Implementing Service Discovery Protocols in Home Network Systems

Rabab J. Mohsin
Ass. Lecturer,College of Eng.,
AL-Mustansiriya University

Abstract:

Home networking systems are gaining increasing research attention in the recent period due to the huge efforts implied by major home appliance companies in order to equip future home appliances with networking services and capabilities. Service discovery protocols being an essential backbone in home network system development receive the majority of research efforts to provide robust and efficient home networks.

In this paper the most popular service discovery protocols are reviewed in a comparative study discussing the advantages and disadvantages of each of them over the others.

Keywords: HNS, Jini, UPnP, HAVi

1. Introduction

Nowadays, devices in a home environment are often equipped with general purpose network connections. There is an increasingly strong requirement that these devices cooperate in an autonomous fashion by using the services (i.e., functionality) they find on the network. To that end devices need to be able to find out about these services, to find the devices that supply them to employ the services and, conversely, they must be able to advertise their own services. In the context of home networking several standard technologies have been proposed for this purpose [1].

A variety of networks and protocols have been proposed for respective application domains. For example, various protocols such as Jini and UPnP (Universal Plug and Play) have been proposed for connecting various appliances without their configuration before using them. HAVi has been proposed for connecting various audio and video home appliances. On the other hand, various network systems such as ATM (Asynchronous Transfer Mode) networks, IEEE 1394, Bluetooth, and VIA have been developed to connect a variety of appliances. Also, in the research community, PEN, networked surface, and CLAN have been proposed for connecting future advanced appliances. However, various useful characteristics of these underlying protocols and networks are hidden from applications if an IP

layer is inserted on the networks. For example, the plug and play functionality provided by Bluetooth and the network bandwidth reservation functionality provided by ATM are not usually available on the top of the IP layer. Also, some appliances do not support the IP layer for connecting to other appliances and services. Moreover, each appliance may assume different control protocols and data formats to communicate each other [2].

1. Service Discovery Protocols

Service discovery protocols (SDP) are network protocols which allow automatic detection of devices and services offered by these devices on a computer network. Service discovery requires a common language to allow software agents to make use of one another's services without the need for continuous user intervention [3]. Below are the major service discovery protocols applied:

1.1 Jini

Jini is a service discovery technology based on Java, developed by Sun Microsystems. Because of the platform-independent nature of Java, Jini can rely on mobile code for interaction between clients and services. Lookup services provide catalogs of available services to clients in a Jini network. On initialization, Jini services register their availability by uploading proxy objects to one or more of these lookup services. The proxy objects are essentially "device drivers" written in Java—they allow the client to control the service. Protocols are included for connecting to lookup services with known locations and for dynamically discovering lookup services within multicast range. Once a client has contacted a lookup service, it can search for interesting services and then download the corresponding service proxy objects. In Jini, searching is based on the type of the proxy object generally specified using a Java interface and on sets of descriptive attributes [4].

The purpose of the Jini architecture is to federate groups of devices and software components into a single, dynamic distributed system. Jini systems provide mechanisms for service construction, lookup, communication, and use in a distributed system. Examples of services include devices such as printers, displays, or disks; software such as applications or utilities; information such as database and files; and users of the system. The heart of the Jini system is a trio of protocols called discovery, join, and lookup. A pair of these protocols -discovery/join- occurs when a device is plugged in. Discovery occurs when a service is looking for a lookup service with which to register. Join occurs when a service has located a lookup service and wishes to join it. Lookup occurs when a client or user needs to locate and invoke a service described by its

interface type (written in the Java programming language) and possibly, other attributes.

The following steps show what interactions are needed among a client, a service provider, and a lookup service for a service to be used by the client in a Jini community:

