Integrating Personal Media and Digital TV with QoS Guarantees using Virtualized Set-Top Boxes: Architecture and Performance Measurements

Bert Vankeirsbilck*, Jelle Nelis*, Dieter Verslype*, Chris Develder*, Tom Van Leeuwen† and Bart Dhoedt*

†Alcatel - Lucent
Copernicuslaan 50, 2018 Antwerp, Belgium

*Ghent University, Department of Information Technology (INTEC)
Broadband Communication Networks (IBCN)
Gaston Crommenlaan 8, bus 201, 9050 Ghent, Belgium
Email: Bert.Vankeirsbilck@intec.ugent.be

***Abstract*—Nowadays, users consume a lot of functionality in their home coming from a service provider located in the Internet. While the home network is typically shielded off as much as possible from the 'outside world', the supplied services could be greatly extended if it was possible to use local information. In this article, an extended service is presented that integrates the user's multimedia content, scattered over multiple devices in the home network, into the Electronic Program Guide (EPG) of the Digital TV. We propose to virtualize the set-top box, by migrating all functionality except user interfacing to the service provider infrastructure. The media in the home network is discovered through standard Universal Plug and Play (UPnP), of which the QoS functionality is exploited to ensure high quality playback over the home network, that basically is out of the control of the service provider. The performance of the subsystems are analysed.**

I. Introduction

Current households are getting more and more decorated with all kinds of dedicated devices and machines, ranging from mobile phones over dishwashers to digital TV Set-Top Boxes (STB). Some are rather personal, others are shared with the family, but the common evolution of all devices is their tendency to be networked and their ability to communicate. This evolution is fueled by the Digital Living Network Alliance (DLNA), having an explicit mission statement to facilitate *a global collaboration of 245 of your most trusted brands, all working together to help you create the home entertainment environment you've always imagined* [1]. In this paper, we propose an architecture that offers not only flexible service and application management to a service provider, but also enables integration and interworking with the user's personal devices within the home. Particularly, we demonstrate how to control media services available in the user's home from within the Digital TV Electronic Program Guide (EPG).

In this case, the Digital Television provider is an important stakeholder having control of key components in the system. The Digital Television providers usually supply the set-top boxes (STBs) that are installed in the home environment. For obvious cost reasons (both production and maintenance related), these mass produced user devices should be simple, maintenance-free, cheap and robust, while keeping the functional possibilities as experienced by the end-user as broad as possible.

This motivates our idea to apply a server centered approach to control in-home media content from within the EPG. In this approach, the STB is kept as lightweight as possible: it only intercepts user actions (e.g. remote control button presses) and shows content on the television screen. The remaining functionality needed to integrate both broadcast content and personal, locally stored content is shifted into the network, i.e. away from the STB. This way, important software components that would normally execute on the STB are now running on a remote server. This optimizes the STB lifetime and keeps the device as simple and robust as possible.

The advantages of this approach are manifold. The STB will be very simple leading to a decreased need for maintenance, reduced energy consumption, less hardware faults and lower cost of STB (both hardware and service cost). Furthermore, since the functionality resides on a provider controlled server, the ease of upgrading and changing this functionality is increased because it does not involve the STB itself. There is also an opportunity to make the STB more generic so that it can connect to other services in both the home network and the Internet. For the user, this could mean that he has one single piece of equipment to control all his devices, present in-home and even beyond.

In this paper, we compare the proposed server centered approach to the other extreme where all functionality for merging broadcast and in-home video content resides at the STB. Universal Plug and Play Audio and Video (UPnP AV) will be used to discover locally stored content. In order to guarantee qualitatively appealing delivery of the in-home media content, we present the possibilitie to use UPnP-QoS to reserve resources in the home network.

The contribution of this paper is threefold: (i) we apply the

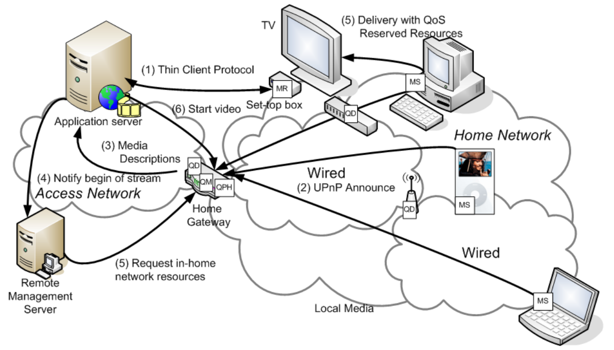


Fig. 1. Server centered architecture for personalized service delivery with quality assurance

thin client paradigm to a digital TV STB, (ii) we propose a flexible integration of local (home) media in the services it delivers, and (iii) we propose a QoS reservation mechanism, including performance results of the first implementation of UPnP QoS v3. Earlier work on providing QoS in the home did either not include the parametrized QoS reservations [2], [3] or were limited to in-home scenarios [4]. Other QoS approaches did not support the plug-and-play flexibility of UPnP and were mainly layer-2 oriented [5], [6]. To the best of our knowledge, we are the first to provide an elaborated architecture for STBs that is thin-client based, although our work is motivated by [7], since we can overcome issues stemming from limited processing resources in residential devices.

