A Personalized HTTP Adaptive Streaming WebTV

Rui Santos Cruz[1], Mário Serafim Nunes[1], and João Espadanal Gonçalves[2]

[1] IST/INESC-ID/INOV, Lisboa, Portugal
mario.nunes@ieee.org, rui.cruz@ieee.org
[2] Instituto Superior Técnico, Lisboa, Portugal
espadanal@gmail.com

Abstract. This paper presents a HTTP based multimedia content streaming architecture able to provide large scale WebTV services to end users connected over several access network technologies. The architecture features smooth Rate Adaptation, context Personalization information (Channel Suggestions), fast program channel switching, Network Portability (Vertical Handover) and Terminal Portability capabilities. The prototypical solution is being developed under the scope of the *My-eDirector 2012* european project.

Keywords: WebTV, Smooth Streaming, Content Adaptation, Rate Adaptation, HTTP Progressive Download.

1 Introduction

An increasing large number of people is already having access to all kinds of multimedia content (video, television, audio, graphics or simply data) from all kinds of networks using a large variety of terminal devices, such as computers, Set-Top-Boxes and televisions, networked portable media players, Internet-enabled smartphones or just mobile phones. The number of multimedia contents accessed by users over IP networks is also reaching very high levels, mostly due to video storage availability and live streaming of videos or television programs.

On the content side the worldwide trend is clearly towards user-generated or personalized contents. This trend, associated with the maturity level of perceptual technologies for context-awareness and the need and demand from operators, service providers and broadcasters for novel revenue generating networked services is the main driver for the development of innovative personalized media streaming services that enable new forms of rich user interaction with live or archived contents.

Streaming producers, most of them being media and entertainment companies, have a particular interest in personalization capabilities as they can affect the audience of a particular streaming service and also because their business models are mainly based on advertisement, depending ultimately on the user's viewing satisfaction.

P. Daras and O. Mayora (Eds.): UCMedia 2009, LNICST 40, pp. 227–233, 2010.

These personalized media streaming services aim to provide a novel viewing experience on the end user's device by means of recommendation services that are used for the selection and viewing of particular streams matched to the user preferences and profile.

But these personalization capabilities require adaptation of contents at different levels (media production, distribution and viewing) and adaptation of the viewing experience to the status of the environment (network capabilities, device type capabilities, physical environment).

This paper shares the experience on the development of a personalized HTTP Adaptive Streaming architecture for WebTV services, realized within the scope of the european project *My eDirector 2012* [1].

This WebTV oriented solution uses standard HTTP transactions for multimedia content streaming that are capable of traversing any firewall or proxy server that lets through standard HTTP traffic.

The HTTP Adaptive Streaming architecture is capable of providing video streaming to end users, connected either directly or through Content Delivery Networks (CDN) over several access network technologies, enabling an uninterrupted streaming experience, guaranteeing, for each user, the best possible video quality reception. With this architecture,, multimedia sources (video and audio) can be encoded in various bitrates, and a HTTP progressive download method provides the capability to react to both bandwidth and local terminal conditions in order to seamlessly switch the video quality that the user is receiving, maximizing the Quality of Experience (QoE).

One of the most important functionalities of the architecture is the seamless bitrate switching. The switching heuristics are implemented on the Client side, not involving any process at the Streaming Server. It is the responsibility of the Client to monitor network conditions, chunk download times, buffer fullness, rendered frame rates, and other factors, such as screen resolution, playing/paused status of the content, or the CPU load, in order to decide when to request higher or lower bitrates from the Streaming Server.

Other important features implemented in the architecture are the Network Portability (Vertical Handover) and terminal Portability capabilities, allowing a flexible user experience in terms of either a continuous uninterrupted playback during network handover with the same terminal or a suspend/resume session playback, either with the same terminal or with a different terminal (with same or with different capabilities like screen resolution or access network attachments).

In this paper, Section 2 describes the overall architecture of the HTTP Adaptive Streaming WebTV solution, Section 3 presents the Web Media Player Client architecture, Sections 4 and 5 the Portability functionalities of the solution prototype and Section 6 concludes the paper.