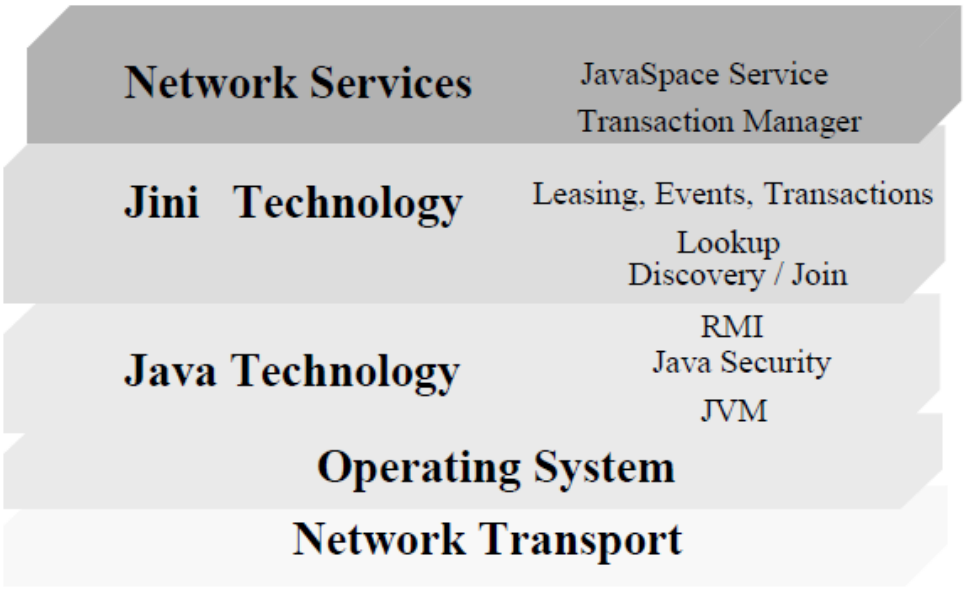
1. Service provider locates a lookup service by multicasting a request on the local network or a remote lookup service known to it in priori.

2. A service provider registers a service object and its service attributes with the lookup service. This service object contains Java programming language interface for the service, including the methods that users and applications will invoke to execute the service, along with any other descriptive attributes.

3. A client requests a service by Java type and, perhaps, other service attributes. A copy of the service object is moved to the client and used by the client to talk to the service.

4. Then, client interacts directly with the service provider via the service object.

Jini connection technology consists of an infrastructure and a programming model which address the fundamental issues of how devices connect with each other to form an impromptu community. Based on Java technology as shown in (Fig.1), Jini technology uses Java Remote Method Invocation protocols to move code around the network. Network services run on top of the Jini software architecture [5].



(Fig.1) Architecture of Jini Connection Technology

2.2 Universal Plug and Play (UPnP)

Universal Plug and Play is a set of protocols for service discovery under development by the Universal Plug and Play Forum, an industry consortium led by Microsoft. UPnP standardizes the protocols spoken between clients (called control points in the UPnP lingo) and services rather than relying on mobile code. Device and service descriptions are coded in XML, and a number of protocols for local auto configuration, discovery, advertisement, client/service interaction, and eventing. Auto-IP, SSDP (Simple Service Discovery Protocol), SOAP (Simple Object Access Protocol), GENA (General Event Notification Architecture) are included in the specification. These protocols tend to be based loosely on existing standards (e.g., HTTP).

The UPnP specification addresses six areas. The first (numbered 0) is Addressing, and is related to device initialization—in order to interact with other UPnP entities, a device or control point must have an IP address. IP addresses will generally be provided by a DHCP server or configured statically, but for environments with little infrastructure (e.g., a home LAN), UPnP uses a protocol named Auto-IP to automatically generate non-routable IP addresses. The second area is Discovery, which allows control points to discover UPnP devices offering services of interest. Discovery also covers device advertisement, used to announce the availability of devices as they enter the network. The discovery phase provides control points with existential information, but little additional information about the discovered devices. Discovery is the concern of SSDP. Description, or the ability to inquire about device specifics (e.g., manufacturer, serial numbers, specific services offered, etc.) is the third phase. Using a URL obtained in the Discovery phase, a control point can download a document called the device description document, which fully describes a device of interest. This document is platform-neutral, expressed entirely in XML.

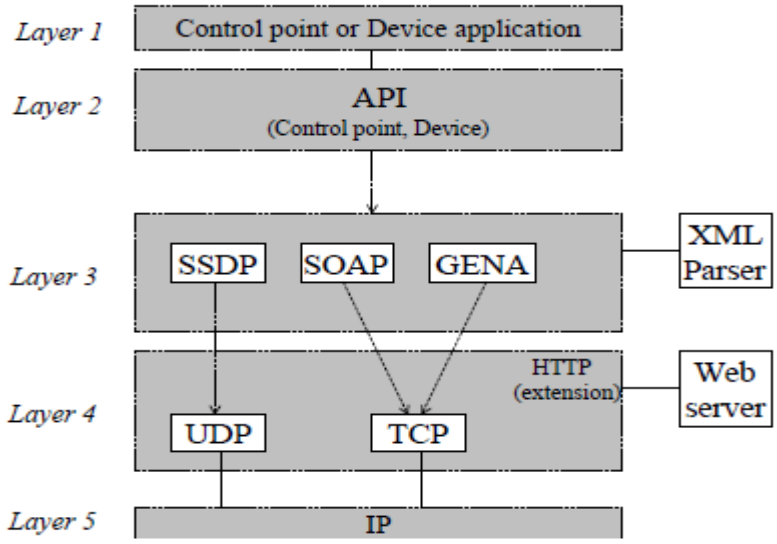


Fig. 2 architectural model of the UPnP system [1]

