# Discovery and Identification of an Application for Inter-Application Communication on a Home Network Using UPnP

Kalaiselvi Arunachalam*, Gopinath Ganapathy
Department of Computer Science, Bharathidasan University, Tiruchirappalli, 620023, India.

* Corresponding author. Tel.:+91-97862-18622; email: kalaiselvi.arunachalam@gmail.com
Manuscript submitted September 16, 2015; accepted December 22, 2015.
doi: 10.17706/jcp.11.6.488-496

**Abstract:** Thousands of applications are available for Desktop, Smartphone and Tablet environments. The inter-application communication across these devices is very limited while using them on a home network to share data. This paper discusses about the possibility of inter-application communication across devices on a home network using UPnP. It provides a mechanism to discover and identify an application on a home network using an Universally Unique Identifier (UUID) for the inter-application communication. This paper also demonstrate the discovery and identification of an application on a home network using UPnP with a prototype implementation.

**Key words:** Application ID, UUID, UPnP, inter-application communication.

## 1. Introduction

The countless applications available in the app marketplaces like iOS App store, Google Play, Windows Phone store and BlackBerry World are used on different mobile devices like Notebook, Smartphone and Tablet by billions of users around the world. An user handle these mobile devices together at home or office and in their daily life. The inter-application communication across these devices is very limited on a home network in order to transfer data like document, photo, video, URL etc. The device-to-device communication on a home network is already implemented using UPnP technology and it is limited to share image, audio and video files. But an application within a device cannot communicate with an application on another device on a home network. To communicate with an application on a particular device on a home network, basically an application must be discovered and identified on a network. Since a device is discovered on a home network using UPnP, it is possible to discover and identify an application residing within a device on a home network using UPnP and Universally Unique Identifier (UUID) mechanism. Once an application is discovered and identified, then the communication can be carried out further. Based on UPnP device architecture specification, UPnP devices communicate with each other on a home network. But there is no UPnP specification available for applications to communicate with each other on a home network. This paper explains about the possibility of inter-application communication across devices on a home network using UPnP. Also it provides a mechanism to discover and identify an application on a home network using UUID and with a prototype implementation to demonstrate the discovery and identification of an application on a home network using UPnP.

## 2. UPnP Technology

Universal Plug and Play (UPnP) is a standard that uses a collection of networking protocols to enable devices like computers, printers, internet gateways, mobile devices and intelligent appliances to automatically connect with one another and share data on a network, as in [1]. The UPnP standard is defined by the UPnP Forum which is an industry initiative of more than 1031 leading companies in computing, printing and networking, consumer electronics, home appliances, automation, control and security and mobile products. The UPnP technology is targeted for home networks, proximity networks, networks in small businesses and small public environments.

## 3. UPnP Architecture

The UPnP architecture enables peer-to-peer networking of personal computers of all types, consumer electronics, mobile devices and networked home appliances, as in [1]-[5].

- The UPnP is media and device independent which can run on any networking technologies including Ethernet, Wi-Fi, Coaxial cable, Phone Line, Power Line and Firewire.
- The UPnP is platform independent where vendors can build products on any operating system and any programming language based on their convenience.
- The UPnP is built upon IP, TCP, UDP, HTTP, SOAP and XML as in Fig. 1.
- The UPnP support zero-configuration in which a device can dynamically join a network, obtain an IP address, advertise its capabilities, search and access the capabilities of other devices and leave a network at any time automatically.
- Apart from the standard services defined by the UPnP Forum, the vendors can extend the standard devices and services by defining their own device and service types with vendor-defined actions, state variables, data structure elements and variable values.

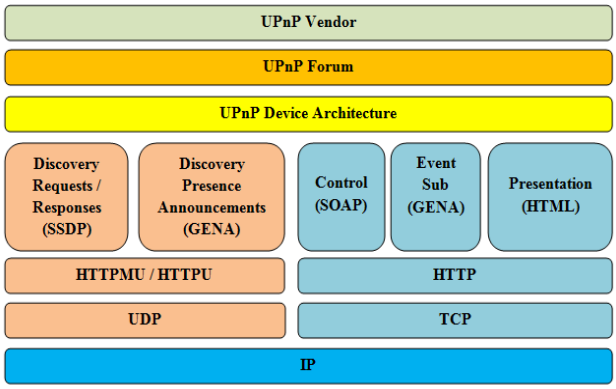


Fig. 1. UPnP architecture.

