Journal of Mobile Multimedia, Vol. 2, No. 4 (2006) 311–326

RESOURCE ADAPTATION FOR MOBILE AV DEVICES IN THE UPNP QOS ARCITECTURE

MICHAEL DITZE
University of Paderborn
Paderborn, Germany[a]
michael.ditze@c-lab.de

Received received date: May 31, 2006
Revised revised date: Sep. 20, 2006

Distributed multimedia applications increasingly populate Audio-Visual (AV) mobile and desktop devices. Crucial to the success of these applications is the delivery of Quality of Service (QoS) which often refers to resource reservation on devices and network links. Resource reservation is usually ensured by an appropriate admission control (AC) that determines if another request for resources can be granted without interfering already accepted traffic flows and processing resources. If an AC cannot be processed or cannot grant the requested resources, proactive measures like resource adaptation may control the resource requirements of particular flows. Resource adaptation adjusts the resource demands of the application according to the available resources. This paper presents a flexible and hybrid resource adaptation framework for multimedia applications that is incorporated into the UPnP QoS architecture. Furthermore, a simple methodology for controlling resource adaptation in mobile networks and devices is presented. An implementation of resource adaptation proves that the resource utilization is optimized and the QoS can be effectively maintained even in overloaded conditions.

Keywords: End-to-end Quality of Service, Service Oriented Architectures, Universal Plug and Play, Mobile Networks, Resource Adaptation

1 Motivation

Mobile multimedia applications, e.g. Video on Demand (VoD), are increasingly used in converging networking environments. Due to their networked and time sensitive character they require Quality of Service (QoS) support. QoS is hereby defined as the joint effect of qualitative service performance which determine the degree of user satisfaction of the service [1]. The user satisfaction is reflected as a collective set of perceivable application and user preference parameters. To provide service performance, these parameters need to be transformed into quantifiable metrics for resource usage for network and device processing resources to allocate resources that may guarantee the timely delivery and processing of traffic flows. Resources are usually granted through an appropriate admission control (AC) that determines if a particular request for resources expressed through a traffic descriptor can be granted without causing interference to already accepted traffic flows. If a request is accepted, traffic handling mechanisms such as traffic policing, traffic shaping and scheduling ensure the resource allocation on the network or the processing devices.

If the AC, and hence the resource reservation request, fails the stream is definitely rejected. As the AC needs to assume worst case conditions to guarantee the resource availability, deterministic approaches for AC often prove to be unnecessarily strict in mobile multimedia systems for the following reasons:

[a]University of Paderborn/C-LAB, Fürstenallee 11, 33102 Paderborn, Germany

- Variable bit rate (vbr) encoded video may cause network traffic to vary significantly over time. In order to provide reservation guarantees the AC needs to assume resource demands according to the traffic flows peak rate which usually results in resource over-reservation and hence low utilization of resources. Furthermore, additional traffic flows may be denied as the AC needs to assume worst case conditions. These conditions occur when all flows simultaneously transmit at their peak rates which in practice will be the exception rather than the rule.

- Peak rate estimation requires a full apriori analysis of the video stream which is an unrealistic assumption for live video feeds that target low latencies. Peak rates may hence not be known to the AC beforehand. Consequently, resource demands may temporarily exceed the allocated resources, hence causing overload.

- Due to the high configuration costs in larger scaled backbone networks, service providers usually do not provide service differentiation combined with an AC, but rather rely on resource overprovisioning combined with a best effort approach [2].

In this paper we propose a hybrid solution for resource adaptation and reservation in mobile multimedia networks and audio-visual (AV) devices as a complementary and alternative approach to the traditional resource reservation schemes. The resource adaptation framework provides means for mobile devices to signal resource bottlenecks, i.e. overloads, to feeding server peers or multimedia gateways through Control Points. The Control Points may take appropriate measures to control the workload according to a simple resource adaptation methodology that proposes how to increase or decrease the resource usage by adapting the content. The methodology abstracts from the content adaptation and distinguishes between network and device resources. Whereas the limited bandwidth on wireless networks has an impact on network QoS parameters like delays, jitter, and packet loss, the lack of CPU time on resource limited mobile devices may result in late frame display. Unlike other solutions, we do not rely on a resource consuming polling method for signaling, but rather deploy an eventing scheme that reduces the network control traffic on the mobile network.

We combine the resource adaptation solution with the UPnP QoS architecture, and hence provide a powerful service-oriented architecture (SOA) to handle QoS in UPnP controlled networks, e.g. home networks. SOAs offer a promising foundation to handle QoS as they provide a common set of well defined interfaces, and thus free the application programmers from the burden of network configuration, QoS negotiation, and session establishment by abstracting from the communication protocols. An adaptation component that acts as a service is generic and reusable and may act as a central allocation facility that controls the resource usage for all applications. Furthermore, services abstract from the implementation and allow to deploy multiple content adaptation approaches like layered encoding or transcoding. Evaluation results proof that our solution optimizes bandwidth utilization without producing congestions and handles media streams effectively even in overloaded conditions.

The rest of the paper is organized as follows: Chapter II gives a short introduction to the UPnP QoS framework. Relevant related work is summarized in Chapter III. Chapter IV introduces the resource adaptation framework whereas Chapter V describes the implementation and evaluation. An outlook on future will be given in Chapter VI.

2 Introduction to the UPnP QoS Architecture

The Universal Plug and Play (UPnP) architecture is designed to connect networked devices by using existing standardized protocols such as TCP/IP, HTTP, and XML. UPnP stacks are available on most of the currently available mobile devices like PDAs and Smartphones. The basic abstractions of UPnP are *Devices*, *Services*, and *Control Points*.