This paper is structured as follows. In section II, an architecture is proposed to integrate personal content into an EPG in a qualitative way, using a server centered approach. In section III, the ideas and the feasibility of the virtualized STB approach are discussed using experimental results. Section IV presents methods and results of UPnP QoS reservations to be made in the Home Network. Finally, section V concludes this article.

II. Architecture

We propose an architecture that combines provider side application logic with services already located in the Home Network to enable a service provider to deliver qualitative services. Fig. 1 presents the architecture for the proposed server centered approach. Application logic is moved away from the STB towards the provider's network so the STB can be a thin client which means it will be less prone to replacement. Using UPnP-QoS and the remote management solution the service provider already uses, resources are reserved throughout both the Access and Home Network for the services offered by the service provider.

The home network comprises all devices that can be interconnected in the user's home environment. These include personal devices such as multimedia players, telephones, personal computers, laptops, handhelds or next generation white goods, as well as devices rented from an external party, such as a STB or a home gateway (HGW). The access network is the connection to services outside the home environment.

We will focus on a service that integrates in-home multimedia content to form a unified EPG. The local media resides on devices that are connected to the home network, such as a laptop, PC or a personal media player. The HGW provides the link to the outside world. It is able to discover the in-home devices and to communicate descriptions of their content to external services. In the home network, the STB only takes care of interaction with the user. The unified EPG is constructed on a server in the access network, and is able to merge broadcast television content with the description of the user's personal content acquired through the HGW. The concept of a unified EPG has been introduced in [8].

The labels in Fig. 1 represent the communication technologies between the parties in the architecture: (1) The communication between STB and application server adheres to the principles of thin client computing: user interaction is sent over the network to the application server, and graphical updates of the EPG are transmitted the other way around. (2) The media servers (e.g. laptop) advertise their shared content into the home network by using UPnP AV. (3) The HGW is able to aggregate and communicate the metadata of in-home shared content out of the home network. As will be shown further in this paper, there is a need to limit or compress the information that is sent uplink into the access network. (4) The application server integrates the metadata of the personal content into the EPG, and on selection of playback of an in-home media element, the Remote Management Server (RMS) is triggered to prepare both the access and the home network for the upcoming service stream. (5) Using a remote management protocol, the RMS instructs the HGW to reserve resources within the home network using UPnP-QoS. To enable this, network elements that are able to reserve network resources shall have to implement UPnP-QoS by way of the QosDevice service. The local QoS management framework on the HGW will use all information it retrieves from the home network to handle the QoS request, but will not, by itself, be responsible for individual resource reservation. (6) The application server will setup the video stream.

III. Integration of Local Content

We have deployed a realistic digital TV testbed according to the architecture presented in the previous section. On the HGW, we have implemented support for UPnP AV [9] allowing dynamic discovery and use of multimedia devices in the home network, allowing the use of Windows Media Player on a laptop as an in-home media server. The EPG Server and HGW communicate through webservices. As thin client protocol, we adapted a Virtual Network Computing (VNC) server and viewer pair [10], in order to merge client side rendered video with server rendered content. This adaptation required changes to both client and server code. The VNC viewer utilizes a recent FFmpeg library for decoding video [11]. We have implemented a custom EPG in Java in conformity with XML-TV [12], a standard format for describing program guides. In order to simulate the uplink bandwidth in the access network, we used a Click Router [13] that applied simple

TABLE I
TESTBED HARDWARE AND SOFTWARE DETAILS

Set-Top Box	AMD Athlon 64 2000 MHz, 2GB RAM DDR1 Ubuntu 9.10 Adapted tightVNC v1.3.9 FFmpeg v. SVN-r20647 (2009-11-06)
EPG Server	AMD Athlon 64 2100 MHz, 2.48 GB RAM Ubuntu 9.10 Adapted tight VNC v1.3.9 Custom EPG using Java and XML-TV
Media Server	Intel Core Duo P8400 2.26 GHz, 3.45 GB RAM Windows Media Player 11 (UPnP AV compliant)
PC Home Gateway	Athlon 64 2100 MHz, 1GB RAM UPnP AV compliant WebService interface to EPG Server
ALIX Home Gateway	Alix3d2 500 MHz, 256 MB RAM Contains same software as PC Home Gateway
Click Router	AMD Athlon 64 2000Mhz, 960 MB RAM Click Modular Router
Network cards	All networks cards used: 100Mbps

