Comparative Study of Service Discovery Protocols for ad-hoc Networks

G. E. RIZOS , D. C. VASILIADIS
University of Peloponnese
Department of Computer Science and Technology
GR-221 00 Tripolis
GREECE

E. STERGIOU
A.T.E.I. of Epirus
Department of Teleinformatic and Management
GR-471 00 Arta
GREECE

S. V. MARGARITI
University of Ioannina
Computer Science Department
GR-451 10 Ioannina
GREECE

Abstract:

In the last years there is the growing trend in electronic devices to aggregate as mush as possible services trying to provide to its user all the possible necessary functions. For example mobile phones acting also as MP3-players.

With technologies like WLAN and Bluetooth that provide wireless connectivity, and even more with the ability of ad-hoc discovery and networking between electronic appliances, an alternative distributed computational model could be exploited where a device will not necessarily implement all its services but discover and use the services provided from other devices with a simple and flexible manner, formulating small "personal area" networks tailored to the needs of each user.

To accomplish this distributed model there is the need of devices to discover ad-hoc the available surrounding services/devices, so there is the need of a discovery mechanism that will provide this functionality with respect to speed of discovery, high degree of discoverability success and low energy consumption.
In this paper we studied the design and implementation of the Service Discovery Protocols to see their abilities and their problems in personal area networking.

Key-Words: Service Discovery Protocols, Bluetooth, Jini, UPnP, Salutation, ad-hoc

1 Introduction

As service in the problem of discovery of services in networks of personal region is fixed each application which allocates a total of well certain contacts via which they can be executed certain calculations or certain operations on behalf of certain other applications-customer. For the communication between services and customers is used some means of communication which varies, and usually is some type of network (for example: Internet, Bluetooth) for which is been disposed some protocol via which is carried out the communication (for example: Internet Protocol, Bluetooth Protocol Specification).

Up to the beginning of decade the '90, the various constructors of appliances, had observed that the various enterprises extended the networks adding new appliances and that at the addition of this new appliances were presented various problems. The problems were primarily problems of compatibility. Each time where it wanted an enterprise it connects the new appliance in the already existing network, should have added software in each node of network so that exists always access in the new equipment. Was particularly improbable the system he was homogeneous, those is to say appliances of network, new or old, were products of same constructor.

So that big constructors (IBM and Hewlett-Packard, as well as Canon, Hitachi, Ricoh, and other big constructors of offices equipment) direct the efforts they develop protocols of discoveries and afterwards to trade these, mainly as department of bigger programs.

Their common objective is they render capable the various appliances (gadgets) as faxes, computers, and the all remainder equipment it is connected in networks and they communicate and they collaborate with them.

Thus in 1995 the certain basic companies of space (mainly USA and Japan) shaped a new consortium - **Salutation**

Consortium so that is found a single solution which would resolve these problems.

The members of consortium realised that, the solution had two parts:

- It should exists a uniform way of description of possibilities that have the various digital appliances, and
- It should exists a single common method of confrontation of information.

As an example a printer will be supposed to describe the total of possibilities in anyone of suitably equipped computer or other element of network with which it is to collaborate.

Fortunately, the Industrial teams as IETF (IETF - Internet Engineering Task Force) - worked already in the first objective (uniform way of description) - developing standards data (data) on the different types of equipment. For example, the IETF models, RFC (RFC - Request-for-Comment), provide a model of description of possibilities of various appliances. The printers as an example, accept different sizes of paper from different entries, they can they use different colours; use some type of classification, and various logos. RFC 1759 (1995) describes how these information, with the situation of printers (as "in use" or "warm-up") should be coded and registered in a file of called table of printers MIB (Management Information Base).

2 Service Discovery Protocols

Most important Service Discovery Protocols that are presented in the international bibliography are:

2.1 SLP

SLP [1] It is a protocol that was developed by the IEFT for services discovery above the IP networks. As Bluetooth SDP [2, 3, 7] supports operations of services discovery with base the type of sought service or various attributes and characteristics of this service and still it supports browsing services , that is to say mass search for the all available services, using IP multicasting.