## 4. Device-to-Device Communication on a Home Network Using UPnP

The device-to-device communication is implemented on a home network using UPnP. Each device has an in-built DHCP client that search for a DHCP server when it is connected to the network to obtain an IP address. In the absence of a DHCP server, the device assigns itself a link-local address or an Auto-IP address based on a pseudo-random algorithm which is implementation-dependent and also the Auto-IP mechanism is based on the RFC-3927 specification, as in [6]. After selecting an address, an UPnP device verify that address with other devices on the network to avoid address conflicts by using Address Resolution Protocol (ARP), as in [1]. After acquiring an IP address without any address conflicts, the device is added to the network and ready to provide its services to the control points on the home network. Whenever a device is added to the network, the SSDP protocol enables that device to advertise its services to control points on

the network. Also, when a control point is added to the network, the SSDP protocol enables that control point to search for devices on the network as in Fig. 2.

Once a control point has discovered a device on network, the control point retrieve the device description from the URL provided by the device in the discovery message to interact with that device. The UPnP device description is in XML format and it includes vendor-specific manufacturer information like the model name, model number, serial number, manufacturer name, URLs to vendor-specific Web sites etc. The device description also includes a list of any embedded devices or services, URLs for control, eventing and presentation, as in [1]. For each service in the device, the service description includes the service type, service name and a list of URLs for description, control and eventing. The description for a service also includes a list of variables that represent the state of the service at run time. The UPnP device and service description documents are written by the UPnP vendor which are based on the standard UPnP Device and Service Template.

The control point retrieve the description of the device and send actions to the services of the device. The control point sends a suitable control message to the control URL for the service which is provided in the device description, as in [1]. Control messages are also expressed in XML using the Simple Object Access Protocol (SOAP). Based on the control message, the service returns action-specific values which are reflected in the state variables.

A UPnP service description includes a list of actions the service responds to and a list of variables that represent the state of the service at run time. If there is any change in the state variables, the service publishes the update in the form of event messages that contain the names of those state variables and their values to the control points that subscribed to receive these information, as in [1]. If there are multiple control points, then the service publishes the update to all control points on the home network.

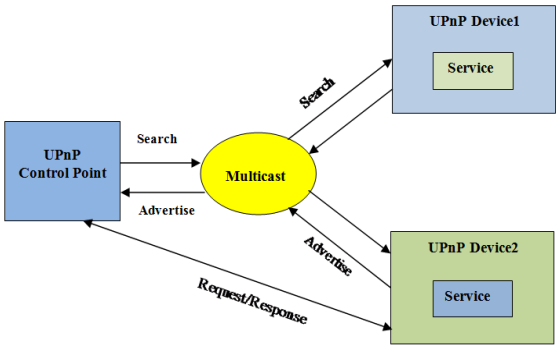


Fig. 2. UPnP device discovery architecture.

If there is any URL for presentation available in a device, then the control point retrieve the content from this URL and load it into a browser. Based on the capabilities of the page, the control point allows an user to view status of a device and control it on a home network.

UPnP device and service specifications are defined and published for Internet gateways/routers, audio-video media devices, printers, scanners, climate control, lighting and wireless LAN access points, digital security cameras, and advanced features such as security, remote user interface, IoT management and control, quality of service (QoS) etc.

## 5. Inter-Application Communication on a Home Network Using UPnP

The UPnP devices can communicate with each other on a home network and share data between them. Though device-to-device communication is implemented on a home network using UPnP, the inter-application communication is not yet implemented. There is no UPnP specification for an application to communicate with another application within a device or on another device on a home network. Since

the devices on a home network are discovered using UPnP along with their services, there is a possibility to discover multiple applications residing within these devices , as in [7] to communicate and share data between them.

