Design of a context aware multimedia management system for home environments

K. Mets, J. Nelis, D. Verslype, P. Leroux, W. Haerick, F. De Turck, C. Develder
Ghent University - IBBT,
Dept. of Information Technology - IBCN, Ghent, Belgium
Email: chris.develder@intec.ugent.be

***Abstract*—We present the design, implementation and performance analysis of a multimedia management system for home environments. It enables comprehensive access and control of media that is distributed over multiple devices. Based on automatic discovery of the media servers and renderers (i.e. audio or display devices) through UPnP, our application integrates all content in a single view (merging content and/or removing duplicate items). The application is tailored to a mobile platform and features a context aware platform to automatically select the nearest display device. Adaptive content filtering can be applied to restrict the list to content supported by this device. Performance measurements on a proof of concept implementation are presented.**

***Keywords*-UPnP, service discovery, context awareness, QoS, home networks, multimedia management**

I. Introduction

The home network complexity increases steadily, in terms of both network technologies and devices, as well as the services they offer. Home network technologies can be quite heterogeneous, covering various wired (e.g., ethernet, coax, powerline) and wireless technologies. Also the devices are diverse in terms of capabilities and functionality (storage, processing, user interface, screen resolution, etc.). However, the user wants to be relieved of any troublesome configuration and network management. To this end, remote management approaches have been devised [1], for example using dynamic service life cycle management features of OSGi [2] and remote management as enabled by TR-069 [3].

For services offered by devices within the home, which are typically not under the control of a service provider, so-called service discovery protocols have been devised. One of the most well-known and widespread examples is UPnP (Universal Plug-and-Play, [4]). It solves the problem of automatically detecting new devices and invoking the services they offer, and will be discussed in more detail in the following section II.

Within the UPnP Forum, a specific architecture has been devised to support multimedia applications: the UPnP AV architecture [5]. Media is offered by so-called media servers and can be played out on so-called media renderers. When a user's home network has multiple servers and renderers, it becomes troublesome to locate the correct server which contains particular content, and trial-and-error could be necessary to find a suitable media renderer, e.g. due to the availability of codecs or support for transport protocols.

This paper addresses the latter problems by proposing a multimedia management application, which offers the user an integrated view on the whole media collection (unifying content available on all servers), and assisting the user in selecting the most convenient media renderer through a context aware platform. Examples of context awareness are the location and possibilities of the media renderers.

The remainder of this paper is structured as follows: in Section II we present the current state of the art and related work on UPnP. Subsequently, we present the architecture of our context aware multimedia management system in Section III. This has been implemented in a proof of concept setup as outlined in IV, featuring an implementation on a mobile smartphone. Section V evaluates the performance of this setup. Final conclusions are summarized in Section VI. Future work is discussed in Section VII.

II. Universal Plug and Play (UPnP)

The heterogeneous network environments, comprising both wired and wireless devices, and a mix of multiple services offered by various devices have created complex network environments. These bring new challenges with respect to service management and particularly service discovery. The latter should be automated as much as possible, and avoid the need for manual configuration. Therefore, so-called service discovery protocols have been devised. In the home network context, the UPnP approach, comprising the Simple Service Discovery Protocol (SSDP), has emerged as one of the most prevalent standards. (Note that the issue of crossing boundaries between different service discovery protocols [6], or even across home networks, is not the scope of this paper.)

In a nutshell, the UPnP architecture comprises a set of open protocols based on established TCP/IP standards, and includes the following capabilities [7]:

- *Service Discovery*: The Simple Service Discovery Protocol (SSDP) uses UDP unicast and multicast packets to advertise a device's available services. This allows a device to advertise its services to so-called UPnP control points, and vice versa control points can use SSDP to search for devices/services of interest.
- *Description*: The detailed information on a particular device has to be retrieved from a URL provided in discovery messages. Apart from device information

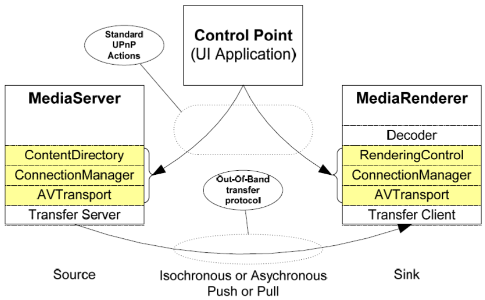


Figure 1. The UPnP-AV architecture: the 3-box model.[5]

(e.g. manufacturer), the service description (in XML) lists the exact service interface in terms of commands (so-called actions) and the respective parameters (arguments).

