Universal Plug and Play

From Wikipedia, the free encyclopedia
(Redirected from Upnp)

Universal Plug and Play (**UPnP**) is a set of computer network protocols promulgated by the UPnP Forum. The goals of UPnP are to allow devices to connect seamlessly and to simplify the implementation of networks in the home and corporate environments. UPnP achieves this by defining and publishing UPnP device control protocols built upon open, Internet-based communication standards.

UPnP is distinct from Plug-and-play, a technology for dynamically attaching devices to a computer directly.

Contents

Overview

The UPnP architecture [1] (http://www.upnp.org/download/UPnPDA10_20000613.htm) offers pervasive peer-to-peer network connectivity of PCs, intelligent appliances, and wireless devices. The UPnP architecture is a distributed, open networking architecture that uses TCP/IP and http to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and everywhere in between.

It enables data communication between any two devices under the command of any control device on the network.

- Media and device independence. UPnP technology can run on any medium including phone lines, power lines (PLC), Ethernet, IR (IrDA), RF (Wi-Fi, bluetooth), and FireWire. No device drivers are used; common protocols are used instead.
- Common base protocols. Base protocol sets are used, on a per-device basis.
- User interface (UI) Control. UPnP architecture enables vendor control over device user interface and interaction using the web browser.
- Operating system and programming language independence. Any operating system and any programming language can be used to build UPnP products. UPnP does not specify or constrain the design of an API for applications running on control points; OS vendors may create APIs that suit their customer's needs. UPnP enables vendor control over device UI and interaction using the browser as well as conventional application programmatic control.
- Internet-based technologies. UPnP technology is built upon IP, TCP, UDP, HTTP, and XML, among others.
- Programmatic control. UPnP architecture also enables conventional application programmatic control.
- Extendable. Each UPnP product can have value-added services layered on top of the basic device architecture

by the individual manufacturers.

The UPnP architecture supports zero-configuration, *invisible* networking and automatic discovery for a breadth of device categories from a wide range of vendors, whereby a device can dynamically join a network, obtain an IP address, announce its name, convey its capabilities upon request, and learn about the presence and capabilities of other devices. DHCP and DNS servers are optional and are only used if they are available on the network. A device can leave a network smoothly and automatically without leaving any unwanted state information behind.

The foundation for UPnP networking is IP addressing. Each device must have a Dynamic Host Configuration Protocol (DHCP) client and search for a DHCP server when the device is first connected to the network. If no DHCP server is available, i.e., the network is unmanaged, the device must assign itself an address. If during the DHCP transaction, the device obtains a domain name, e.g., through a DNS server or via DNS forwarding, the device should use that name in subsequent network operations; otherwise, the device should use its IP address.

Protocol

Discovery

Given an IP address, the first step in UPnP networking is discovery. When a device is added to the network, the UPnP discovery protocol allows that device to advertise its services to control points on the network. Similarly, when a control point is added to the network, the UPnP discovery protocol allows that control point to search for devices of interest on the network. The fundamental exchange in both cases is a discovery message containing a few, essential specifics about the device or one of its services, e.g., its type, identifier, and a pointer to more detailed information. The UPnP discovery protocol is based on the Simple Service Discovery Protocol (SSDP).

Description

The next step in UPnP networking is description. After a control point has discovered a device, the control point still knows very little about the device. For the control point to learn more about the device and its capabilities, or to interact with the device, the control point must retrieve the device's description from the URL provided by the device in the discovery message. The UPnP description for a device is expressed in XML and includes vendor-specific, manufacturer information like the model name and number, serial number, manufacturer name, URLs to vendor-specific web sites, etc. The description also includes a list of any embedded devices or services, as well as URLs for control, eventing, and presentation. For each service, the description includes a list of the commands, or actions, to which the service responds, and parameters, or arguments, for each action; the description for a service also includes a list of variables; these variables model the state of the service at run time, and are described in terms of their data type, range, and event characteristics.

Control

The next step in UPnP networking is control. After a control point has retrieved a description of the device, the control point can send actions to a device's service. To do this, a control point sends a suitable control message to the control URL for the service (provided in the device description). Control messages are also expressed in XML using the Simple Object Access Protocol (SOAP). Like function calls, in response to the control message, the service returns any action-specific values. The effects of the action, if any, are modeled by changes in the variables that describe the run-time state of the service.