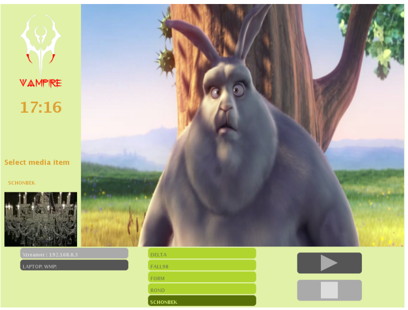


Fig. 2. Screenshot of the custom made unified EPG

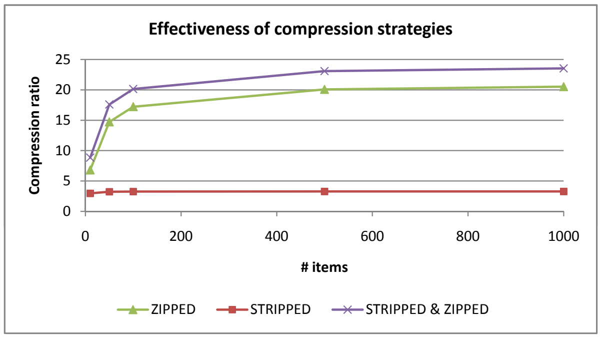


Fig. 3. Effectiveness of compression strategies

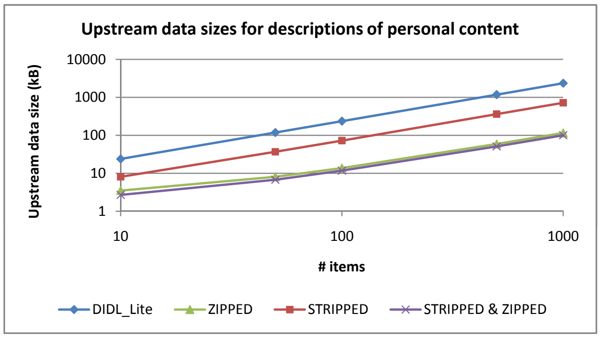


Fig. 4. Upstream bandwidth for browsing personal content

bandwidth shaping. The details of the experimental testbed are summarized in Table I.

Fig. 2 shows how our unified EPG integrates broadcasted video and personal content. The top right video is the broadcast channel or personal content the user is currently watching. Meanwhile, the bottom menu can contain the controls to change broadcast channel or to browse the in-home devices and their content. Fig. 2 shows the latter option. In the left banner, the thumbnail picture of the selected personal video is shown. Through an experimental evaluation, we will compare our server centered approach to a STB centered approach. For this comparison, important performance metrics are discussed in the following subsections. First we tackle the uplink bandwidth that is consumed for transferring the description of the in-home media, to integrate its content into the EPG. Then, we evaluate the delay a user can expect between invoking a browse operation through the remote control and having the results visible on his screen.

A. Compression of personal content metadata

Regarding the bandwidth used to transfer a description of the personal audio/video content, we have investigated the possibilities and effect of both lossy and lossless compression of the DIDL-Lite standard. This standard is derived from a subset of DIDL, the Digital Item Description Language, recently developed within ISO/MPEG21 [14], [15]. Concerning the lossless compression we applied the GZIP algorithm from the java.util.zip library. For lossy compression, the HGW parses the DIDL-Lite formatted description retrieved from the in-home Media Server, and strips it down to only retain the description entries that are supported by the EPG. Although the lossy compression approach might seem to be aimed to restrict the functionality, this mechanism could bring an important advantage of increasing privacy about the personal content. The attributes that were kept for the description are: the item title, media type (audio, video, container), resources in the form of URLs and thumbnails in the form of UPnP:albumArt. Measurements with the combination of both kinds of compression were also conducted.

Fig. 3 presents the effectiveness of the proposed compression strategies. Applying lossy stripping of unneeded description attributes reduces the transferred data size with a constant factor of 3.25. The lossless GZIP algorithm is known to be efficient for textual data, and shows to compress better for larger data sizes following a logarithmic function. In our tests, the GZIP compressor was able to take profit of the repetitive structure of the data and reached a compression factor of 20.5 for 1000 item descriptions. Combining both the lossy and lossless compression techniques, the total size reduction factor further increased to 23.5.

Fig. 4 proves the need for the compression strategies. Using the full DIDL-Lite description would be undesirable from an uplink bandwidth perspective. The required bandwidth increases linearly with the number of items of which the

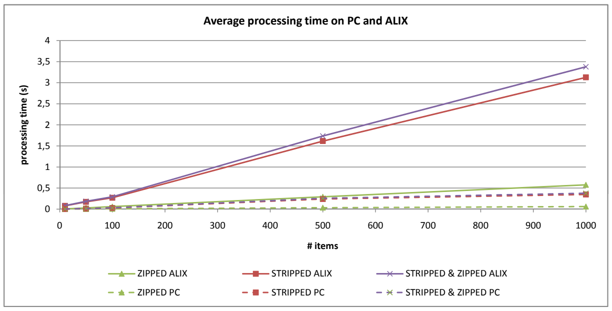


Fig. 5. Time needed for compression in Home Gateway, for two significantly different machines

description needs to be transferred. Fig. 4 shows that the description of 1000 items takes 2.4 MB to be transferred from the HGW to the remote server. Stripping the description attributes compacts the size less than zipping the contents. Combining stripping with zipping slightly increases the compression ratio of zipping the original DIDL-Lite description. Ultimately, transmission of the descriptions of 1000 items from the HGW to the server that builds the EPG requires only 100 kB. This proves that for the user, uplink bandwidth limits will not be an issue for applying a server centered approach. From the service provider perspective, the incoming link capacity will not be an issue either.

B. Delay overhead of virtualizing the set-top box

From the user's perspective, the responsiveness of the EPG is of key importance. We express responsiveness in function of the delay between the registration of a user action on the STB and the presentation of the corresponding graphical update. In this subsection we discuss important parameters that influence this total delay.