- *Control*: The actual service invocation is performed using control messages (also in XML) using the Simple Object Access Protocol (SOAP).
- *Event Notification*: The state of a device and its associated services can be modeled by state variables. The service can publish updates to interested parties (i.e. control points) when these variables change. This is achieved by the GENA protocol (General Event Notification Architecture).

The focus of this paper is on providing the user a comprehensive overview of (audiovisual) media available across the heterogeneous home network. In this context, we will focus mainly on the UPnP-AV architecture [5], which defines three interacting entities: the Control Point, the source of the media content (called the MediaServer, MS), and the sink for the content (called the MediaRenderer, MR) as illustrated in Fig. 1.

As indicated on Fig. 1, the actual streaming of the A/V content happens out-of-band with respect to UPnP-AV. In a heterogeneous home network, there is need for QoS mechanisms to ensure a stream has sufficient bandwidth available. In the recently standardised version 3 of UPnP-QoS [8] (to which we actively contributed), such parametric QoS reservations are supported.

One limitation of UPnP is that its operation is limited to a single home network (single subnet, cf. multicast messages in SSDP). To extend the service reach beyond the home network, UPnP remote access (UPnP-RA) standardised ways to interconnect multiple UPnP domains [4]. Alternate solutions, enabling ubiquitous access to mobile users have been proposed as well [6]. This is however out-of-scope for the current paper.

In this work we will mainly focus on UPnP-AV as a basis to offer an advanced multimedia management system. The current UPnP-AV approach only provides some basic building blocks. The following high level features are not part of the AV framework:

1) *Media integration*: Media servers (MS) offer different views on the content they provide. If multiple MS are present, they may contain duplicate items, or complementary ones (e.g. different tracks of the same album). The framework does not offer an integrated view on all media.
2) *Renderer selection*: When multiple media renderers (MRs) are present, the selection of a particular renderer is the user's responsability.
3) *Format detection and filtering*: The media formats of media present in a MS collection can be conflicting with the capabilities of a Media Renderer (MR). The Search() action on a MS provides a way to filter out particular media types a MR supports, but since this action is optional only a limited subset of Media Servers actually support this.
4) *Format adaptation or transcoding*: When media formats conflict with a particular MR's capabilities, or not enough network resources are available, it would be useful to adapt/transcode the stream on-the-fly (as enabled by e.g. Scalable Video Coding, demonstrated by [9]).

In the media management system described below, we will address the issues 1-3. Issue 4 is subject to future work.

III. Advanced Multimedia Management System

The software architecture of our proposed advanced multimedia management system for home environments is shown in Fig. 2. The modules can be roughly divided over the three functional domains, as explained in the following subsections.

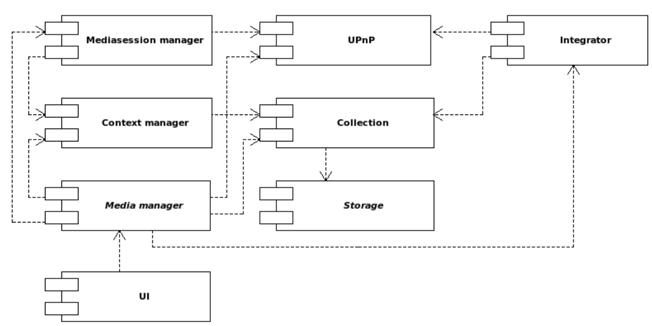


Figure 2. Module view of the advanced multimedia management system for home environments.

The multimedia management system discussed below can be complemented with the aforementioned remote access approaches, or with automatic media adaptation.

A. Media metadata integration

The first subset of modules realizes the media integration, aggregating data on the media as offered by all media servers discovered through UPnP. The *Collection* module comprises the unified collection data. This unified collection is built by the *Integrator* module. The latter is responsible for retrieving metadata from media servers (via the appropriate UPnP-AV service invocations) and integrating this metadata into the unified media collection view.

