

---

# 2II45 - Architecture of Distributed Systems

## Assignment 3

---

Vincent Nuttin - v.m.nuttin@student.tue.nl  
Erasmus Student 2010-2011  
UCLouvain.be - Belgium

December 29, 2010

### Contents

<b>1</b>	<b>Architectural description</b>	<b>1</b>
1.1	General architecture . . . . .	1
1.2	Use case / Scenario : A user wants to control a device . . . . .	3
1.3	Use case / Scenario : A new device is added to the network . . . . .	4
1.4	Architectural patterns in UPnP . . . . .	5
<b>2</b>	<b>Analysis of some concerns</b>	<b>5</b>
2.1	Scalability - parameter : # devices, metric : response time . . . . .	5
2.2	Modifiability - add new or change service . . . . .	5
2.3	Not very well addressed concerns . . . . .	6
<b>3</b>	<b>Discovery protocol - SSDP</b>	<b>6</b>

The main goal of **UPnP** (Universal Plug and Play) is to provide a simple way to connect *devices* (which can offer *services*) on a network. UPnP allows to do this thanks to protocols which are defined over standard layers of the internet network.

There exists another technology named “Plug and Play”. This technology is used for connecting devices on a computer (via USB, or something). In this document, we will talk about UPnP which is the technology related to network devices (and services).

## 1 Architectural description

A UPnP network is composed by two types of entities :

- **Devices** are components which provide one (or more) service(s);
- **Control points** are components which are aware of the devices in the network and can interact with them (using specific (control or event) messages).

Note that a device can contain control points as well.

### 1.1 General architecture

Roughly, UPnP network principles can be described as following. There are six major operations. These operations are described in many documents, but especially in a clear way in [3] and [5].

1. **Addressing:** The device joins the network. It must ask the local DHCP server for a unique IP address. If there exists no DHCP server on the network, the device must assign itself an IP address (“AutoIP”). This address will be used for each operation on the network. If the device received a domain name during the DHCP transaction, this name will be used instead of the IP address.
2. **Discovery:** The discovery protocol of UPnP is known as the “*Simple Service Discovery Protocol*” (*SSDP*). More details are provided in section 3 of this document but, basically, it allows multiple things.
  - When a device is added to the network, this device advertises its services to control points in an automatic way;
  - When a control point is added to the network, this control point can search through the network and find some already connected and running devices.

In both cases, discovery messages are exchanged containing a few (essential) information about the control point or the device (and its service(s)). In these essential information, there are the type of the device, its ID, and a pointer to a more detailed description document.

3. **Description:** Each device provides a XML document containing all the information needed to interact correctly with it. This description document acts as the documentation file of the services provided by the device (list of commands, actions, parameters, arguments, etc.)

4. **Control:** With the complete description file, the control point can send actions to the device service. The control point build some kind a control message in XML using *SOAP* (“*Simple Object Access Protocol*”). The action is then executed by the device and may lead to variable changes in run-time state description.
5. **Eventing:** It is possible that more than one control point are present in the network. To deal with that, UPnP uses an event notification protocol known as *GENA* (for “*General Event Notification Architecture*”). When a variable changes in a service state, the device responsible for this service sends an event message (in XML) to all subscribed control point. The event messages contain the name of the variable and its value at this time. Thanks to this protocol, all control points are equally informed about the service variables.
6. **Presentation:** A device can have a URL for presenting. This URL allows a user (via a control point) to control the device and/or view device status.

All these protocols (DHCP, SSDP, HTTP, SOAP, GENA, ...) run over standard layers of internet networks such as IP, TCP, UDP, etc.