### 5.1. Inter-Application Communication Using UPnP-Advantages

The advantages of UPnP inter-application communication on a home network include the following:

- No manual configuration or configuration servers required.
- No device and media dependency.
- No platform dependency.
- An application can communicate with another application within a device or another device on a home network.
- •Share any type of data between applications across devices
- Control another application residing within a device or another device on a home network.

#### 5.1.1. Examples

- A Chat application on an Android Tablet can send an URL to a Video Player application on a SmartTV to play a video in a home network.
- A voice recorder application on an iPhone can send an audio file to a Windows Media Player on a Windows PC to play the content in a home network.
- A Mail application on an Android Smartphone can send a PDF file to a PDF Reader application on an Mac PC to open it in a home network.
- An user can view the list of applications running on different devices on the home network through UPnP Control Points.
- An user can communicate with a particular application on a particular device through UPnP control points on the home network.

### 5.2. Inter-Application Communication Using UPnP-Disadvantages

The disadvantages of UPnP inter-application communication on a home network include the following:

- Each device on a home network communicate with each other through messages at large which result network traffic and slow down if there is limited bandwidth available. Multiple applications within these devices could increase the network traffic more and further slow down the home network.
- Data security is a problem on a home network as the shared data could be available externally via the internet if there is no security software available.
- The UPnP protocol does not implement any user authentication and authorization that could cause a device vulnerable to attack.

## 6. Proposed Methodology for Inter-Application Communication Using UPnP on a Home Network

Devices on a home network communicate with each other using UPnP and share data between them, as in [1]-[5]. Devices are discovered on a home network using the Auto-IP or link local addresses. An application residing within a device can acquire the IP address from that device and use an Universally Unique Identifier UUID to advertise its presence and to be discovered by other applications on the home network. When a device is added to the network, the SSDP protocol enables an application within that device to advertise the services of that application to control points on the network. Also, when a control point is added to the network, the SSDP protocol enables that control point to search for UPnP enabled devices and applications on the network as in Fig. 3.

Once a control point has discovered an application on the network, the control point retrieve the application description from the URL provided by the application in the discovery message to interact with that application. The UPnP application description is in XML format and it includes the information like the application name, application version, application type, application owner name, owner details, URLs to the application web site etc.

The application description also includes a service, URLs for control, eventing and presentation. For a service in an application, the description includes the service type, service name and a list of actions with URLs for description, control and eventing. The description for a service also includes a list of variables that represent the state of the service at run time.

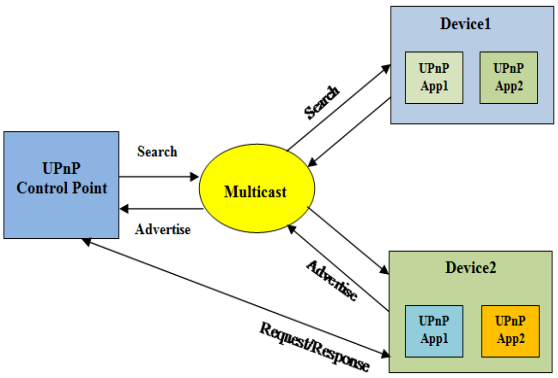


Fig. 3. Proposed UPnP application discovery architecture.

The control point retrieve the description of the application and invoke actions provided by the service of the application. The control point sends a suitable control message to the control URL for the service which is provided in the application description. Control messages are also expressed in XML using the Simple Object Access Protocol (SOAP). Based on the control message, the service returns action-specific values which are reflected in the state variables.

A UPnP application service description includes a list of actions the service responds to and a list of variables that represent the state of the service at run time. If there is any change in the state variables, the application service publishes the update in the form of event messages that contain the names of those state variables and their values to the control points that subscribed to receive these information. If there are multiple control points, then the application service publishes the update to all control points on the home network.

If there is any URL for presentation available in an application, then the control point retrieve the content from this URL and load it into a browser. Based on the capabilities of the page, the control point allows an user to view status of an application and control it on a home network.

The UPnP application and application service specifications are not yet defined by UPnP. Based on the standard UPnP Device and Service Template, it is possible to define the UPnP application and application service specification. This specification can be defined further which can be used by the UPnP vendors to develop UPnP enabled applications where by these applications can communicate with each other directly across various heterogeneous devices on the home network.