The first step after retrieving the metadata is preprocessing. This step removes redundant information and transforms the metadata into the internal format of the management system. Subsequently, the integrated media collection view is composed by sequentially adding each and every media item. The actual integration algorithm comprises three phases, executed on each consecutive media item addition to the collection:

1) *Duplicate search*: The first step after preprocessing is a fast duplicate search looking for exact matches, or simple permutations of e.g. artist and song title;
2) *Candidate set*: If the first step did not reveal an immediate duplicate, pattern matching techniques are used to find likely matches with items already present in the integrated collection. (This step will accommodate for small variations in metadata, e.g. because of misspellings.)
3) *Candidate comparison*: The last step will determine which item of the candidate set can be considered a duplicate of the new item. If so, both locations for this particular media item will be stored in the database structure for the integrated media collection.

B. Context awareness

In order to present the unified collection in an intelligent manner to the user, context awareness is integrated into the multimedia management system, addressing the issues of renderer selection and media filtering. To this end, two context awareness aspects are considered: location and device capabilities. Based on the location of the user, the system can decide which media renderer is most likely to be targetted by the user, e.g. the device that is nearby. The capabilities of the selected device (automatically based on location, or manual overridden by the user) are compared to the unified collection and only compatible multimedia items are shown. This functionality is provided by the *Context Manager* which is responsible for gathering device and location information.

The device information is gathered by using the GetProtocolInfo() of the MediaRenderer device. This returns the list of transport protocol/codec combinations that are supported. Using this information, content can be filtered according to the device's capabilities. There are a couple of ways to do this filtering, GetProtocolInfo() on the MediaServer device returns a list of transport protocol/codec combinations this device offers. If none of the combinations match, the MediaServer device at hand can be ignored for this MediaRenderer device. If some combinations match and the MediaServer device implements the Search() action the multimedia management system can search for the relevant combinations. If the Search() action is not implemented, the Browse() action has to be used and content metadata has to be investigated.

In order to discover which media renderers are located in the same (part of the) room as the user, an accurate indoor positioning system is required. As a GPS signal cannot be used indoor, a lot of research is put into the use of Wifi access points and wireless sensor networks for indoor positioning. The work that is presented in this paper, interfaces with an indoor positioning system that is currently developed at our research group. Without zooming into the details of this positioning framework, this coupling will be shortly described. The device on which the multimedia management system is running, is equipped with an IEEE 802.15.4 radio (e.g. Tmote Invent). This active node sends beacons at regular intervals. The infrastructure nodes (typically at least one node per room) collect the information contained in the beacons and forward measurement reports to a central component, the positioning server. This server receives all the measurement reports from the infrastructure nodes and calculates the user's position. The multimedia management system can then retrieve its position by accessing the positioning server through a web service interface. Once the multimedia management system knows its position, it can check which renders are in its direct neighbourhood. Currently, it is supposed that the media renderers have a fixed location that is initially registered in the system. However, when the renderers itself would also be equipped with a radio module, they could be integrated in the positioning system and automatically inform the system when they are moved. Note that as web service calls are used, there is a strict separation between the positioning framework and the media management framework.

C. Session management

As outlined above, multiple media renderers can coexist in a home environment. The advanced multimedia management system associates a media session with each available media renderer. A media session consists of a playlist containing media items of which the playback on the media renderer can be controlled.

The playlist items also include information on the resources containing a particular instance of the associated media item. Obviously, these resources can refer to different Media Servers and hence a media session can be responsible for the control of several media servers. It is also possible that properties such as quality, transport method, etc. differ between resources. A media session is responsible for selecting a compatible resource for playback.