For the discovery of services it uses the methodologies:

- (active/pull)
- (passive/push)

His mainer architectural sub systems are:

- (UA – User Agent)
- (SA – Service Agent)
- (DA – Directory Agent)

And the mainer operations appear in figure 1 below:

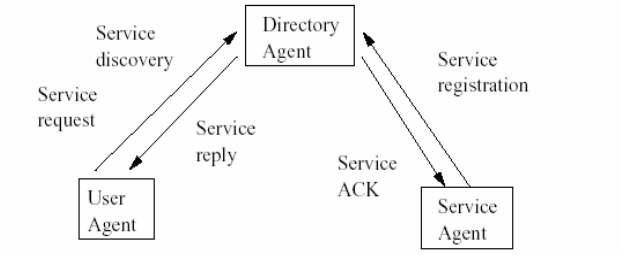

Fig. 1 SL Protocol

2.2 Jini

The system Jini [4] is an extension of language programming Java and was developed by the company Sun Microsystems. It is not protocol as SLP, but it constitutes concrete platform, in Java. In each appliance Jini it should essentially runs a Java Virtual Machine (JVM) as the communication between the appliances becomes using technology RMI (Remote Method Invocation)

His mainer architectural sub systems are:

- Clients
- Server
- Lookup Servers

The appliances are registered in a Jini network using a process that is named Discovery and Join. Jini fixes a protocol which becomes the discovery of Tables of Recovery in a such network SLP, with technique multicast. The conjunction in a Jini network becomes with the registration of appliance-Server in Tables of Recovery in this server that has shouldered this operation and which is the corresponding action of registration in DA on SLP. Afterwards, in the tables of recovery they can be addressed other appliances-customers so that discover information for the services that seek.

The operations appear schematically in figure 2 below:

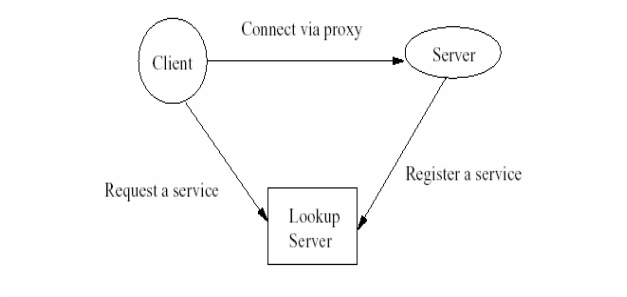

Fig. 2 Jini Protocol

2.3 UPnP

The protocol UpnP (Universal Plug and Play) [5, 6] is an open protocol that was developed from a forum of companies which leads the Microsoft Company, who promotes and extends the already widespread technology of Plug and Play.

For his creation it was supported in pre-existing protocols and technologies as HTTP, XML, SOAP and the protocol UPnP substantially it is the cohesive web between these technologies, with certain extensions where this is essential. It is capable to function above any appliance and platform as it is independent of programming language. It is constituted by various sub systems and the Simple Service Discovery Protocol (SSDP), as the name implies, defines how network services can be discovered on the network.

SSDP is built on HTTPU and HTTPMU and defines methods both for a control point to locate resources of interest on the network, and for devices to announce their availability on the network. By defining the use of both search requests and presence announcements, SSDP eliminates the overhead that would be necessary if only one of these mechanisms is used. As a result, every control point on the network has complete information on network state while keeping network traffic low.

Both control points and devices use SSDP. An UPnP control point, upon booting up, can send an SSDP search request (over HTTPMU), to discover devices and services that are available on the network. The control point can refine the search to find only devices of a particular type (such as a VCR), particular services (such as devices with clock services) or even a particular device.

UPnP devices listen to the multicast port. Upon receiving a search request, the device examines the search criteria to determine if they match. If a match is found, a unicast SSDP (over HTTPU) response is sent to the control point.

Similarly, a device, upon being plugged into the network, will send out multiple SSDP presence announcements advertising the services it supports.

Both presence announcements and unicast device response messages contain a pointer to the location of the device description document, which has information on the set of properties and services supported by the device.

In addition to the discovery capabilities provided, SSDP also provides a way for a device and associated service(s) to gracefully leave the network (bye-bye notification) and includes cache timeouts to purge stale information for self healing.

