IMPROVED CHANNEL SWITCHING FOR HYBRID UNICAST/BROADCAST MOBILE TELEVISION

Stefan Diepolder and Jan Kritzner
Chair for Communication and Distributed System, RWTH Aachen University, Templergraben 55, Aachen, Germany

Keywords: Mobile TV, Switching Algorithms and Multimedia Delivery.

Abstract: Today's 3G networks enable the delivery of mobile television services to the user. However, the distribution mechanism differs from conventional television broadcasting. For an effective mobile TV system both unicast and multicast transmission are combined.

One weak point of digital television is channel switching. Both the underlying network and the data structure impose partially unavoidable delays. In this paper we present a mobile television architecture which supports fast stream switching and present different techniques to reduce at least the perceived switching duration. The different switching scenarios between unicast and multicast are discussed in detail.

This paper presents work in progress, so both already implemented and future extensions are described. Though the implementation has been done with a 3G background the algorithms can be applied to every digital television scenario.

1 INTRODUCTION AND PROBLEM STATEMENT

One important problem of digital television is channel switching. During the standardisation of today's digital television architectures the focus had been on defining the channel characteristics similar to traditional analogue TV. However, this resulted in the problem of slow switching between different channels. Especially the advent of conventional digital television via DVB-S and DVB-T (ETSI, 2004) has confronted a large user base with this problem (Knoche and McCarthy, 2005).

The next subsection will technically introduce the problem of channel switching. Afterwards, the trade-off between switching delay and video quality will be discussed. In the next section first ways to overcome the problem with advanced techniques are shown, which are refined in section 3. First results are given in the Conclusion.

Digital video compression technology is based on reducing redundancy in the video stream. This redundancy may either be spatial, i.e. within a frame, or temporal, i.e. between different frames. This temporal correlation introduces dependencies between pictures, i.e. some pictures can not be displayed without others. I-frames are independent of other frames but P-frames depend on previous frames. This backward dependency chain ends with the previous I-frame. A group of possibly dependent frames is called Group of Frames (GOF). In case of a channel switch necessary frames may not be available at the receiver, and depending ones may be distorted during playback until receiving the next I-frame of the new channel.

The next important factor is the time needed for the access onto the data of the new channel. This may either be a physical channel switch where a receiver has to be tuned to a new frequency, or some logical switch where some transport connection has to be opened.

Therefore, we see that there are two main causes for the channel switch delay:

- physical and logical transport channel switch,
- video structure.

1.1 Mobile Television Switching Architecture

For mobile television several purely broadcast-based standards like DVB-H (Faria et al., 2006) have been defined. In this document an infrastructure based on a 3GPP cellular network is proposed. We combine a Packet Switched Streaming (PSS) server (3GPP, 2005), which provides a unicast streaming service, with a Multimedia Broadcast/ Multicast Service

Diepolder S. and Kritzner J. (2008).
IMPROVED CHANNEL SWITCHING FOR HYBRID UNICAST/BROADCAST MOBILE TELEVISION.
In *Proceedings of the International Conference on Wireless Information Networks and Systems*, pages 213-218
DOI: 10.5220/0002021902130218

(MBMS) (3GPP, 2004) extension.

This way we are able to transmit frequently required channels via multicast and some less frequently requested channels directly to the particular receivers via unicast. Unicast mobile TV must have a back-channel for signaling channel selection, and the server can do the switch without the need of re-establishing the connection to the client. For broadcast channels no interaction with the server is necessary. Different multicast channels may or may not share one bearer. The temporal relation between unicast and multicast is unknown.

To speed-up the process of switching channels in a mobile TV environment it is necessary to add some extra functionality to mobile clients and to the packet switching server architecture. An algorithm for the switch from unicast to multicast may require access of the PSS server to the multicast data streams. Therefore, a unified server for multicast and unicast is necessary (see figure 1).

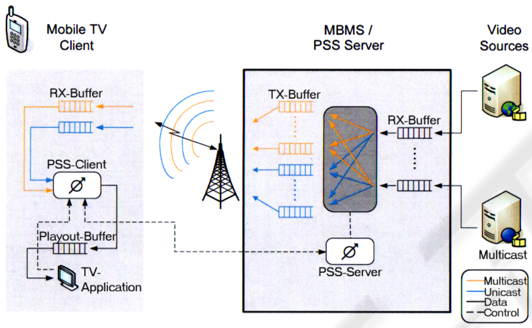

Figure 1: Proposed End-to-End Mobile TV Client/Server Architecture.

The client's architecture has to be improved, also. In addition to an existing conventional unicast PSS client an advanced multicast receiver is necessary. It consists of a multicast receiver, an audio/video demultiplexer, and a buffer. The buffer consists of separated queues for video and audio. The usage is easier given that the queues of the different channels are logically separated. However, they may dynamically share the same memory space because they are not filled simultaneously. The concepts shall be ready to be deployed at the application level because the firmware of today's mobiles is closed. However, in the long run it is possible to integrate new functions into the firmware.