2 The HTTP Adaptive Streaming

The HTTP Adaptive Streaming architecture comprehends a Server component, a Distribution component and a Client component.

The Server component (Service Provider Encoder) is responsible for taking input streams of media from Content Provider Video Streamers and encode them digitally, to a format suitable for delivery, and preparing the media for distribution.

The Distribution component (Smooth Streaming Server) consists of Web Servers responsible for accepting Client requests and delivering prepared media and associated resources to the Client.

The Client component consists of a Web Media Player responsible for determining the appropriate media to request, download those resources, and then reassemble them to be presented as a continuous stream.

The architecture also uses HTTP for personalization and program channel information (Channel Suggestions/Recommendations) and associated Metadata, with exchange format based on eXtensible Application Markup Language (XAML), offering Camera Selection suggestions, fine control over presentation of closed captions and other timed text content.

The HTTP Adaptive Streaming architecture uses a streaming technology based on a HTTP progressive download method.

Both Microsoft, with "Smooth Streaming" [6], and Apple Inc., with "HTTP Live Streaming" [5], propose similar technologies based on HTTP progressive download.

For the initial implementation of the WebTV architecture the choice fall on Microsoft's Smooth Streaming.

Smooth Streaming: The Smooth Streaming technology from Microsoft uses MP4 Media File Format specification, MPEG-4 Part 14 (ISO/IEC 14496-12) [2], for both the storage (*disk*) and transport (*wire*) formats, architected with H.264 [3] video codec support.

The *wire* format defines the structure of the chunks that are sent to the Client, for both stored or live contents, whereas the *disk* format defines the structure of the contiguous files on disk for stored content.

For the Smooth Streaming method, the multimedia source at the Server is segmented into "chunks" from 2 to 4 seconds long, cut along a video GOP (Group of Pictures) boundary (starting with a key-frame) and have no dependencies on past or future chunks/GOPs. The chunks are encoded to the desired delivery format (H.264) and made available at the Distribution server to be delivered upon Client requests. The Client uses the Smooth Streaming method to request the chunks. Each chunk can be decoded independently of other chunks.

3 The Web Media Player Client

The Web Media Player Client architecture, a cross platform, cross browser and cross device Rich Interactive Application (RIA) based on Microsoft *Silverlight* technology [4], takes the form of a "plug-in" with a modular design, as illustrated in Figure 1, facilitating the implementation of enhanced/new functionalities.

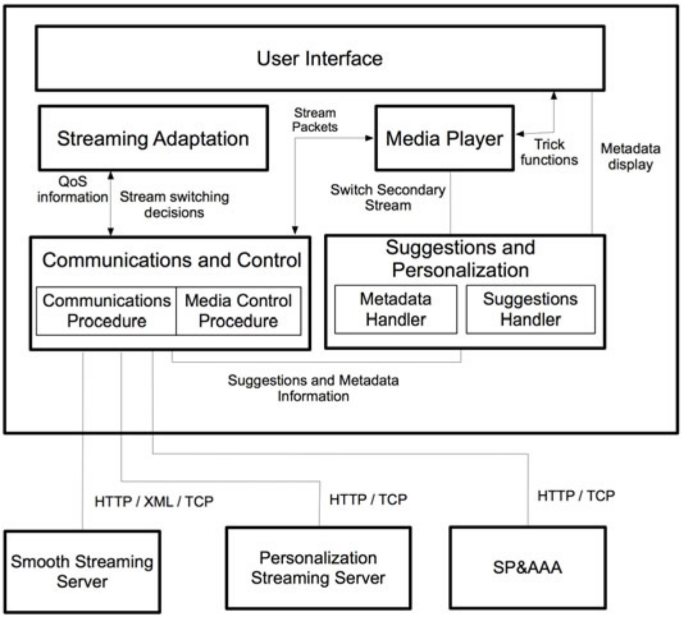


Fig. 1. Web Media Player Client Application Modules

The Communications and Control Module: The Communications and Control Module implements the Smooth Streaming method and is responsible for Session Setup, Session Termination, content negotiations with the Smooth Streaming Server, polling the Personalization Streaming Server (PSS) for Channel Suggestions and stream Metadata and for monitoring network conditions.

