the DA address from the received advertisements and are able to contact the DA themselves via unicast [2].

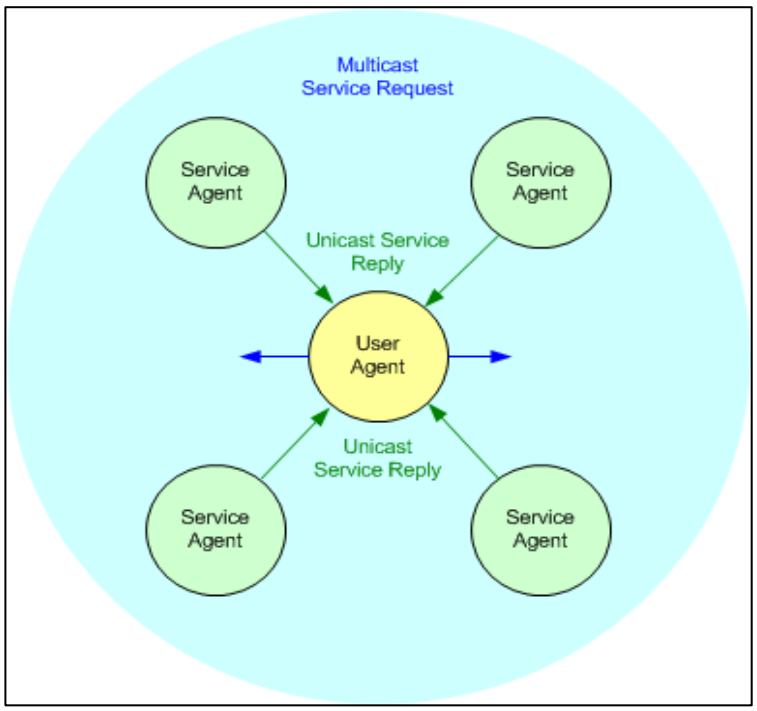

Figure 1.2 : Service Location Protocol without Directory Agent

In fact, the usage of DA is not mandatory. It is used especially in a large network with many services. So that, it allows services to be categorized into different groups or scopes. Therefore, for a small network like home or car network, it is more effective to deploy SLP without using the DA which the architecture is shown in Figure 1.2. In this case, UA is repeatedly send out their *Service Request* to the SLP multicast address. The SAs listen for the multicast request and advertise the requested service to the UA. They will send unicast response to the UA. Furthermore, SAs will multicast an announcement of their existence periodically to the UA, so that the UA can learn about the existence of new services.

1.1.2 Jini

Jini technology is an extension of the programming language Java. It has been developed by Sun Microsystems. The concept of Jini is how devices connect with each other in order to form a simple ad hoc network and how these devices provide services to other devices in the network [2]. Each Jini device is assumed to have a Java Virtual Machine (JVM) running on it. Without JVM, Jini technology cannot be implemented in the device.

The Jini architecture principle is similar to the SLP. The process of devices and applications register with a Jini network is called as *Discovery and Join* process. In order to join a Jini network, a device or application places itself into the *Lookup Table* on a lookup server, which is a database for all services on the network. This concept is similar to the DA in SLP.

1.1.3 Bluetooth Service Discovery Protocol (SDP)

Bluetooth is a new short range wireless transmission technology. The Bluetooth protocol stack contains the Service Discovery Protocol (SDP), which is used to locate services provided by or available via a Bluetooth device. It is based on Piano platform by Motorola and has been modified to suit the dynamic nature of ad hoc communications [2].

With Bluetooth, it addresses service discovery specifically for this environment and focuses on discovering services. It supports the inquiries such as searching for services by service attributes, searching for services by service type, and service browsing without a prior knowledge of the service characteristic. The

SDP does not include functionality for accessing services. Once services are discovered with SDP, the services can be selected, accessed, and used by mechanisms out of the scope of the SDP. It means, there is no need service mapping within the Bluetooth SDP.

1.1.4 Salutation

Salutation is another protocol approach to the service discovery. The Salutation has been developed by an open industry consortium which called as the Salutation Consortium [2]. In the Salutation architecture, it consists of Salutation Managers (SLMs). The SLM has functionality of service brokers. Services will register their capabilities with the SLM, while the clients will query the SLM when they need a service. After discovering a desired service, clients are able to request the utilization of the service through the SLM.