The buffer needs an advanced interface to the application enabling the direct access to different parts of the stream. The normal functionality like fetching the next frame from the buffer or maybe flushing is not sufficient. Special access to different Presentation TimeStamps (PTSs) is needed for improved switching by commands like "Get PTS of next Frame", "Skip to PTS", "Get first PTS", and "Get last PTS".

For our work we are assuming that the client is capable of receiving more than one bearer at a time. At the time of writing no such mobile user equipment exists but it is planned to be available soon. Nevertheless, it can receive multicast and unicast simultaneously which is the basis of the enhanced channel switching algorithms. In our scenario the MBMS architecture may impose a delay of up to several seconds onto the TV data. However, this is not important for channel switching. There is no requirement for temporal alignment between different channels.

As shown in figure 1 the proposed architecture is combining PSS and MBMS into one. The dotted Control Channel from client application back to the PSS-Server is necessary for some switching algorithms, especially when switching from unicast to unicast and from multicast to unicast.

1.2 Switching Delays and Problem Statement

Several aspects affect the duration of a channel switch. First there is the physical time needed to switch a physical bearer. This delay occurs even when the lower layers are technically able to receive more channels simultaneously but the new channel the user wants to switch to is unknown in advance. As an example consider a switch between two multicast channels when the receiver is able to receive two multicast bearers simultaneously. Given that both channels are transmitted over one bearer the delay is zero. Given that they are transmitted over different bearers, the bearer carrying the new channel must be activated. If the old bearer is received further on, the user can zap back to the old channel with much less delay. When unicast streaming is taken into account it is possible to keep transport connections or sessions open to reduce this kind of delay.

The first part of the video decoding delay is buffering time. Due to the fact that usually the same time of the old channel has been buffered as will be buffered for the new channel, the buffering delay can be concealed by playing out the last channel. Yet, the user usually wants a swift reaction onto his channel switch command – the old content is not what the user expects. Therefore, the buffering delay for mobile television should be quite low. In case of multicast streaming, the buffering delay only needs to accommodate the different transmission delays for frames of different size. Unicast streaming protocols may benefit from higher buffering delays given that lost or corrupted data frames can be corrected by retransmissions. This results in a trade-off between the quickness of frame switching and the overall quality. The last delay which has been mentioned consists of the time between the scheduled playout of the first received video frame until the first undistorted frame can be shown. Given that after a channel switch the receiver has to begin the decoding at a random position within a GOF, the expected duration of the distortion is half the duration of a GOF. Again, there is a trade-off between video quality and the speed of switching: Long GOFs result in optimal encoding efficiency but slow switches.

2 CHANNEL-SWITCHING ALGORITHMS

Now, we propose new algorithms to speed up channel switching which are more intelligent compared to the conventional "play out what you have when you should" approach. Generally we are not trying to speed up the time until data can be received from the network but to reduce the perceived time until the presentation of the new channel starts. We will take any possible combination of switches between unicast and multicast into account, and propose algorithms tailored to the problem as much as possible. Furthermore, we will distinguish between audio and video data.

2.1 Unicast to Unicast

Given that a user switches from channel 1 to channel 2, the unicast streaming system begins the transmission with the temporal nearest I-frame which is available in the transmission buffer. Therefore the client sees some kind of virtual channel 2 which has some temporal displacement. Therefore depending on the time of their switching, two users viewing the same TV channel which stand next to each other may have a temporal displacement of up to one GOF-length in the content they view. Usually this is acceptable and people know this effect from the different delays for analogue and digital conventional television. However, users may experience it as bad service if other users see important information like goals in a soccer game some seconds earlier than others. A more important problem is the strongly increased bandwidth requirement given that a user zaps through multiple channels. Then many large I-frames are transmitted, and the bandwidth requirement exceeds normal values. Ideas to mitigate this problem are discussed in section 3.

2.2 Multicast to Multicast

In the unicast-unicast scenario it is possible to reduce the switching delay on the server side. In contrast to the unicast-unicast scenario, in a pure multicast environment the different multicast channels are simply transmitted, possibly even without knowledge how many subscribers use a certain service or how they change channels. Therefore, a purely client-based solution is necessary. A typical switching scenario depicting both the reception and playout over the time is given in figure 2. A simpler graphic depicting the playout process only can be found in the upper half of figure 4. It is compared to the conventional approach starting the video with the first I-frame and skipping undecodable frames from the first GOF. Here d_{GOF} and d_{Buf} denote the GOF and buffering duration respectively.