## 7. Discovery of an Application Using UUID on a Home Network

An instance of an application may run on many devices on a home network. For example, a mail application is available on a Smartphone or on a tablet or on a Smart TV on a home network. It will be difficult to identify a particular instance of this application on a particular device to communicate or share data with it. In order to communicate with a particular application on a particular device on a home

network, a common application identification mechanism is required.

### 7.1. Universally Unique Identifier (UUID)

A Universally Unique Identifier (UUID) is used to uniquely identify an object or an entity without significant central coordination. The UUID is a 128 bit value which is represented by 32 alphanumeric characters divided by 5 groups and each group separated with hyphens, as in [8]. The variant indicates the layout of the UUID and the UUID specification covers one particular variant which has five versions with different algorithms. These versions are based on MAC address & date-time, DCE security, MD5 hash & namespace, random and SHA-1 hash & namespace, as in [8]. A sample UUID look like 5ab7c955-1c4c-388e-8fd2-83ada580effb. The UUID can be created to identify an entity and the same UUID can never be created by anyone to identify anything else. UUIDs are standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE). UUIDs are normally used to identify entities or objects for a short life time across a network.

A UUID can be used to differentiate applications of same service type on different devices on a home network and to uniquely identify them, as in [9], [10]. An application can be identified by a UUID on the home network and its services can be identified by using the application service name and application service type which are defined in the application description document on the home network. A UUID can be associated with an application using a UUID generation module or an utility program so that the application can generate a UUID and advertise the same on the home network. Once an application is identified by the control point on the home network, the services provided by the application can be accessed through application service description which is in XML format. The UPnP control point can communicate further to control and share data with that application.

## 8. Prototype Implementation for Application Discovery Using UUID on a Home Network

The inter-device communication is already implemented on a home network using UPnP. The inter-application communication across different devices on a home network can be achieved by using UPnP. An application can be uniquely identified on a home network by using a UUID mechanism which can be generated by an utility program (for example, "uuidgen" program in linux) or an UUID generation module (based on RFC4122), as in [11] integrated with the application during development. The generated UUID is advertised by the application when the device is connected to the home network.

The discovery of an application using a UUID on a home network is implemented with a sample prototype as in Fig. 4. Also the prototype implementation proves that the multiple instances of the same application on different devices on the home network are uniquely identified by using the UUID as in Fig. 6.

The IACTestAPP is an application developed by using the GUPnP open source framework, as in [12]. The IACTestAPP acquire the IP address of the host on which it run and generate the UUID using uuid_generate function of libuuid (Universally Unique Identifier library in linux based on OSF-DCE) library to advertise its presence on the home network. The generated UUID is based on version 4 of the UUID specification and this algorithm creates the UUID from random or pseudo-random numbers. The Simple Service Discovery Protocol (SSDP) is used for advertisement and discovery of network resources on the home network. The IACTestAPP broadcast its presence on the home network using HTTPMU (HTTP sent over Multicast UDP). The IACTestAPP application run on a PC is discovered and identified by the UPnP Universal Control Point (UPnP Inspector is used here as in Fig. 4) run on another PC on the home network using HTTPU (HTTP sent over Unicast UDP).

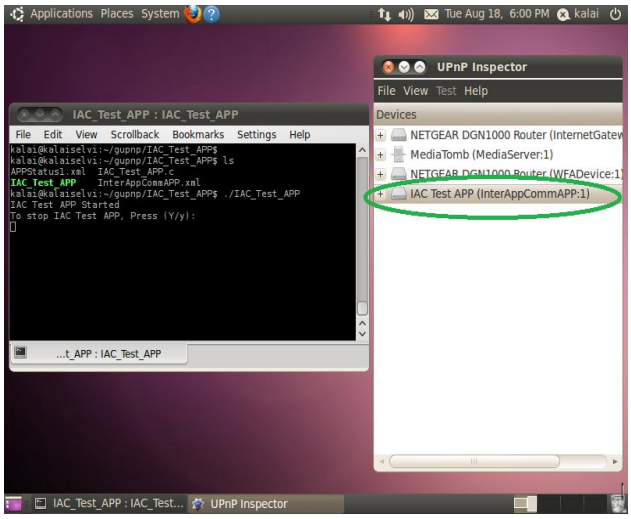


Fig. 4. IACTestAPP discovered by UPnP inspector.