UPnP devices implement a set of standardized protocols to support the discovery, control, and data transfer between connected devices. Each UPnP device exposes device capabilities and corresponding services to other devices through a well defined and commonly available service interface. Services can be invoked through UPnP control points that act as clients and own the task of device discovery, capability query, and device control [3]. The UPnP AV

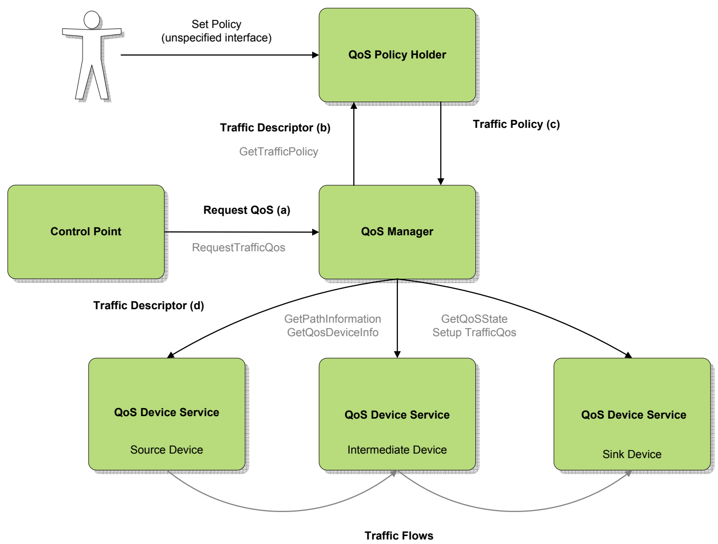

Fig. 1. UPnP QoS architecture

architecture provides means to locate media servers and media renderers, perform capability matching, and set up a streaming connection between them.

As an extension to UPnP, the UPnP QoS architecture provides services to manage QoS in local layer-2 networks by providing policy management and allocation of network resources. The UPnP QoS framework is characterized by three components [4]: *QoS Manager*, *QoS Device*, and *QoS Policy Holder*, whose relationship is illustrated in Fig. 1.

The key component of the framework is the QoS Manager [5]. As a combination of entity and UPnP service it performs the twofold task of a UPnP control point and a QoS service. As a control point, it discovers *QosDevice and* QosPolicyHolder services that run on the local subnet. As a QoS service, it will be invoked from other UPnP control points to perform QoS connection setup and admission control. UPnP allows multiple instances of a QoS Manager to coexist in a network.

The QoS Device service [6] is an UPnP service that is implemented on each QoS-capable source, sink, and intermediate network device all along the traffic path from the source to the sink. It is responsible for coordinating its network resources for traffic streams on its output links and returning its state to the QoS Manager. The QoS Manager uses the states of the individual QoS devices to process the AC.

The QoS Policy Holder service [7] offers policy management responsible for allocating network resources. It provides a centralized mechanism for enabling and disabling admission policies for the local network. Furthermore, it classifies traffic according to information as provided by the QoS Manager.

The theory of operation of the UPnP QoS architecture is depicted in Fig. 1. Greyed terms denote UPnP actions exposed by the respective devices. An UPnP AV Control Point requests the QoS Manager service in the network to setup QoS by providing a *UPnP traffic descriptor* (a). The traffic descriptor contains traffic details that include the address of the traffic source and sink and a reference to the streaming content. In particular, each traffic descriptor contains a list of ordered traffic specifications (TSPECs) that describe the QoS requirements of the flow. Once being in charge, the QoS Manager calls the QoS Policy Holder (b) to retrieve adequate policy and admission for the traffic (c). After determining the network path from the information source to the sink and querying the QoS State on each device on that path, the QoS Manager performs an admission control and decides whether to accept or reject the stream. In case a stream is accepted, the QoS Manager sets up the traffic (d) and notifies the AV control point. In case the request for resources is denied, the AC may choose an alternative TSPEC from the TSPEC list with a lower claim for resources or notify the CP about the denied request.

3 Related Work

Resource adaptation approaches can generally be classified as sender-driven, receiver-driven or transcoder-based [8]. In sender-driven approaches [9, 10] the multimedia source responds to network congestions by adjusting the transmission rate of the stream according to the remaining available bandwidth. Receiver-driven approaches [11, 12] tune the received transmission according to device capabilities, network fluctuations and user preference profiles. As such they often use layered-encoding or layered-transmission schemes in multicast networks [13], or even a combination of them. Transcoder based schemes place multimedia transcoders on gateways or proxies at appropriate network locations such as between neighbored heterogeneous networks in order to deliver different levels of service to networks with different QoS characteristics. In the following we give an overview about related work in each of the adaptation classes. The list of related work is not meant to be exhaustive and essentially gives an overview about most recent achievements.

3.1 Sender driven Resource Adaptation

Koliver et al. [14] describe a fuzzy control approach for QoS in distributed multimedia applications. It uses a quality degree function for mapping application QoS parameters into a quality degree metric that is used by a fuzzy controller for controlling, estimating, and adjusting QoS parameters depending on the resource availability. The availability of resources is determined by continuously monitoring the network traffic. An actuator assists the sender application in configuring itself by invoking services such as filtering and by changing encoder parameters. The approach proves promising for increasing the bandwidth utilization, but imposes a lot of overhead for network monitoring and resource adaptation.

In our approach we use a hybrid approach of resource reservation and resource adaptation that is supported by an admission control. The admission control anticipates the aggregated resource requirements of all traffic flows in the network and rejects additional traffic flows requests for resources if they cannot be granted. If a traffic flow exceeds the resources at disposal for the reasons given in Chapter 1, we deploy resource adaptation to adapt the content to the presently available resources. Admission control ensures that these situations occur less frequently and hence reduced the overhead for resource adaptation.

3.2 Receiver driven Resource Adaptation

In [15], Li and Nahrstedt present a framework that supports the dynamic control and reconfiguration of internal parameters and functionalities of a distributed multimedia application. The approach aims to satisfy system-wide properties like fairness among concurrent applications and application specific requirements. Their approach uses two major models: (1) The Task Control Model supports appropriate adaptation decisions to ensure system-wide properties while the (2) Fuzzy Control Model is responsible for mapping system-level adaptation decisions to application-specific reconfigurations and tuning actions. The proposed framework deploys a global coordinator that ensures that applications do not adapt in a conflicting or unfair way. In this paper we present a more flexible solution that relies on a set of local coordinators that increase the reliability. Moreover, we allow our approach to work with resource reservation schemes that allocate resources as specified in traffic descriptors. Resource adaptation is deployed if resources could not be allocated or temporarily exceed the capacities at disposal.

Biaoshun et al. [16] present a coherently designed QoS framework that integrates various run-time solutions as found in previous middleware designs with different perspectives. The framework supports resource adaptation through application configurations and performances to be tailored to different user behavior and characteristics of ubiquitous environments. Their approach focuses on run-time probing, run-time instantiation and run-time adaptation, thereby allowing multimedia applications to scale and adapt with respect to available system resources. The approach offers promising results and shows a good performance in practice. However, the middleware is tailored to the needs of specific applications and does not offer standardized interfaces like the prevailing home network architectures, e.g. UPnP. Besides, it also suffers from the polling scheme disadvantage that we replace by eventing.