1.1.5 Universal Plug and Play (UPnP)

Universal Plug and Play (UPnP) is being developed by an industry consortium which has been found and is lead by Microsoft. It also can be said that it extends the Microsoft's Plug and Play technology to the case where devices are reachable through a TCP/IP network [2]. The UPnP usage is proposed for small office or home computer networks, where it enables peer-to-peer mechanisms for auto-configuration of devices, service discovery, and also for controlling of services. In UPnP, there is no central service register, such as the DA in SLP or the lookup table in Jini. The UPnP uses the Simple Service Discovery Protocol (SSDP) to

discover services. SSDP uses Hypertext Transfer Protocol (HTTP) over User Datagram Protocol (UDP) and is designed for IP network usage.

1.2 Problem Statement

Nowadays, it is quite difficult to find and use any demanded or desired service in an area which is not familiar to the user. Without having any information about server or service provider, it will be more complicated. Usually people need to call the telephone operator to get the service provider's contact number. Others spend their time surfing the Internet in order to get information of the desired service through a numbers of service providers. More time need to be spent for this purpose.

In a fast changing environment, where services appear and disappear in an unexpected way, it is crucial to employ a fast and efficient service selection process, which will also not distract the user. By using SDP, it enables users to discover and use proximity services effortlessly in the fastest and most efficient way. Nevertheless, in service discovery, devices may also automatically discover network services including their properties and the services that may advertise their existence in a dynamic way. This will make the SDP is the best way to be used not only the client, but even it eases the service providers to advertise their services.

1.3 Objectives

The objectives of this project are as the follows:

i. To implement the Jini technology in a network for the Service Discovery Protocol purpose.
ii. To create services and service providers that can be used in Jini technology.
iii. To analyze the performance of the Jini technology in a network.

1.4 Scope of Work

There are few scopes and limitations in doing this project. The first one is to use Jini technology as the service discovery protocol rather than others. For the implementation, Jini technology is applied during testing process on command prompt and on the Jini technology starter kit.

The second limitation of this project is focusing on three entities of the SDP. They are will be client, server, and service provider. The client and service provider need to have an embedded Jini. In this project, it has one client computer and three computers of service providers with three difference service offered.

The last one is to check the network performance during Jini technology implementation. However, the limitation is to check the network delay by using the Wireshark Network Protocol Analyzer.

1.5 Project Report Outline

This project report consists of five chapters. The first chapter (chapter 1) provides the introduction to this project and also provides the problem statement, objectives, and scope of the project.

Chapter 2 provides pass few years projects which have been done on SDP area. Background of the project including Jini SDP architecture and the way it works are also explained in this chapter.

Chapter 3 details on the methodology of SDP using Jini technology. This chapter explains on how to create services using command prompt and how to test the test bed. Delay of the network will be determined through Wireshark.

Chapter 4 describes the results and discussion of the project experiments. The results obtained from the Jini Technology starter kit, windows's command prompt, and the Wireshark software. Then, the network performance is calculated as the highlight of the project.

Chapter 5 concludes what have been done to complete the project and few numbers of future works are suggested in order to improve the project.

REFERENCES

[1] Marin-Perianu, R., Hartel, P., and Scholten, H. (2005). *A Classification of Service Discovery Protocols*. Technical Report TR-CTIT-05-25, Centre for Telematics and Information Technology, University of Twente, The Netherlands.

[2] Bettstetter, C. and Renner, C. (2000). A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol. *Proceeding of the Open European Summer School (EUNICE)*. September 13-15, 2000.Twente Netherlands.

[3] TTM47AC Laboratory. (September 1, 2003). Institute of Telematik, Norwegian University of Science and Technology.
http://www.item.ntnu.no/fag/ttm47ac/Information/Jini_Introduction.pdf

[4] Newmarch, J. *UPnP Services and Jini Clients*. (2005). Information Systems: New Generations (ISNG), Las Vegas.

[5] Chen, H., Chakraborty, D., Xu, Liang., Joshi, A., and Finin, T. Service Discovery in the Future Electronic Market. *Seventeenth National Conference on Artificial Intelligence, Eleventh Innovative Applications of AI Conference*. Austin. 2000.

[6] Lee, C. and Helal, S. (2002) *International Journal of Computer Research*. Computer and Information Science and Engineering Dept., University of Florida. 11(1), 1-12.

[7] Zhou Xin-lian and Wu Min, Service Discovery Protocol in Wireless Sensor Networks (2006). *Proceedings of the Second International Conference on Semantics, Knowledge, and Grid (SKG'06)*. 1-3 November. Guilin, Guangxi, China: IEEE Computer Society, 101.