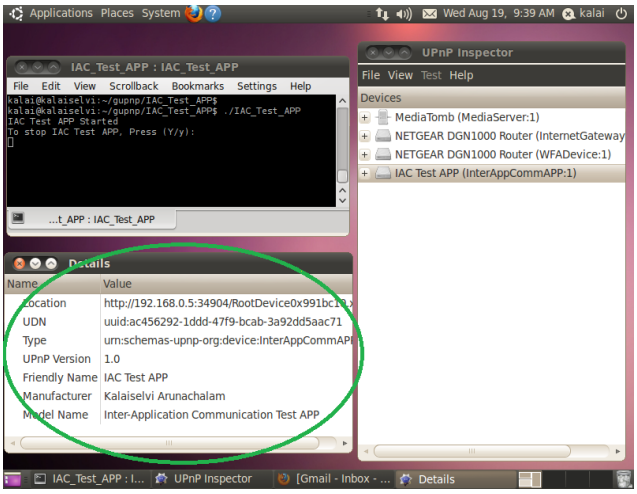


Fig. 5. UPnP Inspector access the details of IACTestAPP after discovery.

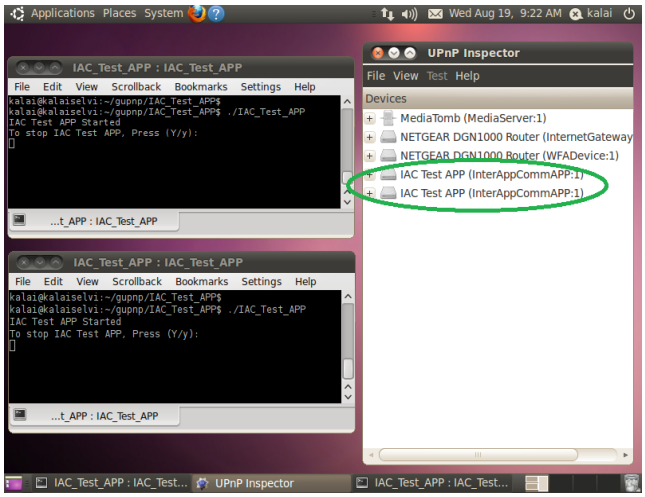


Fig. 6. Multiple instances of IACTestAPP discovered by UPnP Inspector.

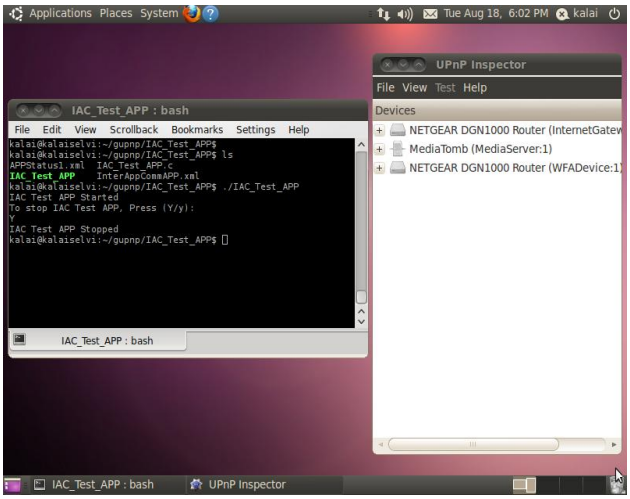


Fig. 7. IACTestAPP delisted from the UPnP Inspector instantly once if it is stopped.

The UPnP Inspector can access the details of each instance of the IACTestAPP application on the home network through the application description document as in Fig. 5 that is obtained from the discovery message which contain the application id (UUID), application name, application owner name, application version etc.

Also the multiple instances of the IACTestAPP applications running on different PCs are discovered and identified by the UPnP Universal Control Point (UPnP Inspector) running on another PC on the home network as in Fig. 6.