Wang and Schulzrinne [17] developed an intelligent service architecture that integrates resource reservation, negotiation, pricing and adaption. Their work focuses on the assumption that content adaptation approaches lead to better resource utilization, and hence lower bandwidth costs. Therefore, they propose a dynamic, congestion-sensitive pricing algorithm for network nodes and a utility function for clients. The utility function reflects the value for a service as perceived by the user and adjusts service requests in response to price variations. Likewise, in case of congestions a larger average network price is assumed which causes network client agents to re-negotiate contracted services with network resource agents. Client agents here try to maximize the surplus between the utility function and the costs for a particular service. Similar to the approach as proposed by Wang and Schulzrinne, this thesis suggests a hybrid solution of resource reservation and content adaptation for improved resource utilization. It autonomously performs adaptation according to a dynamic and extensible methodology that is incorporated as a service in the UPnP QoS framework and does not require user interaction. The methodology considers network as well as and end-device resources in the adaptation decision.

3.3 Transcoder based Resource Adaptation

Gueknkova-Luy et al. [18] develop a generalized session control mechanism by combining the Session Description Protocol next generation (SDPng) with MPEG-21 Digital Items Adaptation (DIA) descriptions. Furthermore, they introduce an adaptation management mechanism for multimedia content distribution over heterogeneous clients. The adaption relies on media gateways or content adaptation nodes that act as network services and perform a certain set of adaptation operations by the session control mechanism. The sevices are invoked by a decision-taking engine that monitors the perceived network QoS. The proposed approach picks up a lot of interesting aspects that are relevant for the ideas presented in this work. However, Guenkova-Luy et al. mainly consider static scenarios where the adaptation methodology is defined prior to the session set up, whereas an online adaptation methodology can more accurately adjust to the available resources. Furthermore, the approach primarily takes network resources into account and neglects end-device resources required for the information processing on client- and server devices.

In their paper [19], El-Khtaib et. al introduce a framework for the transcoding of multimedia streams along with a QoS selection algorithm. The framework consists of a transcoder chain that targets to adapt the multimedia content such as it fits device capabilities while maximizing the user satisfaction at the time. The user satisfaction is computed as an aggregated average of individual QoS parameters that reflect user, content, context, device, and network profiles. The QoS selection algorithm in this approach is complex and does not qualify for dynamic on-line scenarios that need low re-configuration times. To make a dynamic content adaptation model operational, a network monitor is required that continuously tracks the network traffic in order to identify temporal resource overload conditions. We present such a Workload Service in this paper.

4 Solution

In this section we describe the elaboration of the hybrid resource adaptation and the respective extension of the UPnP QoS framework. The resource adaptation is hybrid in that it may collaborate with resource reservation schemes and in that it jointly considers network and device processing resources. The concept of hybrid resource adaptation is deployed in case the AC rejects a particular request for resources or the resource consumption temporarily exceeds the amount of resources initially reserved (e.g. in case of vbr traffic, see Chapter 1). Hybrid resource adaptation requires the network to recognize impending bottlenecks on both, intermediate network and mobile end devices, and engineer appropriate measures to control the workload with respect to the remaining resources.

We introduce three new types of service classes that we refer to as *Workload Services*, *Control Services* and *Adaptation Services*.

- Workload Services perform the task of monitoring QoS parameters on mobile and intermediate devices. They take care of network specific parameters like the used bandwidth, monitored delays and jitter per traffic stream, and end device information on CPU utilization. While it is not required by the approach, Workload Services are most ideally be implemented on every device all the way from the information source to the sink. When a Workload Service anticipates an approaching resource overload, the QoS parameters will be signaled to the Control Service in charge of the traffic stream that caused the overload.

- Control Services are located in the QoS Manager and have knowledge about the network topology and admitted traffic flows together with their requested QoS parameters. The job of the Control Service is to catch resource bottleneck messages and engineer appropriate measures by instructing Adaptation Services. Control Services make decisions on how to adapt resources according to an adaptation methodology. Decisions may range from simple actions like immediate switch-overs to lower bit rate content by removing enhancement layers in layered encoding schemes [20] to computationally intensive actions like transcoding to fit a certain bandwidth.

- Adaptation Services perform the actual content adaptation. While they may be implemented on any device, they heavily depend on the processing power of the underlying hardware. Hence, resource constrained mobile or intermediate devices may be capable of switching to content with additional quality enhancement layers and more powerful devices may transcode multimedia content in real-time.

4.1 Extensions of the UPnP QoS Architecture

We use UPnP to implement these services as UPnP is one of very few middleware approaches that considers QoS. Furthermore, it provides a set of standardized interfaces that allow our approach to work with presently available mobile UPnP devices. One of the most important design goals of the new approach is compatibility. Consequently, all extended services and implementations of the Workload, Control and Adaptation Services are fully compliant with the existing UPnP QoS architecture. As UPnP provides an open framework, we propose the following extensions in order to realize the service interaction required to perform the adaptation [b].

- The UPnP QoS traffic descriptor is extended by an *OptionalResourceAdaptationParams* element that contains four sub-elements that serve as input values for the Adaptation Service to perform transcoding with appropriate data rates. The *ResourceAdaptation* element is represented as a flag that indicates whether a particular traffic stream shall be adapted when a resource bottleneck occurs. The flag is either enabled or disabled. *MeanDataRate* and *MinimumDataRate* are elements that need to be determined and encoded in the traffic descriptor. Whereas the first element describes the expected available bandwidth as determined by the AC, the latter denotes the expected available bandwidth in the worst case, i.e. when all streams claim the maximum peak rate. Last, *QoSManagerId* identifies the QoS Manager in charge of a particular stream.

- At least one Workload Service needs to be implemented on any device from the traffic source to the sink. Additional Workload Services may improve the adaptation. They trigger the UPnP QoS Manager in charge of a particular stream about impending or present resource bottlenecks by throwing a *WorkloadInformation* event. At the same time, they provide a *GetWorkloadInformation* action that allows the QoS Manager to query the devices workload. On request, the Workload Service then passes a state variable to the manager that includes the current workload, the average workload observed over a certain period and the traffic descriptor of the stream to be adapted. The Control Point of the QoS Manager needs to be extended in order to handle these thrown events and variables.