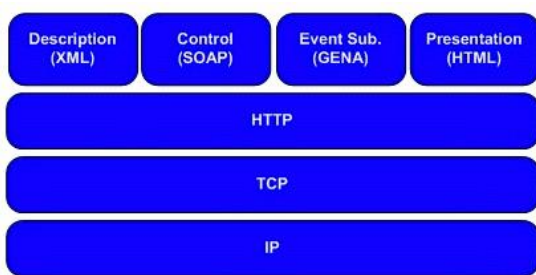


Figure 1: UPnP on TCP

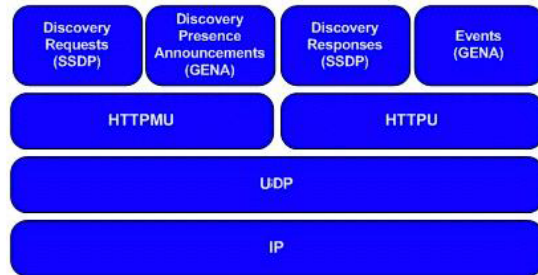


Figure 2: UPnP on UDP

Figures 1 and 2 show the layers involved in the different protocols. As you can see, the description, control and presentation operations run mainly over HTTP/TCP/IP (figure 1). The discovery process runs mainly over UDP/IP. “HTTPMU” stands for HTTP over UDP in multicast mode. “HTTPU” stands for HTTP over UDP in unicast mode (figure 2).

## 1.2 Use case / Scenario : A user wants to control a device

On figure 3, you can observe the (very) simple way to control a device.

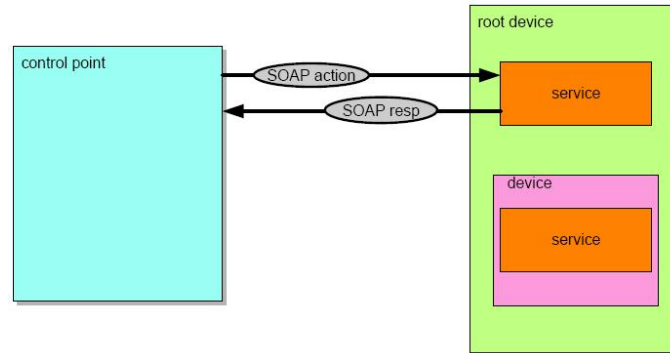


Figure 3: Device controlled by a user

As we said earlier, the control mechanism is driven by control messages. In his article, Michael Jeronimo reminds us the fact that “*UPnP relies on the Simple Object Access Protocol (SOAP) for device control*” [5]. Roughly, **SOAP** relies on XML<sup>1</sup> for the message format and on RPC<sup>2</sup> and HTTP<sup>3</sup> for the transmission of it across the network. The use of XML, and thanks to its robustness, allows the programmers to send highly complex data types. As it is explained in the general architecture section, this control messages often lead to some kind of “execution” on the device which may lead to changes in service state table.

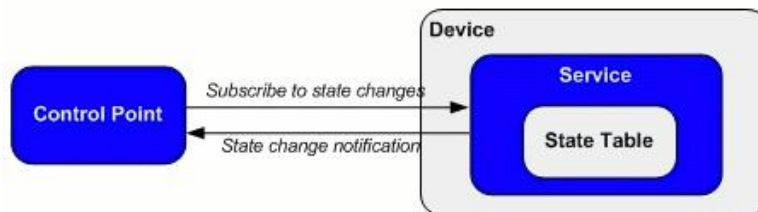


Figure 4: State changes

Figure 4 shows the eventing process. A notification of changes is sent to all control point which had asked to subscribe to the device.

---

<sup>1</sup> Extensible Markup Language

<sup>2</sup> Remote Procedure Call

<sup>3</sup> HyperText Transfer Protocol

## Example

Let's assume this UPnP network :