If a control point determines, after examining a device's description document, that it wishes to interact with services provided by a device then Control is used to send the device commands and receive results. All Control interactions are via a protocol based on HTTP, called SOAP (Simple Object Access Protocol), which runs over TCP. Commands sent via SOAP are addressed to control URLs for the services provided by a device— these control URLs are specified in the device's description document, which is obtained during the Description phase. The API for a service is also expressed in XML.

To be informed of important state changes, UPnP control points subscribe to the event services offered by a device. This mechanism makes up the Eventing portion of the UPnP specification and is handled by the GENA (General Event Notification Architecture) protocol. Eventing consists of notifications of state variable changes. Finally, the Presentation aspect of UPnP allows devices to define a presentation URL, which is the location of an HTML document which provides a graphical interface to the device (a "virtual" remote control).

Unlike Jini and SLP, UPnP does not support service directories communication between devices and clients is always direct. Thus UPnP is aimed at smaller environments where the benefits of directories are reduced due to the small number of devices typically found in the network [4].

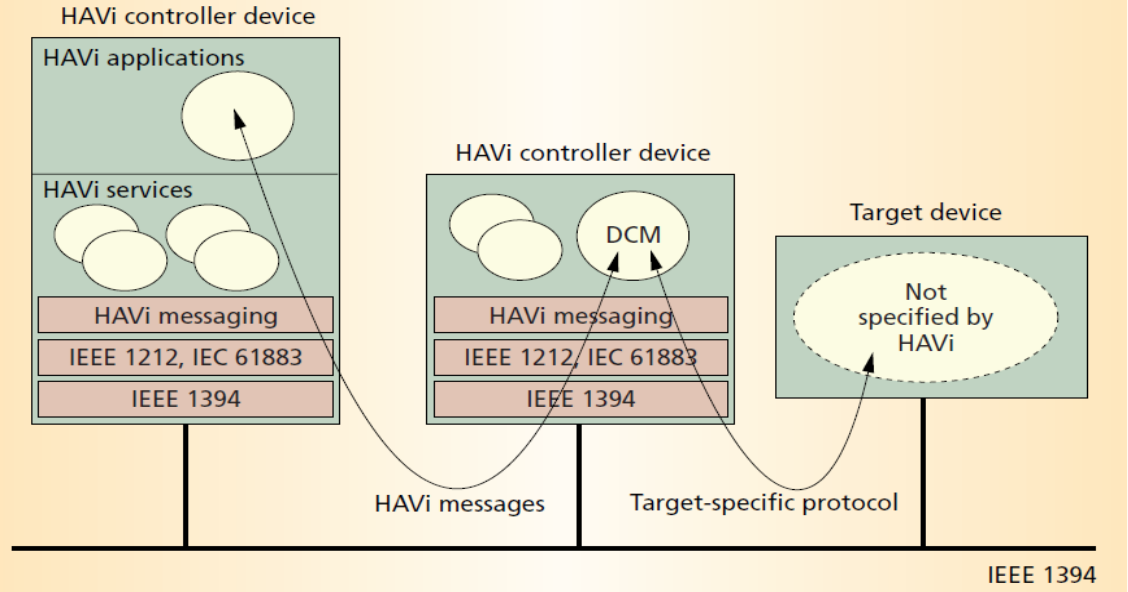
2.3 Home Audio/Video Interoperability (HAVi)

HAVi provides a peer-to-peer environment where any device can detect, query, and control any other device, and groups of devices can cooperate to provide enhanced services. However, given the wide range of functionality consumer electronics devices provide, as well as the need to accommodate

existing non-HAVi compliant devices, the HAVi architecture allows many variations in device functionality and, hence, cost [6].

2.3.1 Device control modules and functional component modules

Before discussing HAVi device categories it is useful to introduce the distinction between controller devices and controlled devices. A HAVi controller acts as a host for a controlled device by running a software proxy called a device control module (DCM) for the controlled device. A controller device may itself be controlled by other devices and so may also be represented by a DCM. A set of HAVi-specified messages provide communication between applications and the DCM. The vendor of the controlled device specifies communication between it and the DCM. Figure 3 shows a communication example.