[b] Cursive printed terms denote actions, events or state variables in UPnP terminology

- As the UPnP QoS Manager is a stateless device, it has no knowledge about the available resources, and hence cannot notify the Adaptation Service with the QoS information required for determining the data rates for adaptation. In order to receive QoS information we allow to throw an *AdjustmentRequest* event that each QoS Manager needs to subscribe to in order to query devices about their QoS states.

4.2 Theory of Operation in the UPnP QoS Architecture

For the sake of simplicity we illustrate the theory of operation only for UPnP QoS and resource adaptation services in a small environment as depicted in Fig.2 that represents a home network with mobile devices. Once video source and sink have been identified, the CP is in charge of asynchronously setting up the QoS session by providing an initial traffic descriptor to the QoS Manager through the *RequestTrafficQos* call (1). This traffic descriptor does not yet contain the *OptionalResourceAdaptationParams* part at this time, but this element is added to the traffic descriptor later in the setup process by the QoS Manager. The QoS Manager may optionally invoke the *GetPathInformation* service exposed by the sink and source devices to determine intermediate nodes on the data path (2). As a further option, the manager may invoke the *GetQosDeviceInfo* on every device to receive information about the port number and protocol information associated with the traffic descriptor (3). The information is stored in the updated traffic descriptor that is returned to the Control Point in response to the *RequestTrafficQos* call.

As a mandatory step (4), the QoS Manager determines the QoS state on all devices along the data path by invoking the devices *GetQoSState* action, thus collecting necessary information to perform the AC for the current stream. The QoS states of the devices allow the QoS Manager to perform an admission control for the current stream. If the QoS Manager concludes that there are insufficient resources, it may reprocess the admission control with an alternative TSPEC listed in the UPnP traffic descriptor. If the AC fails and resources cannot be guaranteed, the QoS Manager deploys resource adaptation by adding the following elements to the traffic descriptor and setting up the session with a *RequestTrafficQoS* call on all devices.

- The *ResourceAdaptation* field must be set to *enabled* for all devices that may experience insufficient resources.
- Mean data rates and minimum data rates need to be calculated and encoded in the traffic descriptor.

Upon session set up, the QoS Manager subscribes to the *WorkloadInformation* event to catch signaled resource overloads and to the *AdjustmentRequest* event of the Adaptation Service. The AV control point may now set up the connection.

The next steps are repeated every time a resource bottleneck is anticipated and hence are marked as step *x* and *x+1*. If any device on the data path discovers resource bottlenecks, it signals a *WorkloadInformation* event to the network (x). The event is caught by the QoS Manager which determines the additional parameters for the extended traffic descriptor and invokes the Adaptation Service (x+1) to perform adaptation with resources that stay within ranges as given in the traffic descriptor.

In the following example we assume that the sink device does not have sufficient resources to transmit a traffic flow. Hence, the QoS Manager provides the sink with a traffic descriptor with the *ResourceAdaptation* element set to *enabled* and the rest of the *OptionalResourceAdaptationParams* elements set to proper values.

In the last step, a call to *SetupTrafficQos* is made on all devices that are involved in the streaming process. Note that in the ordinary UPnP QoS Architecture the same traffic descriptor is used in every call of *SetupTrafficQos*. In our approach we allow the QoS Manager to adapt the traffic descriptor/s *OptionalResourceAdaptationParams* element (depending on the device/'s current workload).

Once the initial traffic descriptor has been updated and the *SetupTrafficQos* action has been successfully processed on all QoS devices involved in the setup process, the QoS Manager returns the complete traffic descriptor as an output argument to the *RequestTrafficQos* call

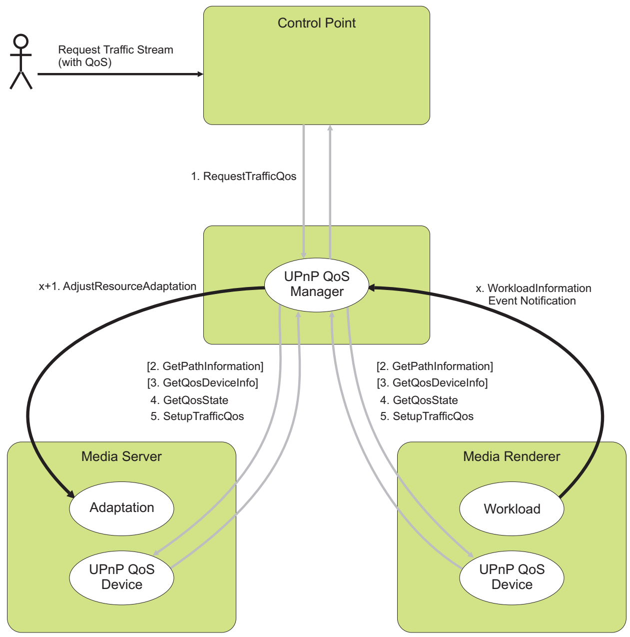

Fig. 2. WorkloadInformation Event Notification

to the control point. The latter may now setup the streaming on either the media server or media renderer device

The resource adaptation framework relies on continuous monitoring in order to properly identify impending and present resource congestions. The monitoring can be either actively performed by any device on the data path from the source to the sink, or it can be triggered by the QoS Manager to update the QoS states. Active and triggered monitoring are controlled through the *WorkloadInformation* and *AdjustmentRequest* events, respectively.

4.2.1 Active Monitoring through the WorkloadInformation Event

In case of active monitoring the Workload Service of the media renderer recognizes the overload through one of the monitored workload parameters and sends a *WorkloadInformation* notification to the network (Fig.2). In this example only the QoS Manager in charge of the resource adaptation on that particular device receives the *WorkloadInformation* event notification. However, it checks whether the *QosManagerId* in the traffic descriptor matches its own identification. If so, the QoS Manager processes the event notification according to the adaptation methodology. It calculates the adapted parameters for the *AdjustResourceAdaptation* action and, if necessary, invokes the Adaptation Service on the media server device. The adaptation ensures that the stream corresponds to the QoS parameters as obtained in the traffic descriptor. It is up to the device manufacturer to implement appropriate content adaptation schemes.