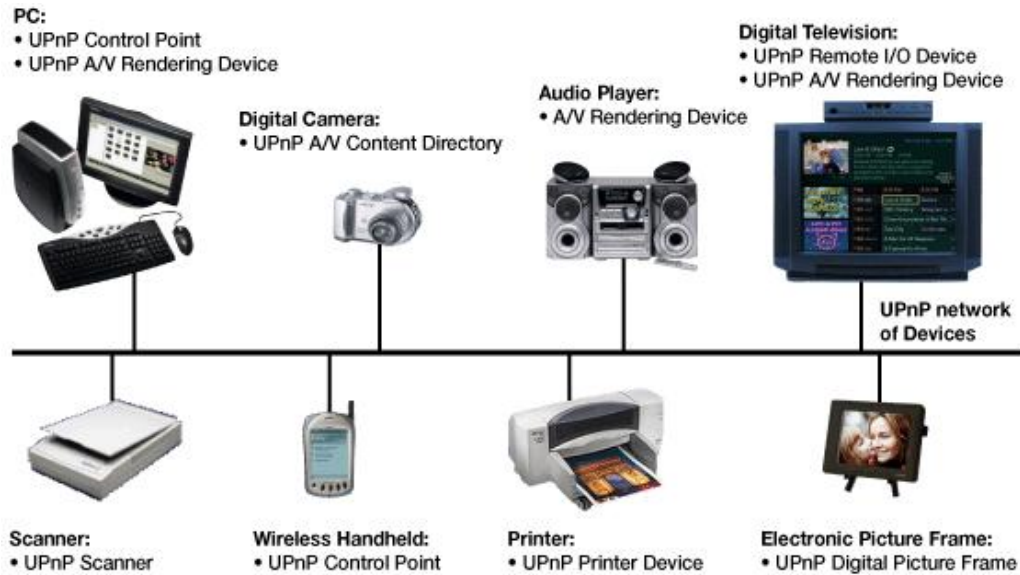


Figure 5: UPnP Network (from [4])

With this network, a user may want to see the current picture of the Electronic Picture Frame (on the bottom right of the network) using its Wireless Handheld. This small computer can act as a control point. Hence, it can send message to the Electronic Picture Frame asking for its current picture. We assume here that the network is running correctly, each device is addressed, discovered, fully described and that the Electronic Picture Frame has the ability to send pointers to pictures.

### 1.3 Use case / Scenario : A new device is added to the network

The full process from connecting a device to using a service of this device can be expressed in four main points (see figure 6).

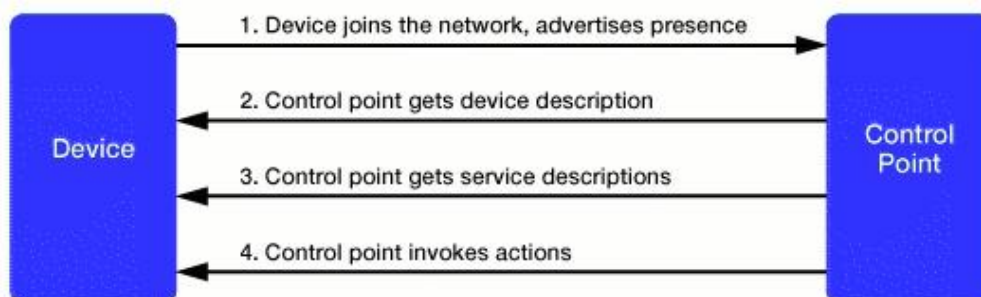


Figure 6: A device is added to the network

It is important to notice that this graphical example shows only one control point. Obviously, the communication between the new device (recently added) and all control points in the network is exactly the same. So, if the network contains more than one control point, the same communication scheme is taking place between these control points and the device.

## 1.4 Architectural patterns in UPnP

It is possible to identify three major architectural patterns in UPnP. The first one is the **service-oriented pattern** because, in a first time, devices offer services. After, those services are discovered by applications and, finally, applications can use services via control points. The second way to see the system is as a **peer-to-peer architecture** because UPnP allows lots of different devices to act as nodes and interact with the other nodes with simple protocols (see figure 5). Finally, this architecture presents many similarities with a **layered architecture** because of its “networking structure”.

To deal with interactions between components, we can roughly see that two main patterns are used. **SOAP** is used for control messages and **GENA** for event messages.

## 2 Analysis of some concerns

Building such a system always implies scalability and modifiability concerns. But other concerns can be analysed as well.

### 2.1 Scalability - parameter : # devices, metric : response time

I'm not sure I understand “response time” correctly but, for me, that means the time between the moment you ask something to a device and the moment you've got the answer from this device. From a scalability point of view, I don't see any significant relation between the number of device and the response time. So, I will say that scalability is well addressed in this case.