The IACTestAPP is delisted instantly from the UPnP Inspector once if it is stopped as in Fig. 7. The IACTestAPP is dynamically listed or delisted in the UPnP Inspector once if it is started or stopped on devices on the home network.

## 9. Conclusion

The availability of device-to-device communication and the lack of inter-application communication on a home network are discussed. Since the device-to-device communication on the home network is already implemented using UPnP, it is possible to implement inter-application communication on a home network using UPnP. To communicate with an application directly on a device, an application must be discovered

and identified uniquely on a home network by using an UUID and with the IP address of the associated host or device. Once an application is discovered and identified on a home network, the UPnP control point can communicate with it directly, control it and share data as well.

A sample prototype application is implemented and the application is discovered and identified by the UPnP Universal Control Point on a home network using the UUID and the IP address of the associated device. The details of a particular instance of the sample application is accessed by the control point on the home network. The sample application is dynamically listed or delisted in the UPnP Universal Control Point once if it is started or stopped. Based on the UPnP Standard Device and Service specification, an UPnP Application and Application Service specification can be analyzed, defined and enhanced further. This standard would help UPnP vendors to develop UPnP enabled applications for an effective inter-application communication across heterogeneous devices on a home network.

## References


[1] Andrew, D., Bryan, R., Maarten, B., John, G., Alan, M., Kim, Y. S., Bruce, F., & Jonathan, T. (2015). *UPnP Device Architecture 2.0*. United States: UPnP Forum.

[2] Bendaoud, K. T., & Merzougui, R. (2013). Service discovery — a survey and comparison. *International Journal of UbiComp,* 23-36.

[3] Ververidis, C. N., & Polyzos, G. C. (2008). Service discovery for mobile Ad Hoc networks: A survey of issues and techniques. *Communications Surveys & Tutorials*, 30-45.

[4] Michael, J. (2003). *UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play.* United States: Intel Press.

[5] Elena, M., Janne, R., Marina, P., & Petri, M. (2008). A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 2097-2128.

[6] Cheshire, S., Aboba, B., & Guttman, E. (2005). *Dynamic Configuration of IPv4 Link-Local Addresses (RFC3927)*. United States: The Internet Engineering Task Force.

[7] Greg, S., Osama, A.-S., Dann, W., & Ralph, N. (2014, July). System and method for providing an inter-application communication framework. *United States Patent.* Retrieved July 16, 2015, from http://www.google.tl/patents/US20140195582

[8] Paul, J. L., & Rich, S. (1997). *UUIDs and GUIDs*. United States: The Internet Engineering Task Force.

[9] Roi, T., Guy, R., Yuval, A., Eran, F., & Gadi, E. (2013, October). Proxy and method for determination of a unique user identification for a plurality of applications accessing the web from a mobile device. *United States Patent*. Retrieved July 16, 2015, from http://www.google.com/patents/US20130262675

[10] Yakov, S. (2015). *Bluetooth Data Exchange between Android Phones without Pairing*. United States: Noom Inc.

[11] Leach, P., Mealling, M., & Salz, R. (2005). *A Universally Unique IDentifier (UUID) URN Namespace (RFC4122)*. United States: The Internet Engineering Task Force.

[12] Jens, G., Jorn, B., Ross, B., & Zeeshan, A. (2013). *The GUPnP Framework*. United States: The GNOME Foundation.


**Kalaiselvi Arunachalam** received the B.Sc. degree in physics from the University of Madras, India and M.C.A degree in computer applications from the Anna University, India. She is currently a Ph.D. scholar in the Department of Computer Science, Bharathidasan University, India. Her research interests include home networking, communication software and systems.

**Gopinath Ganapathy** received the B.Sc. degree in computer science from the Bharathidasan University, India, M.C.A degree in computer applications from the St. Joseph's College Autonomous, India and Ph.D from the Madurai Kamaraj University, India.

He is currently the Professor and Head of School of Computer Science Engineering and Applications, Bharathidasan University, India.

Dr. Gopinath Ganapathy is a professional member in IEEE, ACM, CSI, and ISTE. His research interests include semantic web, NLP, ontology, and text mining.