4.2.2 Triggered Monitoring through the AdjustmentRequest Event

For triggered monitoring, the Adaptation Service of the media server throws an *AdjustmentRequest* (y) to periodically update the QoS states of certain traffic flows (Fig.3). Once again,

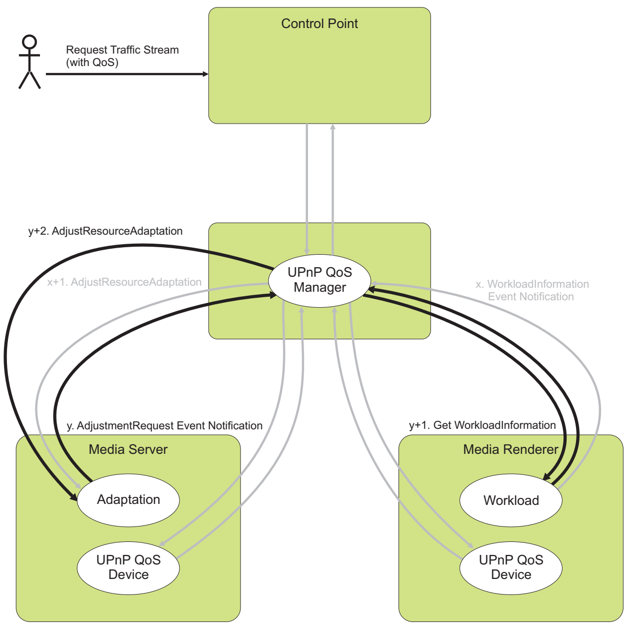

Fig. 3. AdjustmentRequest Event Notification

the QoS Manager responsible for that flow is identified through the *QosManagerId* in the traffic descriptor. As a matter of fact, every QoS Manager that recognizes the *AdjustmentRequest* event checks if the *QosManagerId* corresponds to its own. If so, the QoS Manager responds to the event similar to the active monitoring case as explained above. It analyzes the resource situation by invoking the *GetWorkloadInformation* (y+1) action on all devices along the traffic path. In order to allow a quick identification of these devices, we use the extended traffic descriptor. On response, the QoS Manager determines the need for resource adaptation according to a the adaptation methodology. If required, the QoS Manager invokes the Adaptation Service (y+2).

4.3 Methodology for Resource Adaptation

As indicated, the QoS Manager evaluates the QoS parameters from the Workload Service and determines measures to control the workload. Due to frequent and unpredictable workload variations on both, end devices and network devices, video streaming with vbr traffic is one of the most challenging applications for resource adaptation. We developed a simple methodology for the Control Service in order to make resource adaptation decisions as a tradeoff between available CPU and network resources for video decoding/encoding and transmission along the traffic path, respectively. The methodology abstracts from content adaptation approaches like rate adaptation, layered encoding/decoding as well as transcoding and that are expected to be implemented by device manufacturers or service provider.

- low bandwidth/low processing power: The video is transmitted with a reduced fps rate and/or reduced resolution and increased compression ratio in order to reduce band-

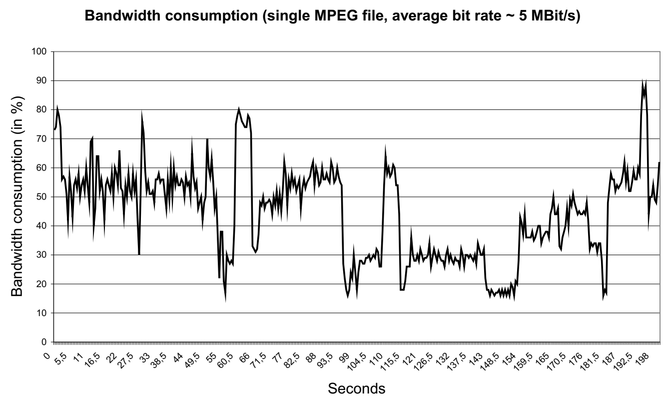

Fig. 4. Bandwidth consumption (single MPEG file, average bit rate 5 MBit/s)

width, thereby decreasing the video quality. The compression algorithm is adapted such as it fits the maximum available bandwidth in order to reduce processing power for decompression.

- high bandwidth/low processing power: The compression algorithm is adjusted such as it fits the maximum processing resources at disposal at the mobile client to reduce resources required for decompression.
- low bandwidth/high processing power: The video is transmitted with low fps rates and the compression ratio is increased to reduce bandwidth requirements.
- high bandwidth/high processing power: Transmit video with optimal quality.

5 Implementation and Evaluation

We implemented the UPnP QoS architecture on Linux with the additional services and enhancements as introduced. Hence, the Control Point establishes sessions through the QoS Manager and the corresponding QoS Policy Holder. As Media Server and Renderer we used Intels Mirco AV sample implementation for Linux that we extended with regard to QoS and resource adaptation support.

Without loss of generality, we set up a testbed that consists of 4 PCs interconnected through a 54 Mbit/sec 802.11g wireless network which is slowed down to 10 MBit/sec in order to easier produce overloaded conditions. We chose MPEG-2 vbr encoded video files at a frame per second (fps) rate of 25 fps and a resolution of 720x576 pixels which conforms to the PAL standard. The mean data rate per stream amounts to 5 Mbit/sec. Due to the nature of vbr we expect 2 streams to temporarily overload the network, thus producing resource bottlenecks. As the Intel Media Server exhibits some protocol restrictions, we used HTTP as the transport protocol. HTTP does not allow packet loss, and hence we chose the bandwidth consumption and displayed frame rate as measures to evaluate the effectiveness of resource adaptation. As in the experiments we chose video transcoding as the content adaptation approach of choice, we also deployed the open source FFmpeg codecs as video encoders.

We defined three scenarios to evaluate the superiority of resource adaptation.

- Scenario 1 transmits two video streams from the media server to the media renderer. In this scenario we do not support resource adaptation and hence expect the frame rate to drop significantly in overloaded conditions.