A typical content negotiation message sequence is illustrated in Figure 2. In this module a Communications Procedure handles all session control while a Media Control Procedure handles network Trick Functions, in order to control the media stream status (PLAY, PAUSE) as well as the session SUSPEND/RESUME feature for terminal Portability.

The Suggestions and Personalization Module: The Suggestions and Personalization Module is used for the treatment of textual information on Personalization and content Metadata, including Camera Selection, Event comments, subpicture information or news about some other events.

A polling method with a resolution of 1 second has been chosen for all requests to the Personalization Streaming Server (PSS), through port 80, preventing blocking/filtering conditions on traversing network boundaries.

The Streaming Adaptation Module: The Streaming Adaptation Module implements the Network Quality and Host Capabilities adaptation strategies for the selection of the most appropriate bitrate at any moment in time.

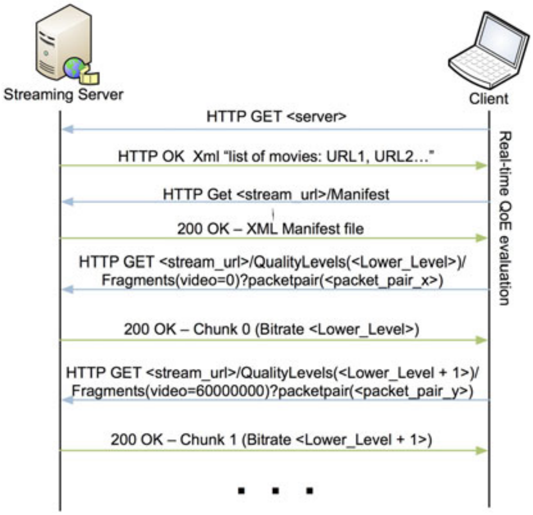

Streaming Server
Client
HTTP GET <server>
HTTP OK Xml "list of movies: URL1, URL2..."
HTTP Get <stream_url>/Manifest
200 OK – XML Manifest file
HTTP GET <stream_url>/QualityLevels(<Lower_Level>)/ Fragments(video=0)?packetpair(<packet_pair_x>)
200 OK – Chunk 0 (Bitrate <Lower_Level>)
HTTP GET <stream_url>/QualityLevels(<Lower_Level + 1>)/ Fragments(video=60000000)?packetpair(<packet_pair_y>)
200 OK – Chunk 1 (Bitrate <Lower_Level + 1>)
Real-time QoE evaluation

Fig. 2. Example of Content negotiation with the Smooth Streaming Server

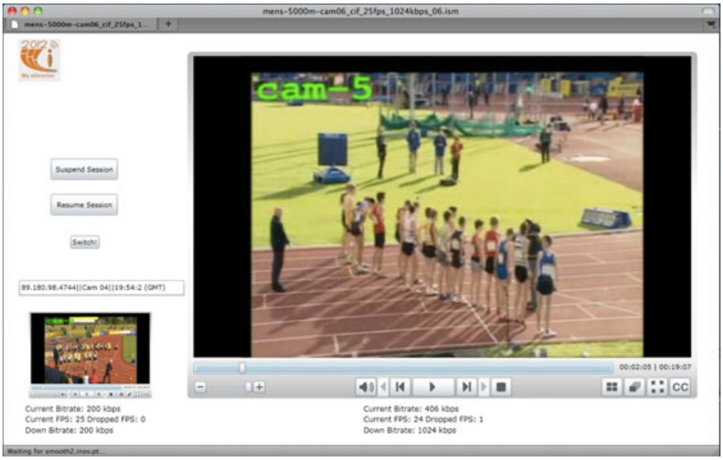
The Network Quality strategy heuristic is responsible for the network logic on determining the adequate bandwidth for an initial fast bitrate ramp up and for a subsequent sustained buffer fullness within lower and upper bitrate boundaries, constrained by a validation of supported bitrate determined by the Host Capabilities strategy.

The Host Capabilities strategy heuristic is responsible for the environment logic. Essentially, it keeps a cumulative playtime validation, on a per bitrate basis, and continuously analyses the dropped frames to infer the bitrates supported by the Client.