The new algorithm works as follows: After the switching command the receiver tunes to channel 2 (d_{Switch}). When the first I-frame of the new channel is received (after an average time of $0.5 \cdot d_{GOF}$) it is presented immediately. Until then either the playout of the previous channel may continue or the old channel freezes to indicate a pending channel switch. After the presentation of the first I-frame it remains on the screen until its normal presentation time elapses (dashed area). Then consecutive P-frames are decoded normally.

The early presentation of that I-frame may result in dropping the last P-frames of the previous channel, and the associated audio-data will be dropped. Furthermore, the first undecodable P-frames of the new channel will be dropped - but the audio data with the according timestamps will be presented. Therefore, the new channel begins with the presentation of a still image, then audio is added, and afterwards movement. This is the desired working scenario and referred to as scenario 1.

However, it is not always possible to present the first I-frame early enough. The scenario as described above only works with sufficient buffer delay, i.e. the I-frame has to be received before the playout of the previous channel ends, as depicted in figure 2. Given that the buffering delay is too low (figure 3) the old channel ends and the picture has to freeze until the first I-frame of the new stream has been received. This is referred to as scenario 2 which can be found in the lower half of figure 4. For this scenario two cases are possible: The I-frame is received sufficiently early, or audio data of the new channel could be presented before the I-frame is received. Whether to drop this audio data or not depends on psychological factors. Playing back audio without the video may irritate the viewer. However, a switch-command taking too long

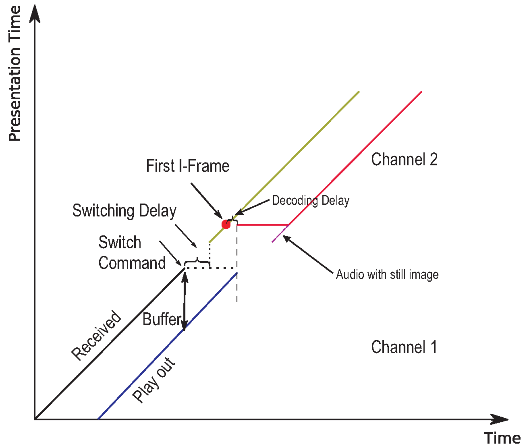

Figure 2: Switch without buffer underruns (optimal case).

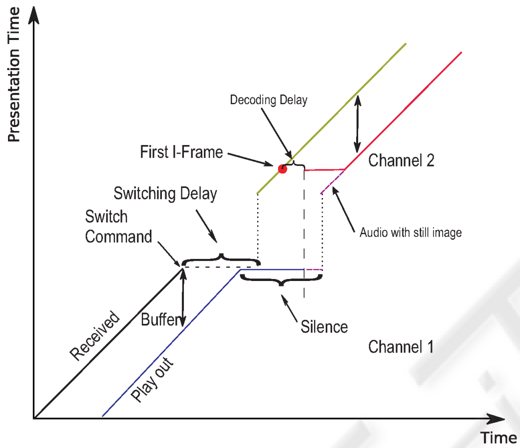

Figure 3: Switch resulting in an empty buffer due to late I-frame.

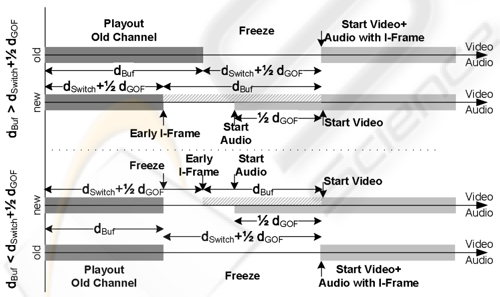

Figure 4: Multicast Switching - Simplified Playout Only.

to show any effect is perceived as bad, also.

Whether the buffering delay is sufficient for avoiding a freeze or not depends on the time of the switch. However, the probability of the desired and the suboptimal case depends on both the buffering delay and the GOF length: If the buffering delay is larger than the GOF-length plus the switching delay, the I-frame will definitely be available (first case). Otherwise the probability for this optimal case is:

$$p_1 = \frac{d_{\text{Buf}}}{d_{\text{GOF}} + d_{\text{Switch}}}.$$

After the switch, it will take on average half a GOF until an I-frame is received, hence the old channel will be shown for

$$E\left[d_{\text{Video}}\right] = \frac{d_{\text{GOF}}}{2} + d_{\text{Switch}}.$$

Then the first I-frame can be shown. Beginning with the presentation of this first I-frame there is a period of silence until the presentation time for the first audio packet has been reached:

$$E\left[d_{\text{Audio}}\right] = d_{\text{Switch}} + d_{\text{Buf}}$$

after which the normal audio playout begins, and the channel switch is completed.

Compared to switching without an algorithm there is no freeze of the old video, and audio can be presented on average half a GOF earlier than before.