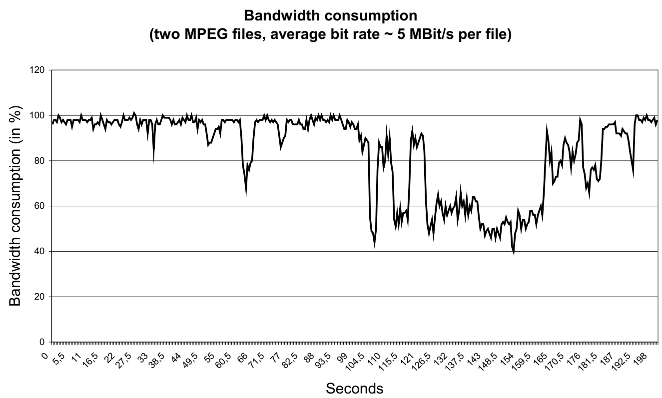

Fig. 5. Bandwidth consumption (Scenario 1: Two MPEG files, average bit rate 5 MBit/s per file)

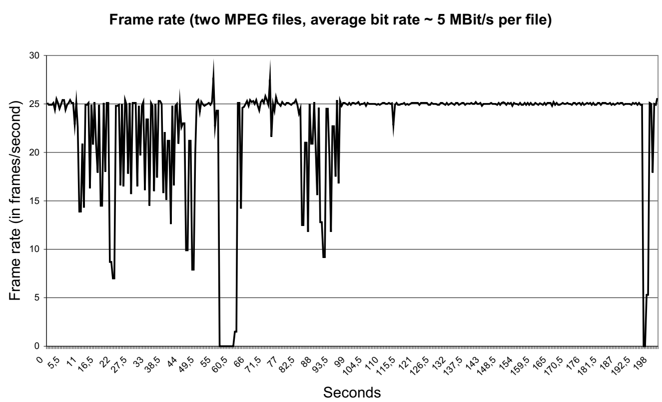

Fig. 6. Frame rates (Scenario 1: Two MPEG files; 3 MBit/s stream)

- Scenario 2 deploys a static resource adaptation for the same set of video streams. Static in this context means that the decision about resource adaptation is made by the AC prior to the session setup. We expect to optimize bandwidth utilization in this scenario while maintaining a high fps rate.

- Scenario 3 aims to demonstrate the full effectiveness of resource adaptation by allowing dynamic resource adaptation at run-time. Similar to Scenario 2, bandwidth utilization should be improved and the frame rate is expected to adjust gracefully.

5.1 Quantitative QoS Evaluation

In the first set of experiments we evaluate the quantitative QoS requirements in terms of bandwidth utilization. Fig. 4 illustrates the network load for one sample MPEG stream that consumes up to 90% of the available bandwidth. The first half of the video requires about 60% of the available resources, so we expect to overload the network as soon as we start a second instance of the stream 5 seconds after the first.

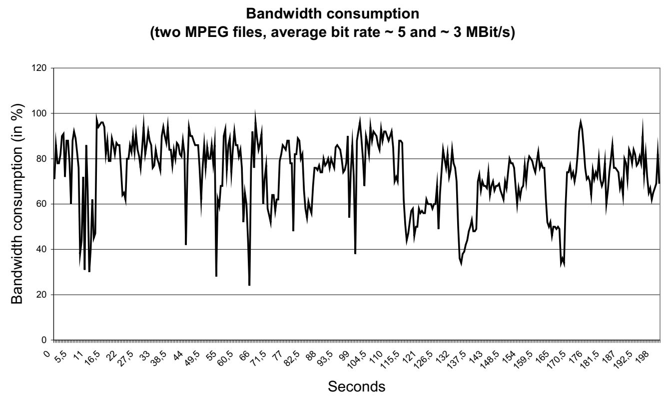

Fig. 7. Bandwidth consumption (Scenario 2: Two MPEG files, average bit rate 5 and 3 MBit/s)

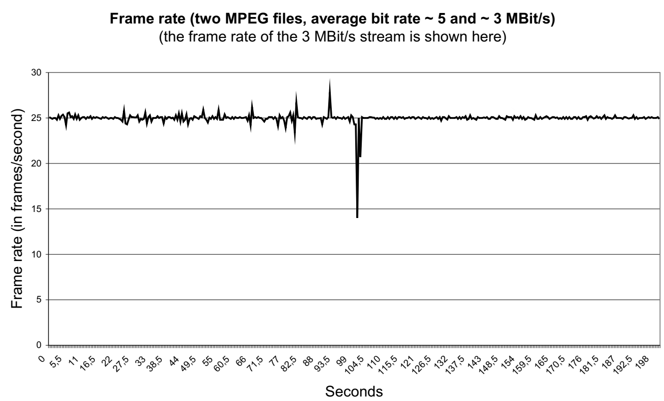

Fig. 8. Frame rates (Scenario 2: Two MPEG files; 3 MBit/s stream)

Scenario 1 As expected, the bandwidth consumption easily amounts to 100% (see. Fig.5) and consequently frame packets will be delayed causing strong jittered display in the first half of the video. At certain times the decoder gets completely out of step, stopping the video playback for six seconds (Fig.6). Due to the lower resource consumption of the video in the second half, the full fps rates of both videos can be displayed. However, once the workload increases towards the end of the video, the frame rate drops dramatically.

Scenario 2 In the second scenario the Control Service reduces the target bit rate of the second video to 1 Mbit/s following the resource adaptation approach. The admission control ensures in that case that the consumed bandwidth does not exceed the available resources. Consequently, the results of this experiment illustrate (Fig.7) that bandwidth utilization has been improved significantly, averagely consuming 70% of the resources during the first half of the video with some workload peaks at 90%. The results are also reflected in Fig.8 that demonstrates that using resource adaptation all frames can be timely decoded compared to Scenario 1 where frames are delayed, and hence dropped by the decoder.

