# Making proper support for Universal Plug and Play in KDE

Armijn Hemel

August 14, 2008

# About me

Professional:

- 1996-2006: computer science at Utrecht University
- 2004-2006: MSc thesis: NixOS (`http://www.nixos.org/`)
- 2000-2008: author Linux Magazine NL, Linux Magazine UK, NetOpus, ...
- 2005-present: `gpl-violations.org`
- 2006-present: board member of NLUUG (`http://www.nluug.nl/`)
- 2006-present: Chief Random Projects at Loohuis Consulting

# A word from our sponsors: Loohuis Consulting

- specialized hosting
- web development (AJAX and other buzzwords)
- GPL license compliance
- UPnP security
- router/embedded security advice

More info: `http://www.loohuis-consulting.nl/`

# A word from our sponsors: NLUUG

- November 6 2008: conference about “mobility”
- November 6/7 2008: Embedded Linux Conference Europe
- November 8 2008: DLNA plugfest

All in Ede, the Netherlands

More info: `http://www.nluug.nl/`

# Today's topics and goals

- UPnP history
- UPnP protocol stack
- debunk common misconceptions about UPnP
- UPnP profiles
- how to fit this in KDE/free desktop

My research started with UPnP security testing. A lot of stuff is wrong and evil things can happen. Let's not make the same mistakes!

More info: `http://www.upnp-hacks.org/`

# Bringing UPnP to KDE/free desktop

There is definitely interest in having proper UPnP support:

- Amarok GSoC proposal (not accepted)
- bug 136385

A lot of efforts are done ad hoc, instead of in a framework. Not so good, since:

- duplication of code
- duplication of work
- UPnP seems simple, but there are nasty quirks you need to know about

Better to have a generic framework in KDE/free desktop:

- handles SOAP quirks of devices (lots!)
- handles eventing
- offers a clean interface to write UPnP control points

# NLnet sponsoring

NLnet has promised to sponsor UPnP support in KDE:

- 3 developer sprints
- a few devices

We need to set dates!

Today's session: brainstorm and trying to get you people thinking about what the best way to implement things is.

# Who should attend?

Anyone working on networked applications that:

- needs to be able to control port forwards
- needs to be able to browse networked media resources
- needs to negotiate WPA2 keys with a router
- needs to display remote user interfaces

Skills needed:

- (some) SOAP
- (some) XML
- TCP/IP
- application specific knowledge
- secure programming ;-)

Prime targets: Amarok, DragonPlayer, Kopete, NetworkManager

# Universal Plug and Play - introduction

Bring the desktop “plug and play” concept (Windows 98/Windows ME) to the (local) network.

Benefits:

- no configuration on the part of the user
- no installation of software, drivers, etcetera

UPnP is not unique:

- JINI (Sun Microsystems)
- IETF ZeroConf (Apple “Bonjour”, KDE, GNOME)

# History of UPnP

- early 1999 as reaction by Microsoft to Sun's JINI
- early 2000: first products with UPnP (Windows ME, Intel's Open Source UPnP SDK)
- Windows ME and Windows XP have UPnP support built-in since their release
- September 2007: ISO standard

UPnP organizations:

- UPnP Forum: create and publish new UPnP standards.
- UPnP Implementers Corporation: UPnP certification and logo licensing.

# UPnP protocol stack

0. addressing
1. discovery
2. description
3. control
4. eventing
5. presentation

# UPnP protocol - addressing

Zeroth, optional, step. If no DHCP server is found use "auto-addressing":

1. randomly pick an IP address from `169.254/16` IP range
2. if IP address is taken, abandon IP address and goto 1
3. else keep IP address

More auto-addressing:

- RFC 3927
- IETF ZeroConf
- Fedora (has a default route for `169.254/16`)

# UPnP protocol - discovery

First step: discover devices on the network

On boot-up send a HTTP header to UDP port 1900 on 239.255.255.250 (this is called HTTP-U):

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: ssdp:discover
MX: 10
ST: ssdp:all
```

Other UPnP devices should reply via UDP unicast:

```
HTTP/1.1 200 OK
CACHE-CONTROL:max-age=1800
EXT:
LOCATION:http://10.0.0.138:80/IGD.xml
SERVER:SpeedTouch 510 4.0.0.9.0 UPnP/1.0 (DG233B00011961)
ST:upnp:rootdevice
USN:uuid:UPnP-SpeedTouch510-1_00::upnp:rootdevice
```

# UPnP protocol - discovery (continued)

Periodically send notifications to 239.255.255.250 on port 1900 UDP:

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=180
Location: http://192.168.1.1:5431/dyndev/uuid:0014-bf09
NT: upnp:rootdevice
NTS: ssdp:alive
SERVER:LINUX/2.4 UPnP/1.0 BRCM400/1.0
USN: uuid:0014-bf09::upnp:rootdevice
```