1) Home Gateway Compression Delay: The presented compression techniques can be applied to reduce the uplink network delay, but this will come at the cost of a compression delay on the HGW. Fig. 5 presents the delay introduced by applying the aforementioned compression mechanisms at the HGW. We have experimented with two machines to play the role of HGW, an ordinary PC and a resource limited ALIX board resembling a realistic scenario. It is shown that on either machine, zipping works very fast in comparison to the stripping method. This is mainly because for stripping, the XML metadata need to be parsed, and all elements need to be processed and rewritten. The zipping algorithm works on a byte level. It does not need any knowledge of the semantic content of the descriptions and thus outperforms the other algorithm. In our setup, the resource constrained ALIX board operates a factor of 10 slower than the PC based HGW. Even on this 'slow' device, the efficient zipping method takes only 577 ms to compress the list of description of 1000 items, for 10 items at a time 7 ms suffice.

2) UPnP Browse at Media Server: An important factor in the total delay is the one caused by the creation of the list of media item descriptions, which is handled by the media server. Fig. 6 presents the time it takes for the HGW to invoke a UPnP Browse on a UPnP AV compliant Windows Media Player 11 to compile a list of descriptions of media items in its playlist. The actual time to create the list of media items is the same for either HGW, but could not be measured directly. We could only measure this at the HGW, and that processing of the result is a part of the measurements. We have done the experiments with both the ALIX and PC HGW. Fig. 6 shows that in both cases, the elapsed computation time increases linearly with the number of items requested. When requesting the descriptions of 10 items from the ALIX HGW, the list is completely available after 433 ms. When requesting 1000 items at once, the delay runs up to 7.2 s. For the PC version of the HGW, the processing happens much faster, leading to 284 ms and 4.7 s respectively. The number of items requested are entirely decided upon by the application programmer that designs the EPG that integrates local content metadata. Using the figures presented here, this programmer could opt for preloading a number of items to increase the interactivity of the application. However, attention must be drawn to the fact that loading in larger groups appears to decrease overhead. For instance, loading 1000 items in groups of 10 with the ALIX HGW would take 100 times 433 ms, making 43 s in total. The advantage of such a grouping approach is that the first descriptions are compiled at the media server after 433 ms. When they would have loaded all at once, the application would have to wait for the complete list of descriptions to be available after 7.2 s.

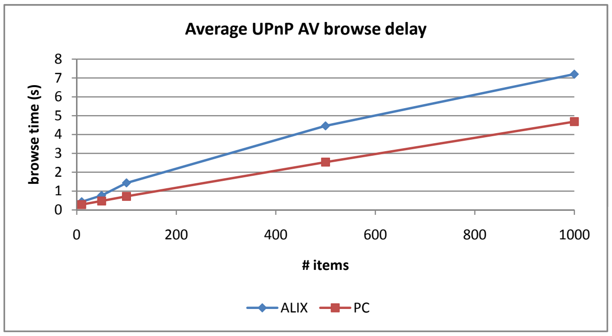


Fig. 6. Delay for browsing media items from Windows Media Player 11 using DLNA

3) Thin Client Protocol: Applying the thin client computing paradigm introduces extra delay between the application and the user interface. In [16], the authors empirically found that the delay of thin client protocols depends on the content to be sent, and state that in reasonable downlink bandwidth networks (1 Mbps) the order of magnitude is 100 ms or less. This is a constant extra delay, independent of the number of items to be displayed.

4) Total Delay: Fig. 7 shows the total delay that can be expected when summing all factors that were discussed before. For this test, the realistic ALIX HGW was used, and the uplink speed in the Access Network was varied from 256 Kbps to 100 Mbps. Remark that for this experiment, the zipping