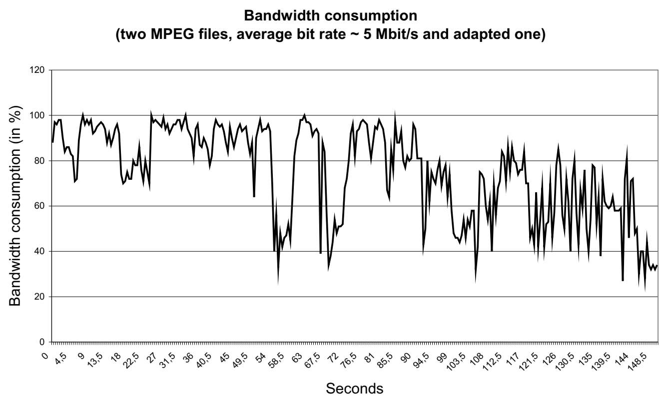

Fig. 9. Bandwidth consumption (Scenario 3: Two MPEG files, average bit rate 5 MBit/s + adapted stream)

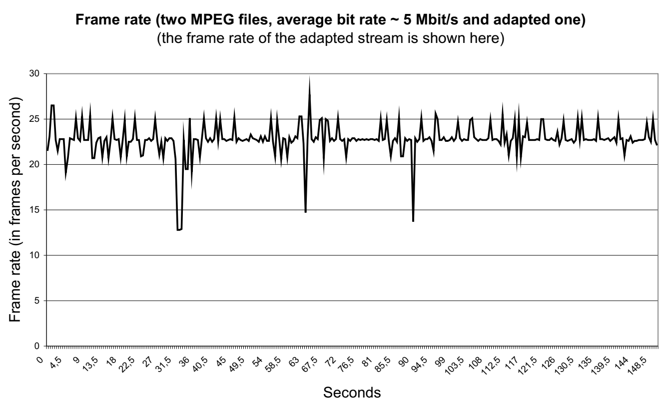

Fig. 10. Frame rate (Scenario 3: Two MPEG files, adapted stream)

Scenario 3 In the last scenario we deploy dynamic resource adaptation at run-time, allowing the server peer to adjust the target rate according to the analyzed workload. As expected, the bandwidth utilization improves since the encoder buys available resources with an increased video quality, resulting in higher workloads (Fig.9). The fps rate ranges between an acceptable 20 and 25 frames. Sometimes, however, the frame rate drops further below which occurs if impending resource bottlenecks have not been timely recognized by the Workload Service. Nonetheless, resource adaptation demonstrates a good performance in this highly challenging scenario. A subjective image comparison also reflects these results.

5.2 Qualitative QoS Evaluation

As QoS denotes the level of service as delivered to the user, in the second set of experiments we evaluate the image quality of three picture samples taken from three different video files used in the scenarios above. The samples are extracted from the first half of the video stream that exhibits a fairly high bitrate. In order to show the differences among the three samples, part of the images is shown at a larger scale [c] in the right bottom of each sample. Furthermore,

[c] The images are enlarged by a zoom factor of 3 in each direction.

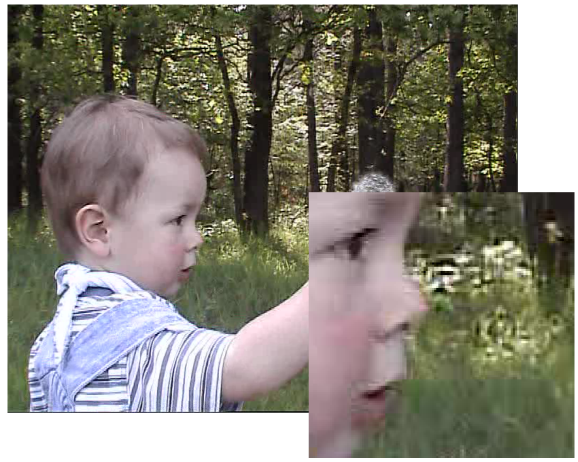

Fig. 11. Single frame of the 5 MBit/s video

we evaluated the perceived visual quality of the video during playback with a set of test users in our usability lab.

Figure 11 displays the first image sample that was extracted from the 5 MBit/s video. The video bitrate corresponds to the average bitrate of Digital Video Broadcasting(DVB). To our surprise, even in this fairly high quality, the image already exhibits compression artifacts. Our investigations identified the open source encoder FFmpeg as the cause of the lower quality in comparison to some professional studio equipment used in public broadcasting services, and the additional hurdle that the multimedia file was converted to 5 MBit/s with a single pass encoding scheme [d]. However, test users evaluated the visual quality during video playback as better than acceptable.

Figure 12 shows the same extract taken from the 3 MBit/s video file. The quality difference between the 5 MBit/s and the 3 MBit/s screenshot is surprisingly small. Only slightly more compression artifacts emerge. Similarly, test users rated the visual quality as acceptable.

The last image sample was extracted from the 1 MBit/s video stream. The visual quality of this sample does not satisfy the qualitative QoS requirements. The checked pattern of two high compression rates can be easily recognized. Image details that are still visible in the 3 MBit/s version get blurred and worsen the perceived image quality significantly. Square edges appear in areas of high contrast. As a matter of fact, the visual quality still decreases during playback when artifacts move across the screen. Consequently, the sample has been rated as unsatisfactory by most of the test users.

As a conclusion, the adapted 5 MBit/s and 3 MBit/s samples will match most of the qualitative QoS requirements that users will ask for. In contrast, 1 MBit/s adapted video streams do not prove to be satisfactory. The result suggests a minimum threshold for resource adaptation. At the same time, however, the qualitative comparison of the samples leads to the assumption that resource adaption can significantly lower the resource consumption while maintaining an acceptable perceived visual quality.

6 Summary and Future Work

This paper presented a new hybrid solution for resource adaptation and reservation accommodated into the UPnP QoS architecture. It provides means to dynamically adapt video content to the available bandwidth by exploiting an event supported signaling approach that notifies feeding peers about impending resource bottlenecks. Evaluation results illustrate that

[d] A better quality can be achieved if two or more passes are used for encoding. Unfortunately, at the time of writing this paper, 2-pass video encoding did not match the real-time requirements of video adaptation.