The Media Player Module: The Media Player Module corresponds to the media player itself, responsible for the reproduction of streams received from the Communications and Control Module. The media player can reproduce content from a variety of sources, including multiple bitrate streaming of live, video on demand H.264/AAC/MP3 and unicast HTTP Windows Media Services streams.

The User Interface Module: The User Interface Module implements the Application Graphical User Interface (GUI), dynamically generated, based on the type of terminal the Client is running.

Figure 3 illustrates the prototype UI for a desktop/laptop browser and Figure 4 for a smartphone or Personal Digital Assistant (PDA) browser. Via the GUI the user is able to authenticate in the Service Platform (SP&AAA) and also to select contents, or events to watch. The GUI also presents program Suggestions to the user based on the Personalization information.



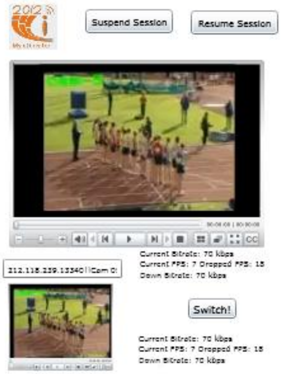
89.180.98.4744||Cam 04||19:54:2 (GMT)

Current Bitrate: 200 kbps
Current FPS: 25 Dropped FPS: 0
Down Bitrate: 200 kbps

Current Bitrate: 406 kbps
Current FPS: 24 Dropped FPS: 1
Down Bitrate: 1024 kbps

Waiting for smooth2.inov.pt...

Fig. 3. Web Media Player Client Application in a standard browser



212.118.239.13340||Cam 0:

Current Bitrate: 70 kbps
Current FPS: 7 Dropped FPS: 18
Down Bitrate: 70 kbps

Current Bitrate: 70 kbps
Current FPS: 7 Dropped FPS: 18
Down Bitrate: 70 kbps

Fig. 4. Web Media Player Client Application in a PDA browser

4 Network Portability

The Network Portability, or Vertical Handover, is assured within this architecture in a transparent manner, as long as the Client terminal provides adequate networking capabilities (multiple active interfaces, like WiFi/WiMax, 3G, etc.).

During the handover, the Web Media Player continues the playback with the "chunks" still in buffer. After the IP connection re-established (and the HTTP/TCP session automatically re-established), the Web Media Player starts requesting new "chunks", from the new connection, completely unaware of the network handover (unless the handover process takes longer than the playback of the buffered content).

5 Terminal Portability

The terminal Portability feature allows the user to suspend the current session on one terminal and resume it later on the same, or on a different terminal and independently of its type (e.g., transfer from a personal computer to a PDA).

By means of two buttons, a SUSPEND and a RESUME, the user is requested to type a code (with duplicate verification) that will be used later for resuming that session.

The Web Media Player Client exchanges XML files with the server, by HTTP POST methods, that includes the chosen code and/or the URL of the stream and the time of playback for that stream.

These RESUME requests always start with the lowest resolution/bitrate, in order to start re-playing the video as quickly as possible.

6 Conclusion and Future Work

This project is currently under development. Almost all components are implemented, but not fully tested.

The current roadmap includes both stored (video-on-demand) and live streaming media player capabilities and a full graphical design of the User Interface, the fix of eventual bugs that may exist and an exhaustive test plan (with both Functional and Performance tests), to prove the effectiveness of this solution.

Future work on this project will include the deployment of objective quality estimations at the Client and enhanced Streaming Adaptation heuristics related with physical environment status (like adaptation of resolution and zooming capabilities to light brightness, sound level and movement of user/device or display orientation of the device).

References

1. My eDirector (2012), `http://www.myedirector2012.eu`
2. ISO/IEC: Information technology – Coding of audio-visual objects – Part 14: MP4 file format. Tech. Rep. ISO/IEC 14496-14:2003 (2003)
3. ISO/IEC: Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding. Tech. Rep. ISO/IEC 14496-10:2005 (2005)
4. Moroney, L.: Introducing Microsoft Silverlight 3. Microsoft Press, Redmond (2009)
5. Pantos, R.: HTTP Live Streaming. IETF (2009)
6. Zambelli, A.: IIS Smooth Streaming Technical Overview. Microsoft Corporation (2009)