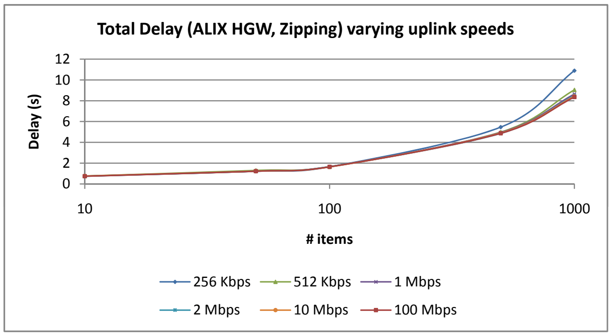


Fig. 7. Total delay for browsing media items in the virtual STB scenario

algorithm was applied but there was no preloading or paging active, so that requesting a certain number of items was always done as one request, resulting in transmission of all metadata in one response. One of the big differences between the virtualized STB approach and the original STB centered approach is that the descriptions of the media items need to be transferred to the EPG server to merge them into the EPG. The time needed to send this metadata from the HGW to the EPG server will depend on the number of bytes to transfer, and the actual bitrate available in the upstream direction in the access network. However, Fig. 7 shows that in this setting, due to the efficient compression technique, the slowest uplink network speeds (i.e. 256 and 512 Kbps) only start to play a significant role from 500 items on. This means that in this setup, for less items, the network is not the bottleneck. Fig. 7 shows that for displaying 100 items on the television screen, the total delay sums to 1.664 s. As explained, for this amount of items to be fetched, the uplink network speed has a negligible effect. We must take into account that the UPnP Browse action is not overhead of the virtualized approach in comparison with the original STB centered approach, as in the latter case, fetching the media descriptions would take an equal amount of time. Combining the results presented in Fig. 7 with the ones from Fig. 6, where we found that acquiring 100 items with the ALIX HGW takes 1.432 s, we deduce that the total overhead delay of virtualizing the STB is limited to 232 ms. Breaking this overhead down into individual causes of delay we come to the 100 ms thin client overhead, 61.4 ms consumed for zipping the media descriptions on the ALIX HGW (as can be found in Fig. 5), and a delay of 70.6 ms for transmission of the compressed list from the HGW to the EPG server. Note however that cyphering of the metadata is not taken into account for this setup. We expect similar processing behaviour as the zipping strategy, as cyphering also operates on the byte-level. Depending on the cyphering method used, the size of the transmitted data can increase.

IV. Local Network Resource Reservation Mechanism

To enable high quality delivery of in-home and remote content, resources in both the access network and the home network have to be managed. Access providers use their own mechanism to manage the resources on the access network, but don't have any control on the management of in-home network resources.

We present an architecture for a Service Provider to manage in-home network resources for services they provide. The local network resource reservation mechanism is responsible for in-home network resources, but can be remotely managed by the service provider through the use of a remote management protocol. Reservation of resources in the home network ensures qualitative services as perceived by the customer. By enforcing policies, the service provider can guarantee premium services like Video on Demand or streaming local content to the STB-controlled television set to be protected against services that are more tolerant with regard to network resources. Examples of these services are Internet connectivity, peer-to-peer networks, etc.

One of the major problems when managing in-home network resources, is the heterogeneity of the networking technologies used. As opposed to the access network where one single party controls the resources, the home network can be composed of multiple networking segments that each need to be managed. Every networking technology has its own admission mechanism and the network resource reservation mechanism should be able to interact with all of these. Note that not all networking technologies are able to qualitatively guarantee a steady bandwidth. The QoS management framework can be used to adapt service delivery based on the possibilities of the home networking technologies.

A. *UPnP-QoS*

UPnP-QoS v3 [17] defines a mechanism to perform policy based Quality of Service (QoS) in the home network. Prioritized QoS as well as parameterized QoS is supported. For prioritized QoS, the only policy element used is the so called TrafficImportanceNumber. This priority number categorizes all streams into eight classes. Parameterized QoS lets an application reserve network resources by specifying its requirements during QoS setup. It is possible the network is unable to fulfill the request, therefore the concept of UserImportanceNumber is introduced. This policy element indicates a relative priority of a specific stream which will be used to resolve resource contention. If streams of lower importance have reserved resources on the network, the QoS management framework can decide to release these resources in favor of the stream currently being admitted.

UPnP-QoS defines three new UPnP services to enable local network resource management. First of all, the QosManager (QM) service [18] serves as an entry point for an application. This service provides functionality to request, update and release resources for a particular stream. The QosPolicyHolder (QPH) service [19] acts as a repository of policy rules that help the QM service to take decisions based on the policies present in its policy database. These two services together form the QoS management framework, but cannot function without being able to reserve resources on a particular device. The QosDevice (QD) service [20] tackles this issue by providing an

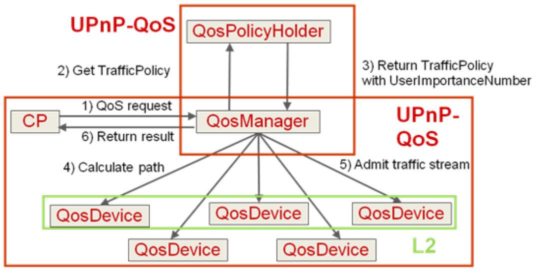


Fig. 8. Overview of UPnP-QoS communication

interface for the QM service to use. It provides an abstraction layer towards the different networking technologies, so QoS management can be performed in a uniform manner while the heterogeneity of the underlying network is handled by the QD services on the network devices in the home network.

B. Communication flow

Fig. 8 shows the communication within UPnP-QoS. Whenever an application decides it needs to qualitatively transport traffic from one destination to another, it can invoke the QM service to manage the in-home resources. No restrictions are put on the streams, source as well as destination can be in the home network or outside. Obviously, at least one of both needs to reside in the home network, as otherwise no in-home resources will be involved. The application needs to provide information that can be put in two categories: information to identify the traffic and information about the QoS requirements of the traffic.

The QM service uses the QPH service to know how the current stream should be handled in terms of admission on the network. Most of the communication however will flow between the QM service and the QD services. The path information will be queried on all QD services on the network to find out which QD services are on the path of the stream. After this, the subset of QD services that are on the path will be used to communicate the QoS requirements to. Depending on the outcome of these requests, the QM service will return the necessary information towards the requesting application.