IV. Proof of concept implementation

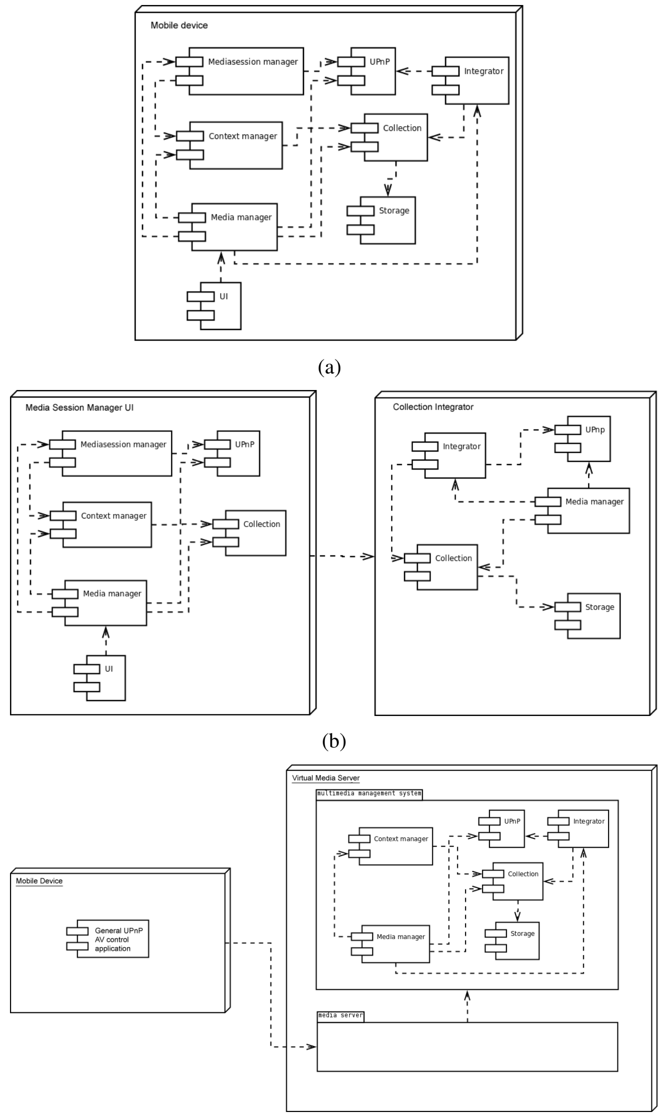
As discussed above, the multimedia management system architecture is modular. Hence, to deploy the media management system, various choices can be made as to where to instantiate each module. Three approaches are depicted in Fig. 3:

- *Monolithic*: The conceptually simplest choice is to have all components integrated on a single device (Fig. 3(a)). This means a single standalone application needs to be installed.
- *Distributed*: The components taking care of compiling the integrated view on the entire media collection can be split off in a separate 'Collection Integrator' application. The actual user interface and control of media playout are separated into a 'Media Session Manager UI'.
- *Virtual media server*: The multimedia management system can behave itself like a UPnP AV Media Server. This way, a mobile device can for example use an existing general AV control application to access the collection.

If multiple devices are present which all can be used as a control point and hence each run an instance of a multimedia management system, the (relatively time consuming) integration into a unified view of the media collection will be executed multiple times when each of the devices uses a *Monolithic* media management system. To avoid overhead (e.g. many identical UPnP calls to the Media Server devices), the *Distributed* approach will be obviously beneficial. In addition, the mobile device can be relieved by locating the 'Collection Integrator' on a more powerful system (e.g. PC or laptop). Note that the *Distributed* approach implies that the Collection module on the 'Media Session Manager' will be a stub, interfacing with the actual integrated collection as maintained by the 'Collection Integrator'.

The last *Virtual Media Server* solution has the advantage that a mobile device can use standard UPnP AV control software. However, the context awareness features with respect to location discovery are not present. Also, the user needs to be aware of which media server is the virtual one representing an integrated view on all media.

The advanced media management system has been implemented and tested on both a mobile device (Nokia N800) and a laptop (iBook G4), as shown in Fig. 4. These are the setups used for the performance measurements discussed next.



(c)

Figure 3. Deployment alternatives for the advanced multimedia management system: (a) Monolithic system, (b) Distributed, (c) Virtual Media Server.

V. Performance assessment

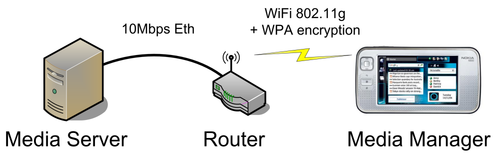
We assessed the performance of the media management system, and in particular the integration of media. This was done by running the *Monolithic* deployment of Fig. 3(a) on the Media Manager device of the setups outlined in Fig. 4. The Media Server machine was used to run open source UPnP media server software (Twonkymedia [10]), populated with actual audio and video data.

A. Integration time for collections without duplicates

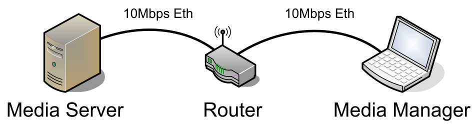
The first experiment we performed was aimed at measuring the time needed to integrate the metadata as retrieved from a collection without any duplicates. The results are summarized in Fig. 5.

