Department of Computer Science
and Information Systems
Birkbeck College, University of London

MSc Computer Science

Internet and Web Technologies

Universal Plug and Play

Purpose, protocols and security concerns

Author:
Alexander Worton

1 The purpose of UPnP

The purpose of Universal Plug and Play (UPnP) is similar to that of Zero-Conf, the technology used in Apple's Bonjour service [19]. UPnP provides easy connectivity between networked Internet Protocol (IP) devices by enabling zero configuration auto discovery and connectivity between them [7].

When a UPnP device is powered on, it automatically attempts to connect to the network, obtain an IP address and then broadcast and receive information about itself and other devices on the network, including the capabilities of what it and others can do [4].

The original focus of UPnP was on devices within the home, such as media servers, Airplay devices, printers and security cameras, however the scope has since grown to include connected cars, work and industry [8].

UPnP brings the benefits that Plug 'n' Play has for directly connected devices such as USB sticks to IP networked devices; the ability to connect a new device to the network and have it just work with other connected devices such as PCs and smart phones, without requiring technical configuration and set up before use.

2 UPnP Protocols

The UPnP architecture describes six levels of protocols including addressing, discovery, description, control, eventing and presentation.

2.1 Addressing

The first level of the UPnP protocol deals with establishing a presence on the network. This involves obtaining an IP address. UPnP devices implement Automatic Private IP Addressing (AutoIP) [13] to obtain or generate an IP and address themselves. Firstly, the device will attempt to obtain an IP address via Dynamic Host Configuration Protocol (DHCP) [9]. This protocol uses UDP on ports 67 and 68 for the server and client respectively.

Initially, the device will broadcast a Discover UDP message to the whole network using the broadcast address 255.255.255.255 as the destination. The message will include the source IP address of 0.0.0.0 as it does not yet have one, and will be sent from UDP port 68 to port 67 [5], ensuring that a server is the one to receive the message.

Source MAC addr	Dest MAC addr	Source IP addr	Dest IP addr	Packet Description
Client	Broadcast	0.0.0.0	255.255.255.255	DHCP Discover
DHCPsrvr	Broadcast	DHCPsrvr	255.255.255.255	DHCP Offer
Client	Broadcast	0.0.0.0	255.255.255.255	DHCP Request
DHCPsrvr	Broadcast	DHCPsrvr	255.255.255.255	DHCP ACK

Table 1: Client-server DHCP conversation summary [14]

Following receipt of the discovery message, the DHCP server will respond by broadcasting an offer which will state the offered IP address and the time duration that it will be valid for. The source and destination ports for the UDP offer message will be reversed, such that the source is server port 67 and destination 68. This needs to be broadcast as the client still does not have an IP address at this time and cannot otherwise be located [5], but clients with an established IP address will ignore the message.

The client will then broadcast a request message which will indicate the specific IP and duration of the offer that it is accepting. The final response from the server is to acknowledge the accepted offer, and upon client receipt of the ACK, the client will start using the confirmed IP address.

Should the device fail to obtain an IP address through DHCP, it will fail over to generating and assigning a link-local IP address to itself in the range of 169.254.0.0/16 [18].

2.2 Discovery

Once established on the network, the UPnP device will broadcast a series of Simple Service Discovery Protocol (SSDP) [9] messages to announce its presence and notify other devices of its available services via a link to its device description document (DDD). These messages are sent over HTTP using UDP rather than the more common and fault tolerant TCP protocol as this is connectionless and allows for multicast. This protocol is referred to as HTTPMU when sent multicast and HTTPU when sent unicast. [1].

It is also possible for control devices to broadcast search requests. Again, these are sent over HTTPMU with receiving devices listening on the multicast port. Upon receipt of a search request, a client device will inspect the message

to determine if it meets the search criteria, for example a specific device type such as a media player. If the receiving device determines that it matches the request, a unicast response is sent directly back to the control unit that originated the search request using HTTPU with a link to its DDD [9].

2.3 Description

The device description document is an XML format document which holds information describing the device which may include a unique identifier (UUID), device type, manufacturer and model name along with a list of available services and the presentation Uniform Resource Locator (URL) [2].

Once a device obtains an SSDP broadcast message or search response, in order to learn more about the sender device it needs to download and parse the DDD XML file from the provided URL using HTTP. This will include a list of services the device offers which may also provide a link to a service Description, also known as Device Control Protocols (DCP) [9].

A DCP, like the DDD is another XML document containing details of the service including how to use it. This will include the control URLs, messages and responses that the service will use to handle and notify results of received control messages.

2.4 Eventing

In addition to the information a device provides on controlling services, it can also provide information regarding the service's state. Through the Generic Event Notification Architecture (GENA) [9], an eventing protocol which uses HTTP over TCP/IP and multicast UDP for communication [15], one device can subscribe to another device. Upon initial registration a subscribed device will be sent an XML document with a list of all the state variables and their current values so the subscribing device can initialise them [9].

When the state changes or is updated, the device publishes, in XML format, the state variable and the new value to all subscribers. This happens regardless of whether the variable changed internally or from a control action, and there is no ability to restrict or select which state values to observe [9].

2.5 Presentation

A device can provide an HTML page for access over HTTP using TCP/IP as a presentation layer allowing access to and control of its services by a user on a control device. The URL for this interface is published in the DDD, and the implementation and functionality of the page are determined by the device vendor [9].