It is important to note that so far all communication is done using standard UPnP communication. All networking technology specific communication is abstracted by a QD service specifically implemented for that technology. The UPnP QosDevice:3 Addendum [21] specifies the mapping of all UPnP-QoS related parameters to parameters related to several networking technologies. Using this abstraction layer the QD service offers, the problem of heterogeneity of the home network can be solved.

C. Implementation

We have implemented the complete UPnP-QoS v3 specifications, together with a framework that simplifies the implementation of QD services. For more information about said implementation, we kindly refer you to [22].

A demo setup using Multimedia over Coax Alliance (MoCA) hardware cards was built to show the use cases mentioned below. More information about this demonstration can be found in [23]. The setup consists of three PCs, each containing a MoCA interface card connected to a coaxial network, a shared medium. All three PCs contain a QD service implementation that communicates with the driver of the MoCA interface card to request, update and release resources on the MoCA network.

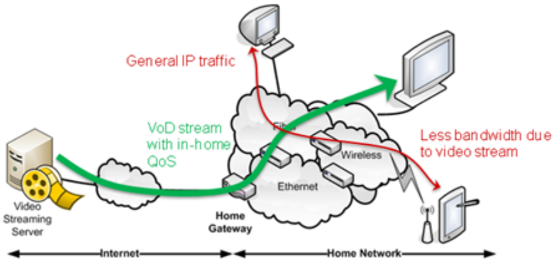


Fig. 9. Basic QoS reservation use case

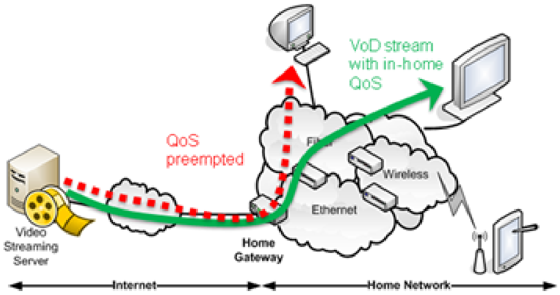


Fig. 10. Resource contention use case

We want to show the necessary steps for reliable, policy based resource reservation on a heterogeneous home network. To demonstrate this, we will consider three scenarios.

1) Basic QoS reservation: First of all, a basic reservation of network resources is considered, this scenario shows the differences in technological capabilities can be abstracted to offer a unified resource reservation mechanism across the home network (see Fig. 9). QoS is being requested for a stream originating outside the home network. Due to the fact QoS has been reserved for the VoD stream, the IP traffic within the home network will be restricted to the remaining bandwidth.

2) Resource contention: The second scenario will cover resource contention detection in that a resource reservation will be done when resources are scarce. This scenario will present the end user with a message indicating there was a problem when trying to provide QoS.

A third scenario (depicted in Fig. 10) tries to solve the problems encountered in the second scenario without user interaction. Based on predefined policies, resources are released to make room for the reservation at hand.

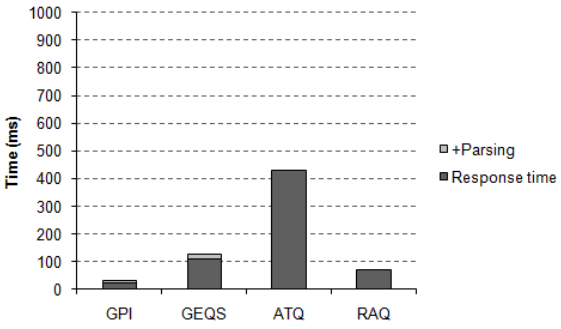


Fig. 11. Invocation times for QosDevice actions on MoCA implementation (Network Coordinator)

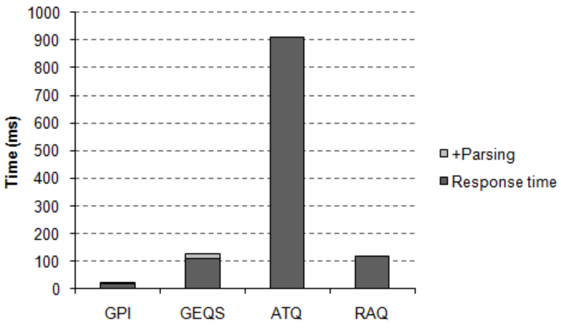


Fig. 12. Invocation times for QosDevice actions on MoCA implementation (non-Network Coordinator)

D. Performance measurements

Tests have been run on each of the setups discussed above. The test invokes the most important actions one hundred times in a row and parses its output arguments. The actions GetPathInformation (GPI) and GetExtendedQosState (GEQS) are actions that don't have to interact with lower layers while AdmitTrafficQos (ATQ) and ReleaseAdmittedQos (RAQ) change the state of the underlying networking technology. To handle this, ATQ and RAQ are always invoked in pairs where the RAQ invocation cancels the reservation done by ATQ.

Fig. 11 and Fig. 12 show the results of the test performed on the MoCA implementation. It is interesting to note that MoCA uses a Network Coordinator to do actual admission of QoS requests. Fig. 11 shows the results of the Network Coordinator node. Fig. 12 shows the result on a node that is not the Network Coordinator in the MoCA 1.1 network. As expected the results for GPI and GEQS are comparable since no interaction is needed with the MoCA network. The results for ATQ and RAQ however, reveal an interesting observation: there is a considerable penalty for having to contact the Network Coordinator. This observation is valid for invocations to ATQ which performs a QoS request as well as for invocations to RAQ which releases resources.