Nevertheless, a scalability problem can occur when we are talking about discovery process (metric can be : “discovery time”). If we must deal with a huge number of devices (including control points), the time needed to send our discovery presence announcements (GENA over HTTPMU over UDP) can grow very quickly and may lead to really bad usability of the network.

### 2.2 Modifiability - add new or change service

In my opinion, modifiability is well addressed in UPnP because we typically have a strong cohesion and a weak coupling between services. I mean, services are different and a change in one of them does not lead to any interference with another one.

## 2.3 Not very well addressed concerns

In my opinion, there are two main concerns which are not so well addressed in UPnP.

### Security

Security is the main issue of UPnP. There is a lack of authentication. Actually, there exists some authentication mechanisms with UPnP but they are really complex. So, a lot of UPnP implementations do not use these mechanisms and let the system unsecured. The exact behaviour to have when you want to use this security issue is out of this document scope; but this security issue may lead to networks vulnerable to attacks (hackers can control web sites from Adobe Flash programs using problems in router implementations, etc.).

### Usability

I think that usability may be a bad addressed concern but I must explain in which circumstances. Devices have the opportunity to provide a “presentation URL” (as explain in section 1.1). If they provide such kind of “presentation URL”, there is no problem. But, since some devices do not provide any kind of “presentation URL”, usability is not well addressed because it becomes really hard to monitor these devices (e.g. to know if the device is ready or not, busy or not, connected or not, crashed or not, etc.)

## 3 Discovery protocol - SSDP

The discovery protocol of UPnP is known as the “*Simple Service Discovery Protocol*” (*SSDP*). SSDP is a protocol of automatic discovery allowing identification of devices on a network using a broadcast over UDP. This protocol is not often used because of the security issues described in section 2.3.

SSDP is targeted for use in residential or small office environments [1]. Here are some major points to think about if we want to extend SSDP across a local network.

- First, we must think about the scope of the broadcast used in SSDP. As we said, SSDP uses a broadcast over UDP to discover peers and, if we extend SSDP across a local network, the number of UPnP devices must be really big ! So, we need to limit the scope of this broadcast with some kind of TTL <sup>4</sup> or something like that.
- Secondly, we must think about NAT traversal. Many routers already implement some kind of functionality allowing an external source to access an internal device but these kind of behaviour would become mandatory if we allow external UPnP devices to access internal UPnP devices and services.
- Thirdly, we still have this security issue described in section 2.3. If we allow discovery across our local network, it would become really dangerous to use this network as a “private” local network as well.

---

<sup>4</sup> Time To Live

## References

- [1] Simple service discovery protocol (wikipedia.org). Webpage consulted on November 27<sup>th</sup>, 2010 : [http://en.wikipedia.org/wiki/Simple\\_Service\\_Discovery\\_Protocol](http://en.wikipedia.org/wiki/Simple_Service_Discovery_Protocol).
- [2] Soap (wikipedia.org). Webpage consulted on November 28<sup>th</sup>, 2010 : <http://en.wikipedia.org/wiki/SOAP>.
- [3] Universal plug and play (wikipedia.org). Webpage consulted on November 26<sup>th</sup>, 2010 : [http://en.wikipedia.org/wiki/Universal\\_Plug\\_and\\_Play](http://en.wikipedia.org/wiki/Universal_Plug_and_Play).
- [4] Upnp - le multimedia et la domotique. Webpage consulted on November 30<sup>th</sup>, 2010 : [http://www.maison-domotique.com/dossiers/20090515\\_upnp.php](http://www.maison-domotique.com/dossiers/20090515_upnp.php).
- [5] Michael Jeronimo. It just works: Upnp in the digital home. *The Journal of Spontaneous Networking*, October 5<sup>th</sup>, 2004. Webpage consulted on November 26<sup>th</sup>, 2010 : [http://www.artima.com/spontaneous/upnp\\_digihome.html](http://www.artima.com/spontaneous/upnp_digihome.html).
- [6] UPnP Forum. *UPnP Device Architecture 1.1*, October 15<sup>th</sup>, 2008.