# UPnP protocol - description

Second step: find out what devices can do

`LOCATION` points to XML:

```
Location: http://192.168.1.1:5431/dyndev/uuid:0014-bf09
```

This file describes (per "profile"):

- control URL
- events URL
- SCPD URL (description of which functions are available, in XML)

```
<service>
<serviceType>urn:schemas-upnp-org:service:
    WANIPConnection:1</serviceType>
<serviceId>urn:upnp-org:serviceId:WANIPConnection</serviceId>
<controlURL>/ipc</controlURL>
<eventSubURL>/ipc</eventSubURL>
<SCPDURL>/ipc.xml</SCPDURL>
</service>
```

# UPnP protocol - control

Third step: controlling a device

Devices can be controlled by sending SOAP requests to the “control URL”.

There is no authentication/authorization in UPnP, being on the LAN is enough to do this.

No administrative privileges needed: any user can do this.

# UPnP protocol - eventing

Fourth step: keeping devices informed

Changes in “state variables” are sent over the network to subscribed clients.

Clients can subscribe to events, if they provide one or more callback URLs.

# UPnP protocol - presentation

Fifth step: human interface

Presentation is the human controllable interface: the webinterface of the device.

# UPnP profiles

UPnP defines profiles: a set of actions, state variables and other profiles that implement specific functionality.

Standardized profiles:

- Internet Gateway Device (IGD)
- MediaServer and MediaRenderer (A/V)
- HVAC
- and more

Most popular: Internet Gateway Device and (recently) MediaServer and MediaRenderer.

For many people IGD is UPnP. This is not true. For others MediaServer and MediaRenderer is UPnP. This is not true either.

# Things that need to be done

- device discovery
- passing device state to programs that are interested (new devices joining, devices leaving, etcetera)
- creating a good interface that programs can use to send SOAP requests
- creating a service that can be used to process event callbacks (HTTP server) and send them to interested programs
- application support

# Example: Internet Gateway Device profile

The Internet Gateway Device (IGD) is an interesting first target:

- There are millions of routers that implement the UPnP IGD (most widespread use)
- Used actively by a lot of applications and other devices.

The Internet Gateway Device profile allows port forwarding.

Network Address Translation (NAT) does not easily work with predefined ports.

Workaround: programs dynamically agree on ports. Firewalls need to be dynamically adapted for this to work.

- MSN/Windows Live Messenger ("webcam", file transfers)
- remote assistance (Windows XP)
- X-Box
- many bittorrent clients

# Internet Gateway Device and KDE

Applications in KDE that would use this profile:

- Kopete (portmappings for webcam and file transfers in MSN)
- KTorrent (open extra ports)
- NetworkManager (statistics, connection control)

# MediaServer

MediaServer:

- stores media files
- lets applications browse and search files
- optionally streams

Applications in KDE that invoke actions on devices with this profile:

- Amarok
- DragonPlayer
- digiKam
- Dolphin (?)

# DLNA

"Digital Living Network Alliance" is an industry consortium.

Core technologies of DLNA:

- UPnP

(amongst others)

Lots of devices are "DLNA certified".

What does it mean? I don't know.

Can we be certified? Unlikely, since DRM is involved.

Would still be fun to test with certified devices (hey Nokia, the N95 8GB is DLNA certified!)

# Wi-Fi Protected Setup

New standard: Wi-Fi Protected Setup

UPnP is used for exchanging WPA2 keys over a wired connection. All new home routers you can buy at a shop will get support for this!

In Windows Vista this is called "Windows Connect Now", part of "Windows Rally".

`http://en.wikipedia.org/wiki/Wi-Fi_Protected_Setup`
`http://en.wikipedia.org/wiki/Windows_Rally`

Applications in KDE that invoke actions on devices with this profile:

- NetworkManager

# Other profiles

Other profiles (not used a lot, but devices exist)

- RemoteUIClient (NETGEAR EVA8000)
- WLAN Access Point
- Digital Security Camera
- Scanner

# Let's get to work! (soon)

I have:

- UPnP devices (quite a few routers, some mediaservers, some others).
- lots of knowledge about how particular stacks (mis)behave

You have:

- application knowledge
- a platform to leverage a lot of work
- the opportunity to make the best UPnP enabled desktop

Deal?