The total integration time measured is divided into three constituents:

1) *Integration* is the time for running the integration algorithm as explained before;
2) *Storage* is the time spent for storage operations in the database structure, maintaining all collected media data;
3) *Other* is the remaining time, which is dominated by

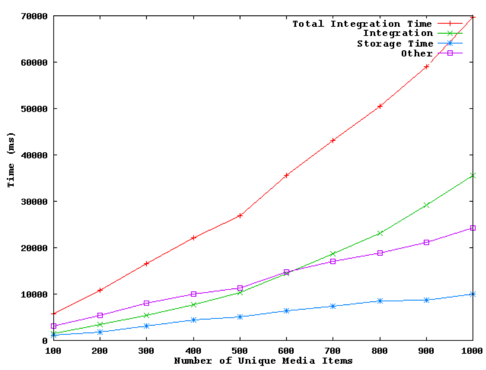


(a)

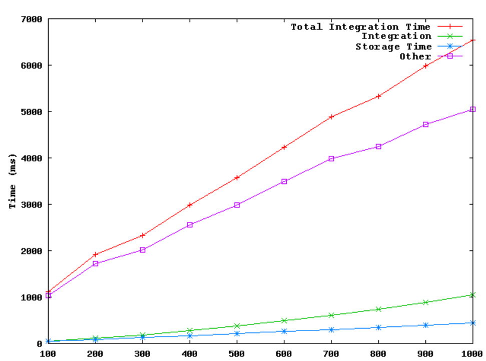


(b)

Figure 4. Proof of concept demonstration setups: (a) Wireless, (b) Wired. The Nokia N800 Internet Tablet (running Internet Tablet OS edition 2008) has 128MB RAM and uses a 320 MHz ARM processor. The server had an Intel Core 2 Quad 2.40 GHz processor and 2GB of RAM. The laptop is an iBook G4, with a 1.2GHz PowerPC G4 processor and 768MB RAM. The router is a D-Link DI-264+.

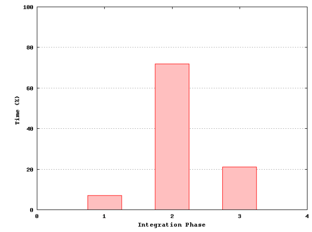


(a)

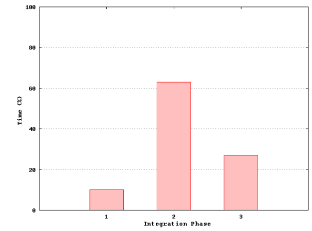


(b)

Figure 5. Measurements of integration times for a collection without duplicate media items for both proof of concept demonstration setups: (a) Wireless, (b) Wired. Note the different Y-axis scales for both graphs.



(a)



(b)

Figure 6. Integration phase times (1 = Duplicate Search, 2 = Candidate Set, 3 = Candidate Comparison; see Section III-A), for a collection without duplicate media items in both proof of concept demonstration setups: (a) Wireless, (b) Wired.

the UPnP communication between our management system and the UPnP Media Server offering the imported media items.

The results show that running the algorithm on the mobile device is significantly more time consuming (factor 10 difference) than using the wired laptop. This indicates that the *Monolithic* deployment may not lead to acceptable performance for large collections. Looking at the different constituents in the mobile case, we note that UPnP communication (*Other*) amounts to 56%–35% of the total time, whereas database access (*Storage*) is responible for around 18%. In the wired PC case, we note that UPnP communication dominates the overall much smaller total integration time.

B. Integration phases

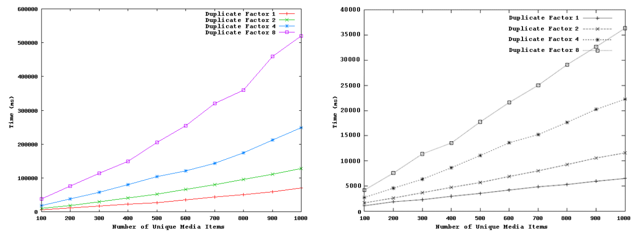
As explained before, the integration algorithm itself comprises three phases. Fig 6 shows there relative contribution to the *Integration* time discussed previously. As expected, the bulk of integration time is spent in the pattern matching search for candidates to merge with (i.e. likely duplicates).

C. Influence of duplicates

The last experiment measured the impact of duplicate items in the collection on total integration time. The results are summarized in Fig. 7(b), and clearly show a (slightly less than linear) increase of the total integration times with increasing duplicate factors.