Event notification

The next step in UPnP networking is event notification, or "eventing". A UPnP description for a service includes a list of actions the service responds to and a list of variables that model the state of the service at run time. The service publishes updates when these variables change, and a control point may subscribe to receive this information. The service publishes updates by sending event messages. Event messages contain the names of one or more state variables and the current value of those variables. These messages are also expressed in XML and formatted using the General Event Notification Architecture (GENA). A special initial event message is sent when a control point first subscribes; this event message contains the names and values for all evented variables and allows the subscriber to initialize its model of the state of the service. To support scenarios with multiple control points, eventing is designed

to keep all control points equally informed about the effects of any action. Therefore, all subscribers are sent all event messages, subscribers receive event messages for all "evented" variables that have changed, and event messages are sent no matter why the state variable changed (either in response to a requested action or because the state the service is modeling changed).

Presentation

The final step in UPnP networking is presentation. If a device has a URL for presentation, then the control point can retrieve a page from this URL, load the page into a web browser, and depending on the capabilities of the page, allow a user to control the device and/or view device status. The degree to which each of these can be accomplished depends on the specific capabilities of the presentation page and device.

Problems with UPnP

- UPnP uses HTTP over UDP (known as HTTPU and HTTPMU for unicast and multicast), even though this is not standardised and is specified only in an Internet-Draft that expired in 2001. [2] (http://www.upnp.org/download/draft-goland-http-udp-04.txt)
- UPnP does not have a lightweight authentication protocol, while the available security protocols are complex. As a result, many UPnP devices ship with UPnP turned off by default as a security measure.

NAT traversal

UPnP comes with a solution for Network Address Translation (NAT) traversal: Internet Gateway Device (IGD) protocol.

See also

- domotics
- zeroconf
- Bonjour
- JINI
- SNMP
- OSGi
- Service Location Protocol
- Salutation
- CORBA

References

- Golden G. Richard: *Service and Device Discovery : Protocols and Programming*, McGraw-Hill Professional, ISBN 0071379592
- Michael Jeronimo, Jack Weast: *UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play*, Intel Press, ISBN 0971786119

External links

Documentation

- UPnP™ Forum (http://www.upnp.org/) , Universal Plug and Play Device Architecture (http://www.upnp.org/download/UPnPDA10_20000613.htm)
- The Jini, Vision (http://developer.java.sun.com/developer/technicalArticles/jini/JiniVision/jiniology.html)
- technique comparison (http://cswl.com/whiteppr/tech/upnp.html#b)
- Universal Plug and Play in Windows XP (http://www.microsoft.com/technet/prodtechnol/winxppro/evaluate/upnpxp.mspx)

API / SDK

- http://www.cybergarage.org/net/upnp/java/index.html, http://sourceforge.net/projects/cgupnpjava/
- An Open Source UPnP Development Kit (http://upnp.sourceforge.net/)
- Code examples and tools for UPnP development by Intel (http://www.intel.com/technology/upnp/)
- UPnP (ANSI C and Java) stacks and solutions from ProtoSys (http://www.protosys.com/)
- Java Open Source UPnP SDK (http://www.sbbi.net/site/upnp/)
- Using VBScript to control a UPnP firewall (http://www.knoxscape.com/Upnp/NAT.htm)
- Nattraverso: a Python library for NAT traversal (https://nattraverso.python-hosting.com/)
- .NET library supporting UPnP (http://www.hypermoose.com/)

Software

- Emulemorph is a fork of the popular p2p application emule and supports automatic firewall configuration using upnp (http://emulemorph.sourceforge.net/)
- TwonkyMedia: a UPnP AV Media Server (http://www.twonkyvision.de)
- TVersity: a UPnP AV Media Server (http://www.tversity.com)
- EBS UPnP: Embedded UPnP Server, Client and AV Stacks in ANSI-C for all embedded platforms (http://www.ebsembeddedsoftware.com/product_upnp_overview.htm)
- VoIP softphone that supports UPnP IGD (http://www.pacphone.com/)
- Cidero (http://www.cidero.com/) Java appliocation for controlling A/V - Devices (Server & Clients)

Retrieved from "http://en.wikipedia.org/wiki/Universal_Plug_and_Play"

Categories: Network protocols | Microsoft Windows