These protocols appear schematically in figure 3 below:

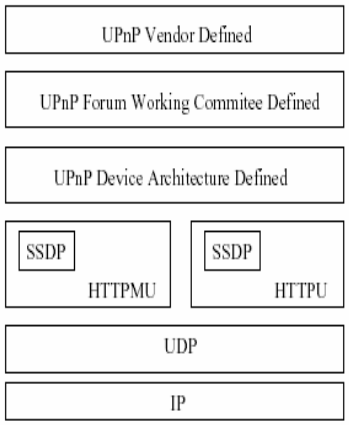

Fig. 3 Protocols in UPnP for discovery of services

2.4 Salutation

The **Salutation Architecture** was created to solve the problems of ***service discovery and utilization*** among a broad set of appliances and equipment and in an environment of widespread connectivity and mobility.

The architecture provides a standard method for applications, services and devices to describe and to advertise their capabilities to other applications, services and devices and to find out their capabilities. The architecture also

enables applications, services and devices to search other applications, services or devices for a particular capability, and to request and establish interoperable sessions with them to utilize their capabilities.

Given the diverse nature of target appliances and equipment in an environment of widespread connectivity, *the Salutation architecture is processor, operating system, and communication protocol independent,* and allows for scalable implementations, even in very low-price devices.

The Salutation Architecture defines an entity called the **Salutation Manager (SLM)** that functions as a service broker for applications, services and devices called a Networked Entity. The Salutation Manager allows Networked Entities to discover and utilize the capabilities of other Networked Entities.

A Networked Entity may be a service provider, called a **Service**. The concept of a 'service' is broken down into a collection of **Functional Units (FU)**, each unit representing some essential feature (e.g. Fax, Print, Scan or even subfeatures like Rasterize). The Service registers its capability with a Salutation Manager.

A Networked Entity may be a service user, called a **Client**. The Client discovers Services and requests to use them through a Salutation Manager. A Networked Entity may serve as either a Client or a Service, or both.

The Salutation Manager provides a transport-independent interface, called the **Salutation Manager Application Program Interface (SLM-API**), to Services and Clients. The architecture defines an abstract procedural SLM-API.

The Salutation Manager communicates with other Salutation Managers to perform its role as a service broker, using the **Salutation Manager Protocol**. The Salutation Manager Protocol uses Remote Procedure Call (Sun Microsystems' Open Networking Computing Remote Procedure Call version 2), that is, a Salutation Manager makes a Remote Procedure Call to another Salutation Manager, which returns a Remote Procedure Call reply.

The Salutation Manager provides also a transport-independent interface, called the **Salutation Manager Transport Interface (SLM-TI**), to transport-dependent entities, called **Transport Managers**. The Transport Manager is introduced to make the Salutation Manager transport-independent. The Salutation Manager and Transport Manager(s) together perform the service broker role.

This architecture appears schematically in figure 4 below:

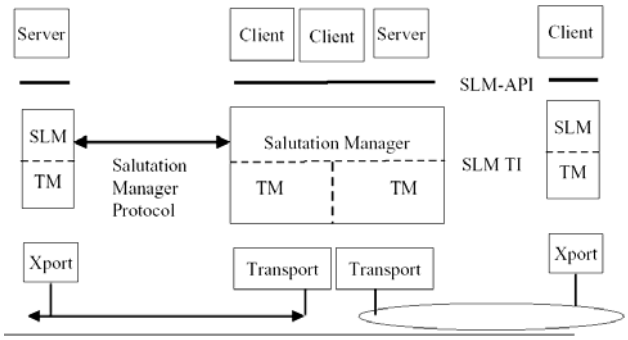

Fig. 4 Model of Salutation Manager

3 Discovery Strategies

In the paper [8] you can see an analysis and modelising of different strategies than can be used for the discovery of services in dynamic, organized networks. There are suggested two different strategies that were used and measured so that it could be countered their performance and their efficiency.

- ***Publishing to all, question to all*** (post-to-all, query-to-all). In this strategy the services publish to all nods their availability and the clients ask the nodes for the seeking questions. As it is known this strategy is demanding and not efficient in using the characteristics of the topology of the network etc so that it can succeed adaptation and rapidity.
- ***Incremental publishing / question*** (incremental post / query). In this strategy, the publishing/question for the existence of services is confronted step by step to the nearby group of nodes and then to the others. This however depends on the occasions. We can distinguish two occasions.
  - *Incremental publishing, question to all.* In this occasion the publishing of services is not over

at once but gradually. The questions are sent to all nodes.