(Fig. 3) Communication in a HAVi network

The DCM exposes a well-defined interface accessible through HAVi APIs. These APIs provide the only access point for HAVi applications to control the device, although non-HAVi applications are free to use private APIs.

A DCM aggregates smaller units called functional component modules (FCMs), which allow application control of related device-function groups. For example, a television set might contain a tuner FCM, a display FCM, and perhaps a clock FCM. HAVi 1.0 specifies 10 FCM APIs: tuner, VCR, clock, camera, AV disc, amplifier, display, AV display, modem, and Web proxy.

Central to the HAVi architecture, DCMs provide flexibility in accommodating new devices and features. We can distinguish DCMs in several ways [6].

- ***Embedded and uploaded.*** An embedded DCM forms part of a controller's resident software, whereas an uploaded DCM originates from an external source and is dynamically added to the controller's software.
- ***Native and bytecode***. A native DCM, implemented for a specific platform, may include machine code for a specific processor or may access platform-specific APIs. A portable DCM is implemented in Java bytecode.
- ***Standard and proprietary***. A standard DCM provides the HAVi-specified APIs. It offers basic functionality but can control a wide range of devices. A proprietary DCM provides vendor-specific APIs in addition to the standard HAVi APIs. Such a DCM would offer additional features and capabilities over a standard DCM, but would control a narrower range of devices, perhaps only a specific device or model.

3. Discussion

As it is obvious from the service discovery protocols explained above that each one has its own advantages and drawbacks. In comparison, the following points have been found noteworthy.

As related to Jini protocol, it's based on Java platform and has a number of advantages including that it provides simple service attributes that are suitable for a wide range of home appliances while other service discovery protocols may target specific appliances only.

One of the major drawbacks of Jini is that it is java platform dependent which imposes computational requirements that are over the capabilities of most home appliances which include only simple computational units for cost reasons.

On the other hand UPnP is a peer to peer service discovery protocol. One of its major advantages is that it provides high level service description capabilities in the form of XML profiles on the opposite of Jini. Also UPnP is based on TCP/IP which provides high interoperability and flexibility.

One of UPnP advantages is also a drawback from another side, as XML files provide high description capabilities they are also difficult to interpret by home appliances.

And last but not least HAVi's advantage is that it supports multi streaming communication throughout the home network, and this also can be stated as a drawback because this implies that it only targets audio and video devices while Jini and UPnP target a broader range of home appliances.

4. Conclusions

From the applied study it can be concluded that there is no single universal service discovery protocol until now, and each protocol is preferred in certain application areas while there have been researches and studies on how to integrate these service discovery protocols to work together in a single network.

Service discovery protocols are the backbone of home appliance networks and remain the major challenge in their development.

Home appliance networks are opening new horizons that will lead to new capabilities in example companies will be able to update their manufactured house hold goods online and remotely. Also companies may be informed about appliance malfunctioning and fix them online without user interaction.

5. References

1. T. Tranmanh, L.M.G. Feijs, J.J. Lukkien,*" Implementation and validation of UPnP for embedded systems in a home networking environment",*2004 .
2. Tatsuo Nakajima,Daiki Ueno, Eiji Tokunaga, Hiro Ishikawa*," A Virtual Overlay Network for Integrating Home Appliances"*,IEEE 2002;
3. http://en.wikipedia.org/wiki/Service_Discovery_Protocol
4. J. Allard, V. Chinta, S. Gundala, G. G. Richard III*," Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability"*,2005.
5. Choonhwa Lee and Sumi Helal*," PROTOCOLS FOR SERVICE DISCOVERY IN DYNAMIC ANDMOBILE NETWORKS",*2002.
6. Rodger Lea,Simon Gibbs,Alec Dara-Abrams *,"Networking Home Entertainment Devices with HAVi"* ,IEEE 2004

الخلاصة:

نظم الشبكات المنزلية تكتسب اهتماما بحثيا متزايدا في الفترة الاخيرة بسبب الجهود الكبيرة التي تقوم بها كبرى شركات الأجهزة المنزلية من أجل تجهيز الأجهزة المنزلية المستقبلية بخدمات وقدرات شبكية. بروتوكولات اكتشاف الخدمة كونها عمودا أساسيا في تطوير انظمة الشبكات المنزلية تتلقى معظم الجهود البحثية لتوفير شبكات منزلية قوية وفعالة.

في هذا البحث تتم مراجعة بروتوكولات اكتشاف الخدمة الأكثر شعبية في دراسة مقارنة تناقش مزايا وعيوب كل واحد منهم على حساب الآخرى.