This indicates that the multiple occurrences of a single media item in the collection exposed by a typical Media Server (e.g. once in file structure based on Artist, once in an Album-oriented structure, etc.) will have a performance penalty on the media management system. Optimisation opportunities hence arise, based on interpretation of the file structure (and automatically skipping duplicate parts instead of integrating their items again).

Comparing the exectution times on a mobile device with those on a wired laptop, we note unacceptably high times for the mobile case. This confirms that the *Monolithic* approach is not optimal.



(a) (b)

Figure 7. Measurements of total integration times for a collection with duplicate media items for both proof of concept demonstration setups: (a) Wireless, (b) Wired. Note the different Y-axis scales for both graphs.

VI. Conclusions

Home networks are increasingly heterogeneous, both in terms of network technologies and devices, as well as the services they offer. To relieve the end user from manual configuration efforts, service discovery protocols have been devised. The most prevalent one in home network environments is UPnP. For the case of multimedia management–which this paper focused on–the relevant framework hence is that of UPnP-AV. Yet, some open issues arise in the current state of the art. Our proposed multimedia management system enables an integrated view of all discovered media (*Media integration*), automatic playout device selection based on location awareness (*Renderer selection*), and offering automatic *format detection and filtering* based on the capabilities of the selected renderer. Measurements on a proof of concept implementation are provided, resulting in some guidelines for deployment approaches of the proposed advanced multimedia management system.

VII. Future work

The designed context aware multimedia management system can be extended further to include support for Quality of Service. Future version of the management system can integrate UPnP-QoS [8] control point functionality to ensure end-to-end QoS in the home network (possibly as part of a dynamic service management platform [11] we developed earlier).

Integration with online multimedia services can benefit the correctness of information in the unified content database of the multimedia management system. Another extension is the support for content transcoding to enable users to consume more content.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7) under project 212 352 ALPHA "Architectures for fLexible Photonic Home and Access networks." C. Develder is supported by the Research Foundation – Flanders (FWO) as a post-doctoral fellow.

References

[1] W. Haerick, N. Goeminne, K. Cauwel, G. De Jans, F. De Turck, B. Dhoedt, P. Demeester, S. Bracke, W. Acke, and C. Bouchat, "Success in home service deployment: zero-touch or chaos?" *J. Communications Netw.*, vol. 4, pp. 36–43, Jul. 2005.

[2] R. S. Hall and H. Cervantes, "An OSGi implementation and experience report," in *Proc. 1st IEEE Consumer Commun. and Networking Conf. (CCNC 2004)*, 5–8 Jan. 2004, pp. 394–399.

[3] B. Forum, "TR-069 cpe wan management protocol v1.1," December 2007. [Online]. Available: http://www.broadband-forum.org/technical/download/TR-069Amendment2.pdf

[4] "UPnP forum." [Online]. Available: www.upnp.org

[5] J. Ritchie, T. Kühnel, J. Kang, and W. van der Beek, "UPnP AV Architecture:1," 30 Sep. 2008. [Online]. Available: http://www.upnp.org/specs/av/default.asp

[6] P.-G. Raverdy, V. Issarny, R. Chibout, and A. de La Chapelle, "A multi-protocol approach to service discovery and access in pervasive environments," in *Proc. 3rd Int. Conf. on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS 2006)*, San Jose, CA, USA, 17-21 Jul. 2006.

[7] A. Presser *et al.*, "UPnP Device Architecture," 15 Oct. 2008. [Online]. Available: http://www.upnp.org/resources/documents.asp

[8] UPnP Forum, "UPnP QoS." [Online]. Available: http://www.upnp.org/specs/qos

[9] C. Develder, P. Lambert, W. Van Lancker, S. Moens, R. Van de Walle, J. Nelis, D. Verslype, S. Latré, N. Staelens, B. Vermeulen, B. Masschelein, T. Van Leeuwen, J.-F. Macq, K. Struyve, F. De Turck, and B. Dhoedt, "Delivering scalable video with QoS to the home," 2009, submitted to Telecommunication Systems, Special Issue on the Quality of Experience issues in Multimedia Provision.

[10] "Twonkymedia." [Online]. Available: http://twonkymedia.com

[11] N. Goeminne, K. Cauwel, F. De Turck, and B. Dhoedt, "Deploying QoS sensitive services in OSGi enabled home networks based on UPnP," in *Proc. Int. Conf. on Internet Computing (ICOMP2006)*, Las Vegas, NV, USA, 26-29 Jun. 2006.