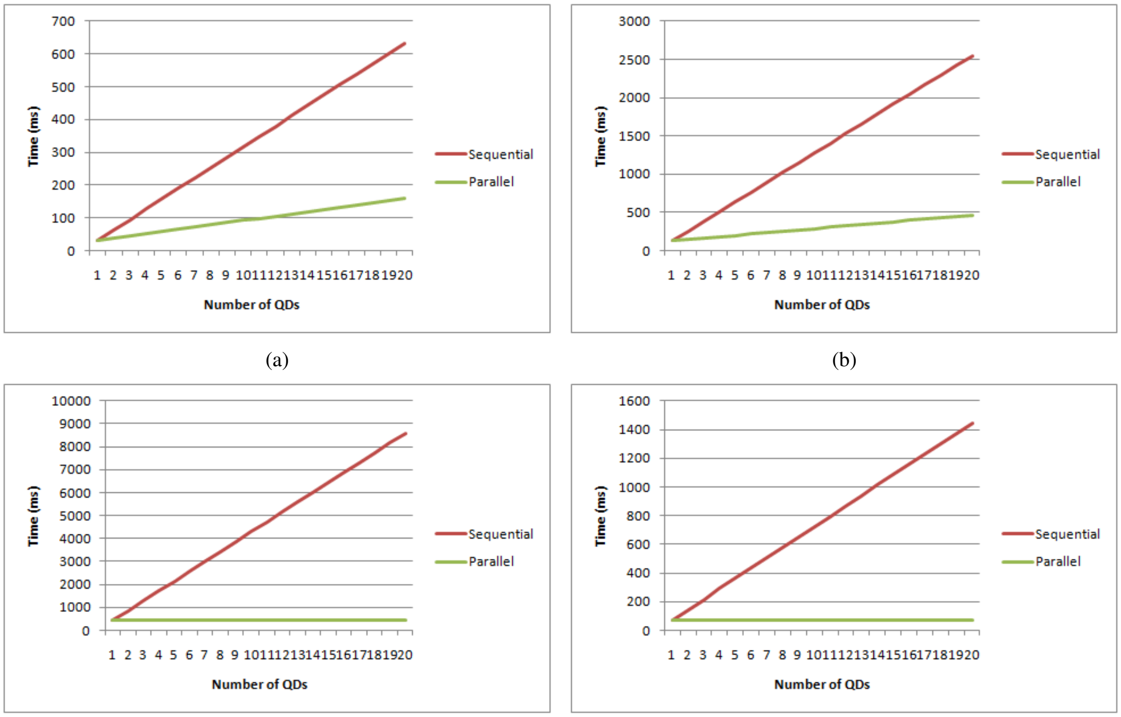
E. Optimisation

Assuming at least two QDs per layer-2 segment, our measurements (for a single QD) suggest that even for reasonably simple home networks (a couple of layer-2 segments) the whole procedure could already take a couple of seconds. To cope with user impatience, this can be limited by parallelizing each series of QD action X invocations. Note however that the processing of the output results still all needs to be done by the QM. We analyzed the performance of each series of QD action calls by measuring both response ($resp$) and processing times ($proc$) in the testbed, and comparing parallel invocation times (t_p) with the basic sequential implementation (t_s). Results are summarized in Fig. 13. For the theoretical background on this calculations, we refer you to [22].

A further optimization can be done by caching the PathInformation at the QM, since the QDs send this information upon changes (via evented UPnP state variables). Furthermore, if we assume that most QoS requests succeed, another optimization can be applied. Indeed, the invocations of *GetExtendedQosState* and *AdmitTrafficQos* are not entirely dependent on one another: the QM Entity can already admit QoS on one QD service before having information about another. However, these optimizations require more housekeeping in case of failure to admit QoS on a particular QD service. A sequential implementation would typically just release all previously admitted resources and fail without requesting the resources on the next QD service in line. When any of the parallel admissions fail the QM Entity should wait for all QD services to return and release each resource that has been admitted successfully before failing, and subsequently make extra release calls (RAQ)—which can again be issued in parallel—to each QD where ATQ requests succeeded. While sequential implementation will always perform a less than or equal number of ATQ invocations than the parallel implementation, the latter's performance in case of failing ATQs is however not compromised (since subsequent RAQ calls can be again issued in parallel).

V. Conclusion

In this paper, an architecture is proposed that offers not only flexible service and application management to a service provider, but also enables integration and interworking with the user's personal devices within the home. This architecture combines provider-side application logic with services already located in the home network to enable the service provider to deliver qualitative services. The article focuses on a service that integrates in-home multimedia content within a unified EPG. We propose to adopt a server centered approach to control in-home media content, so that the STB in the home network can be kept as lightweight as possible, i.e. only intercepting user actions and presenting content to the user. By means of experimental results, it is shown that the overhead that comes with this approach is actually inferior tot he advantages it brings. Since the service provider is rather shut out of the home network, but still wiches to guarantee qualitative services, the need for remote controllable QoS reservations is explained. The performance of this UPnP QoS architecture is analysed with experimental results, and optimizations are proposed.