Simple Object Access Protocol (SOAP) [22] defines how HTTP and XML are used to send control messages and receive reponse data sent back from the device such as status codes.

A criticism of the presentation layer is that it does not always provide access to all the information and control that is available via SOAP [11].

3 Security concerns

UPnP provides an ease of use in auto-configuring network devices. The cost of this convenient ease of use is security. Despite the UPnP forum publishing a design document for implementing device security in 2003 [6], devices are not required to implement any security in order to be UPnP compliant [21] [10] and most do not implement the security design document [21].

A number of security concerns have arisen, some due to the design of UPnP itself and others due to its implementations, such as one from 2002 where the Windows XP UPnP implementation was vulnerable to a remotely executable buffer overflow and both a Distributed (DDOS) and non-Distributed (DOS) Denial of Service attack [20] [10].

Another widespread security concern of UPnP was a result of routers which suffered from a flaw whereby they acted on UPnP control requests from the Wide Area Network (WAN) [17] as well as the Local Area Network (LAN) [16]. This allowed an attacker to open ports on the router from outside the network using UPnP control requests.

While this was not an issue with UPnP itself, the architecture of UPnP provides security concerns. By design, all nodes on the LAN are trusted, so any compromised device is capable of compromising the rest of the network. Even when excluding control points, which are the only devices covered by the UPnP forum security document [6], all devices will respond to search and information requests made under discovery which can help to identify vulnerable devices through their manufacturer and device information.

The effect of absent security in UPnP devices, particularly the router or gateway devices is the ability to alter DNS to re-route traffic to a different site, perform man-in-the-middle attacks [12] and hijack devices for DOS and DDOS attacks [3].

References

[1] Upnp device architecture. `https://embeddedinn.wordpress.com/tutorials/upnp-device-architecture/`. (No date) Accessed: 15 April 2017.

[2] Upnp service discovery tutorial and user guide. `https://www.appinf.com/docs/poco/00200-UPnPSSDPTutorialAndUserGuide.html`. (2017) Accessed: 15 April 2017.

[3] Akamai. Akamai warns of upnp devices used in ddos attacks. `https://www.akamai.com/uk/en/about/news/press/2014-press/akamai-warns-of-upnp-devices-used-in-ddos-attacks.jsp`, October 2014. Accessed: 17 April 2017.

[4] Stephen J. Bigelow. Universal plug and play: Networking made easy. http://www.pcmag.com/article2/0,2817,1231047,00.asp, September 2003. Accessed: 11 April 2017.

[5] Pieter De Decker. Automatic ip address assignment: How dhcp works. `https://www.youtube.com/watch?v=RUZohsAxPxQ`, October 2013. Accessed: 15 April 2017.

[6] UPnP Forum. Upnp security ceremonies. Technical report, institution, November 2003.

[7] UPnP Forum. Upnp™ device architecture 1.1, October 2008.

[8] UPnP Forum. Upnp: The discovery & service layer for the internet of things, April 2015.

[9] Tom Fout. Universal plug and play in windows xp. Technical report, Microsoft, https://technet.microsoft.com/en-us/library/bb457049.aspx, August 2001. Accessed: 11 April 2017.

[10] AAMM Haque. Upnp networking: Architecture and security issues. In *TKK Seminar on Network Security*. Citeseer, 2007.

[11] Armijn Hemel. Upnp stack layout. `http://www.upnp-hacks.org/upnp.html`, 2006-2011. Accessed: 15 April 2017.

[12] Computer Hope. Man-in-the-middle attack. `http://www.computerhope.com/jargon/m/mitma.htm`. (No date) Accessed: 17 April 2017.

[13] Charles M. Kozierok. Dhcp autoconfiguration / automatic private ip addressing (apipa). `http://www.tcpipguide.com/free/t_DHCPAutoconfigurationAutomaticPrivateIPAddressingA.htm`, 2001-2005. Accessed: 17 April 2017.

[14] Microsoft. Dhcp dynamic host configuration protocol basics. Technical report, Microsoft, https://support.microsoft.com/en-gb/help/169289/dhcp-dynamic-host-configuration-protocol-basics. (No date) Accessed: 11 April 2017.

[15] Microsoft. Understanding universal plug and play. Technical report, Microsoft Corporation, 2000.

[16] Bradley Mitchell. An introduction to the essential concepts of a lan. `https://www.lifewire.com/local-area-network-816382`, March 2017. Accessed: 17 April 2017.

[17] Bradley Mitchell. Wan definition and an explanation on how wans work. `https://www.lifewire.com/wide-area-network-816383`, March 2017. Accessed: 17 April 2017.

[18] Petri Palmila. Zeroconf and upnp techniques. *Helsinki University of Technology, Tech. Rep*, 2007.

[19] pcmag.com. Upnp encyclopedia term. http://www.pcmag.com/encyclopedia/term/53504/upnp. (No date) Accessed: 11 April 2017.

[20] Paul Schmehl. title. `https://www.symantec.com/connect/articles/microsoft-upnp-universal-plug-and-play-vulnerability`, February 2002. Accessed: 17 April 2017.

[21] Kristian Selén. Upnp security in internet gateway devices. In *TKK T-110.5190 Seminar on Internetworking*, 2006.

[22] W3Schools. Xml soap. `https://www.w3schools.com/xml/xml_soap.asp`. (No date) Accessed: 15 April 2017.