- *Publishing to all, occasional question.* In this occasion the services are being published to all nodes. However the questions are made gradually.

These strategies have been measured when the facts were:

- The discovering success is embodied by the mathematical type, number of services which have been discovered to the number of experiments which have been made.
- The discovering fastness, taking as measure the delay before we can discover the specific service.
- The discovering cost, taking as measure the number of messages that every strategy demands and the measures showed the repercussive results.

3.1 Compare Discovery Strategies

The first strategy has always the biggest cost (or about the same) with the other strategies. So this is used as a critical point to the performance of the others. On the other hand, the cost is multiple to the lackage of nods or paths of communication for successive discovery of the wanted service. The best performance relatively to the cost is the third strategy, where we have occasional questions to the nodes. As bigger is the number of the services so it maintains the success and the discovering fastness. However, the cost is diminished up to five times (for the measures which have been made). The discovering success does not show to be affected very much and depends mainly on the number of services. Of course the discovering success is getting bigger in association with the coherent services. As a result, from this paper we can conclude that with questions to qualified nodes in this first stage, we have much better results relatively to the cost without having important repercussions (most times not at all) to the success and discovery fastness.

4 Conclusion

The basic difference between Jini and SLP [9] is that apart from information on the services and the addresses for communication between them, Jini provides the possibility of be contained in the Lookup Table a program in Java language for this service. This program can be transported automatically (without asks him/regulates expressly the customer) in JVM the customer of service and functions as driver ensuring the connection and the communication with the service, without it needs certain other work than the side of user and increasing his use.

UPnP functions only above IP networks [9] and his basic difference with SLP and Jini is that does not exist a certain central entity in which register itself the services as Directory Agent in SLP or Lookup Server in Jini, therefore are more useful and functional for ad hoc networks.

However, for facility of Salutation of (greeting) in the protocols of communications, Salutation Consortium and Bluetooth SIG they resolved the from each other incompatibility in one year. In few months they solved a specification Bluetooth for the incorporation of “greeting” Salutation in the system Bluetooth.

This specification allows in the designers of Bluetooth to apply the “greeting” in two ways:

- or by the replacement of Salutation Manager with the software of protocol of discoveries of service Bluetooth (Bluetooth SDP)
- or with the addition of translator API in each appliance, so that, the directives of “greeting” in the code Bluetooth to be translate.

The “greeting” (Salutation) continues functioning with Bluetooth SIG and it is officially recognized as part of protocol Bluetooth.

References:

[1] Erik Guttman, Charles Perkins, and Michael Day. *Service Location Protocol: Version 2. Internet RFC 2608, 1999.*

[2] Eugene A. Gryazin. Service discovery in Bluetooth. *Available at: http://citeseer.nj.nec.com/392311.html.*

[3] Christian Schwingenschogl and Anton Heigl. Development of a service discovery architecture for the Bluetooth radio system. *In Proceedings EUNICE 2000,Sixth EUNICE Open European Summer School, September 2000.*

[4] Sun Microsystems. Technical white paper: Jini architectural overview, 1999. *Available at: http://www.sun.com/jini/.*

[5] Universal Plug & Play Forum. Universal Plug and Play device architecture, 2000. *http://www.upnp.org.*

[6] Ravi Gauba and Nigel Davies. Optimizing UPnP network performance for ubiquitous computing environments.

[7] Sedov I., Haase M., Preuss S., Cap C., and Timmermann D. Time and energy efficient Service discovery in Bluetooth. *In Proceedings of the 57th IEEE Vehicular Technology Conference , 2003.* Available at: http://citeseer.nj.nec.com/570807.html.

[8] Michel Barbeau and E. Kranakis. Modeling and performance analysis of service discovery strategies in ad hoc networks (extended abstract). In *Proceedings of 2003 International Conference on Wireless networks (ICWN),* June 2003. Available at: http://citeseer.nj.nec.com/587240.html.

[9] Christian Bettstetter and Christoph Renner. A comparison of service discovery protocols and implementation of the Service Location Protocol. *In Proceedings of the EUNICE 2000, Sixth EUNICE Open European Summer School,* September 2000.