(a)

(b)

(c)

(d)

Fig. 13. Parallelization of the calls to each QosDevice action keeps the response times of each series of calls below 0.5 seconds: (a) *GetPathInformation*, (b) *GetExtendedQosState*, (c) *AdmitTrafficQos*, (d) *ReleaseAdmittedQos*.

ACKNOWLEDGMENT

This work has been partly funded by the Agency for Promotion of Innovation by Science and Technology in Flanders (IWT-Vlaanderen), through the IWT Vampire project, and partly by funding of the European Community's Seventh Framework Programme (FP7) under project 212 352 ALPHA "Architectures for fLexible Photonic Home and Access networks". Bert Vankeirsbilck is funded by a Ph.D grant of the IWT-Vlaanderen. Chris Develder is supported as a postdoctoral fellow of the Research Foundation Flanders (FWO).

REFERENCES

[1] "DLNA for HD video streaming in home networking environments," Whitepaper, http://www.dlna.org/about_us/about/DLNA_Whitepaper.pdf.

[2] H. Lee, S. Moon, and J. W. Kim, "Enhanced UPnP QoS architecture for network-adaptive streaming service in home networks," *IEEE Trans. on Consumer Electronics*, vol. 53, no. 3, pp. 898–904, Aug. 2007.

[3] T. Jurglionis *et al.*, "Distributed video game streaming system for pervasive gaming," in *Streaming Day 2009*, Genova, Italy, Sept. 2009.

[4] R. Seepold, N. M. Madrid, and J. M. Fernandez, "QoS management for distributed multimedia services," in *10th IFIP/IEEE Int. Conf. on Management of Multimedia and Mobile Networks and Services (MMNS 2007)*, San Jose, CA, November 2007.

[5] Y. He *et al.*, "A reservation based backoff method for video streaming in 802.11 home networks," *IEEE Journal on Selected areas in Communications*, vol. 28, no. 3, pp. 332 – 343, 2010.

[6] J.-P. Javaudin *et al.*, "Inter-MAC concept for gigabit home networks," in *Workshop on Inter-Mac Management and Multiple interfaces for Multi-Gigabit Wireless Systems (MGWS), part of IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC 2009)*, Tokyo, Japan, Sept. 2009.

[7] C. Ge *et al.*, "The intelligent STB - implementation of next generation AOF residential gateway in digital home," in *2nd International Conference on Pervasive computing and applications, 2007, ICPCA 2007*, July 2007, pp. 256 – 261.

[8] M. Obrist, *et al.*, "Connecting TV & PC: an in-situ field evaluation of a unified electronic program guide concept," in *the 7th Conference on European Interactive Television*, Leuven, Belgium, June 2009, pp. 91–100.

[9] "UPnP AV," UPnP Forum, http://www.upnp.org/default.asp.

[10] Q. Richardson, *et al.*, "Virtual network computing," *IEEE Internet Computing, IEEE Computer Society*, vol. 02, pp. 33–38, 1998.

[11] "FFmpeg: a complete, cross-platform solution to record, convert and stream audio and video," http://ffmpeg.org/.

[12] "XML-TV file format," December 2008, http://wiki.xmltv.org/index.php/XMLTVFormat.

[13] R. Morris, *et al.*, "The click modular router," *SIGOPS Oper. Syst. Rev.*, vol. 33, pp. 217–231, 1999.

[14] "IEC/IEC CD 21000-2:2001, information technology multimedia framework part 2: Digital item declaration," July 2001.

[15] "XML schema for ContentDirectory:3 structure and metadata (DIDL-Lite)," UPnP Forum, September 2008, latest version: http://www.upnp.org/schemas/av/didl-lite-v2.xsd.

[16] A. Lai and J. Nieh., "On the performance of thin client computing," *ACM Trans. on Computer Systems*, vol. 24, no. 2, pp. 175–209, 2006.

[17] "UPnP QoS architecture v3.0," UPnP Forum, http://upnp.org/specs/qos/UPnP-qos-Architecture-v3.pdf.

[18] "UPnP QosManager:3," UPnP Forum, http://upnp.org/specs/qos/UPnP-qos-QosManager-v3-Service.pdf.

[19] "UPnP QosPolicyHolder:3," UPnP Forum, http://upnp.org/specs/qos/UPnP-qos-QosPolicyHolder-v3-Service.pdf.

[20] "UPnP QosDevice:3," UPnP Forum, http://upnp.org/specs/qos/UPnP-qos-QosDevice-v3-Service.pdf.

[21] "UPnP QosDevice:3 underlying technology interface addendum," UPnP Forum, http://upnp.org/specs/qos/UPnP-qos-QosDevice-v3-Addendum.pdf.

[22] C. Develder, *et al.*, "Delivering scalable video with QoS to the home," *Telecommun. Syst.*, 2010, in press.

[23] "http://ir.entropic.com/releasedetail.cfm?ReleaseID=402263," Online.