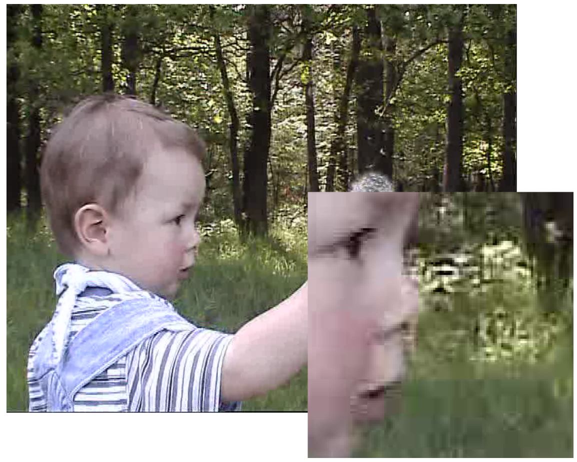

Fig. 12. Single frame of the 3 MBit/s video

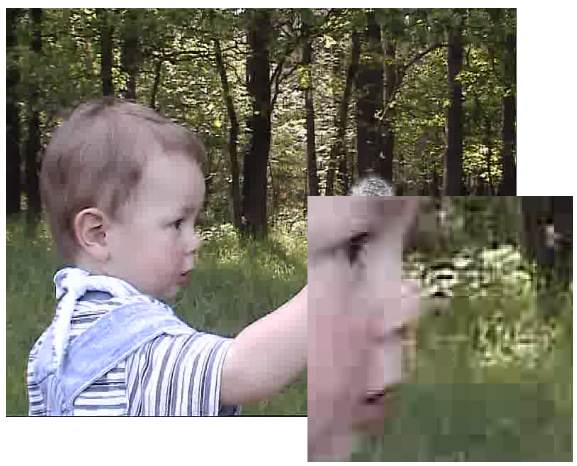

Fig. 13. Single frame of the 1 MBit/s video

using resource adaptation in mobile networks may significantly reduce bandwidth utilization and the smoothed fps rate leads to an enhanced video playback in overloaded conditions.

We already combined EuQoS and UPnP QoS [21] in order to perform layer-3 resource reservation across heterogeneous access networks, and in future we will integrate resource adaptation into this combined architecture. EuQoS targets to design and develop a flexible end-to-end QoS assurance system across heterogeneous access networks. It uses a combination of horizontal and cross layer signaling protocols in order to negotiate for resources and configure the communication system accordingly. If resources cannot be granted by an Admission Control, middleware may consider to take appropriate measures like the ones described here to reduce the demanded bandwidth and restart negotiation. Furthermore, we will improve the Workload Service through traffic analysis and prediction to earlier catch impending workload peaks.

7 Acknowledgements

This work has been achieved in collaboration with the European funded Integrated IST Project EuQoS (www.euqos.org) that researches, integrates, tests, validates, and demonstrates end-to-end QoS technologies to support the infrastructure upgrade for advanced QoS aware applications over multiple, heterogeneous network domains.

References

1. ITU-T Recommendation E.800 (1994), *Quality of service and dependability vocabulary*, Geveva, Switzerland.
2. Cisco Systems White Paper (2003), *Service Provider Quality-of-Service Overview*.
3. UPnP^{TM} Forum (2003), *"UPnP^{TM} Device Architecture 1.0"*, Version 1.0.1, www.upnp.org.
4. UPnP^{TM} Forum (2005), *"UPnP^{TM} QoS Architecture 1.0"*, Version 1.0, www.upnp.org.
5. UPnP^{TM} Forum (2005), *"UPnP^{TM} QoS Manager Architecture 1.0"*, Version 1.0, www.upnp.org.
6. UPnP^{TM} Forum (2005), *"UPnP^{TM} QoS Device Architecture 1.0"*, Version 1.0, www.upnp.org.
7. UPnP^{TM} Forum (2005), *"UPnP^{TM} QoS Policy Holder Architecture 1.0"*, Version 1.0, www.upnp.org.
8. X. Wang and H. Schulzrinne (1999), *Comparison of Adaptive Internet Multimedia Applications*, In IEICE Trans. Commun., vol.82, pp.806-818.
9. S. Jacobs and A.Eleftheriadis (1997), *Real-time dynamic shaping and control for Internet video applications*, Workshop on Multimedia Signal Processing, Princeton, USA.
10. D. Sisalem and H. Schulzrinne (1998), *The loss-delay adjustment algorithm: a TCP-friendly adaptation scheme*, in Proc. of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), Camebridge, England.
11. B.J. Vickers, C. Albuquerque and T. Suda (1998), *Adaptive multicast of multi-layered video: rate based and credit-based approaches*, in Proc. of IEEE Infocom 98, San Fransisco, USA.
12. D. Sisalem and F.Emanuel (1997), *QoS control using adaptive layered data transmission*, in Proc. of IEEE Multimedia Systems, Austin, USA.
13. S. McCanne, V. Jacobson and M. Vetterli, (1996), *Receiver-driven Layered Multicast*, In Proc. of SIGCOMM96, CA, USA.
14. C. Koliver, K. Nahrstedt, J.M. Farines, J.D.S Fraga and S.A. Sandri (2002), *Specification, Mapping and Control of QoS Adaptation*, Real Time Systems Journal, Kluwer Academic Publishers, Special Issue on Control-Theoretical Approaches to Real-Time Computing, Vol. 23.
15. B. Li and K.Nahrstedt (1999), *A Control-Based Middleware Framework for Quality-of-Service Adaptations*, IEEE Journal on Selected Areas in Communications, Vol.17, No.9, pp.1632-1650.
16. L. Baochun, X. Dongyan, and K. Narhstedt (2002), *An integrated runtime QoS-aware middleware framework for distributed multimedia applications*, In ACM Multimedia Systems Journal, ACM Springer-Verlag, Vol. 8, Issue 5, pp. 420-430.
17. X. Wang and H. Schulzrinne (2005), *Incentive-Compatible Adaptation of Internet Real-Time Multimedia* IEEE Journal on Selected Areas in Communication, Vol.23:417-436.
18. T. Guenkova-Luy, M. Gomez, C. Timmerer, I.Wolf and A. Kassler (2006), *Advanced Multimedia Management - Control Model And Content Adaptation*, In Proc. of Internet and Multimedia Systems and Applications (EuroIMSA), Innsbruck, Austria.
19. K. El-Khatib, G. Bochmann and A. El Saddik (2004), *A QoS-based Framework for Distributed Content Adaptation*, In Proc. of the IEEE First International Conference on heterogeneous wired/wireless Networks (QSHINE 2004), Dallas, USA.
20. S. Devillers, C. Timmerer, J. Heuer, and H. Hellwagner (2005), *Bitstream Syntax Description-Based Adaptation in Streaming and Constrained Environments*, IEEE Transactions on Multimedia, vol. 7, no. 3, pp. 463-470.
21. M. Ditze and I. Jahnich (2005), *Towards End-to-End QoS in Service Oriented Architectures*, In Proc. of the IEEE International Conference on Industrial Informatics, Perth, Australia.