The second scenario is more likely for short buffering delays compared to the GOF-length:

$$p_2 = 1 - \frac{d_{\text{buf}}}{d_{\text{GOF}} + d_{\text{Switch}}}.$$

The expected duration of the old channel freeze before the first I-frame can be shown is

$$E\left[d_{\text{Freeze}}\right] = d_{\text{GOF}} + d_{\text{Switch}} - d_{\text{buf}}.$$

2.3 Multicast to Unicast

The switch from multicast to unicast is done similar to the switch from unicast to unicast. The transport connection is kept open if possible, and the client requests the new channel from the server. Then the server starts the unicast transmission of the new channel with an I-frame which can be presented directly, and audio is added when the buffer is filled sufficiently. It is possible that the reserved bandwidth of the unicast bearer is not available immediately but only after some seconds. This problem can only be solved by over-provisioning of the unicast channel. Then the amount of buffered playout time is larger than the duration of the buffering process.

2.4 Unicast to Multicast

In the first step the switch from unicast to multicast is identical to the switch between two multicast channels. The scenario with the freeze (scenario 2) benefits when one I-frame of the new channel is transmitted before stopping the unicast bearer. Then the playout of the old channel ends with the presentation of a frame from the new channel.

3 FURTHER IMPROVEMENTS

In the latter sections simple algorithms and a basic architecture for mobile TV have been introduced. They have already been implemented. However, this paper presents work in progress. More advanced ideas and extensions are discussed now. One general technique to reduce the experienced switching delay is adaptive playout; it could be used for any switching scenario. In the following it is shown which further improvements could be made to the presented algorithms.

3.1 Unicast to Unicast

For unicast-unicast switching two improvements could be made: One on the server side, and another one by replacing the client architecture. One problem of a switch is the peak in the required bandwidth. This happens because a new channel always begins with an I-frame. Especially when a user simply zaps through different channels quickly, the reserved rate of the bearer would not be sufficient. This could be solved by encoding specialised switching GOFs with an I-frame of reduced quality, and consecutive P-frames with better quantisers gradually improving the video quality towards the end of the GOF. When a new channel is requested the server transmits such a switching GOF at the beginning, and afterwards continues with the presentation of the normal TV channel. The lower quality of the beginning of the first GOF reduces the necessary bandwidth of a switching event, and mitigates the effects of channel hopping.

Another approach to achieve the same goal could be the usage of switching P-frames for the channel switch instead of I-frames. However, it is not guaranteed that there is enough correlation between the old and the new channel to enable a substantial reduction in the required bandwidth.

On the client side advanced control over the playout buffer would be beneficial. Especially the detection of the transition between one channel and the other is important for advanced playout control, e.g. a quick presentation of the first I-frame of the new channel as done by a multicast-multicast switch. Up to now the switching duration is identical to the buffering time.

3.2 Multicast to Multicast

Given that the client is able to receive more than one multicast channel at a given time it could be beneficial to receive the data of both channels simultaneously. For example, conventional TV sets often have a special button to go back to the previous channel. Given that both of these channels are received such a channel switch could be performed much faster. A more complicated scenario would use a unicast connection to support channel switches. When the user changes from one multicast channel to the other the server pushes one I-frame of the new channel to the client. However, this technique has problems given that the reserved bandwidth of the unicast bearer is not available fast enough.

3.3 Multicast to Unicast

For multicast to unicast switches a good initial solution had already been found. The addition of adaptive playout to this scenario would result in further improvements.

3.4 Unicast to Multicast

When the user switches from a unicast to a multicast channel the server appends an I-frame of the new channel to the old data stream which is eventually (scenario 2) played out to give a first impression of the program. For a quicker reaction it would be good to present this I-frame as soon as it is available at the client and not after the buffer has been played out. This is only possible if the advanced buffer control with direct data access is available.

Even more interesting would be the transmission of some kind of prefix data which would enable the client to seamlessly switch to the new channel, e.g. by making the first received P-frames of the multicast transmission decodable. However, due to the possibly high difference in the forward trip times of the unicast and the multicast system some kind of synchronisation technique is necessary determining which data of the new multicast channel will be missing.

4 CONCLUSIONS

One of the main problems of today's mobile TV systems, slow channel switching, has been analysed, and a mobile TV system is developed to achieve a faster channel switch. The different alternatives for the transmission of mobile TV have been compared, and the causes for slow channel switching have been analysed. To accelerate the switching process, algorithms have been proposed which are implemented in a mobile TV system using enhanced buffer management.

Slow channel switching is caused by the structure of digital video, and this cannot be solved because up to now it is the only way to achieve good compression ratios. The aim was to attenuate the effect of the video structure, and to improve the perceived quality.

The proposed algorithms for unicast and multicast switching gain an improvement regarding the switch delay (multicast) or resource utilisation (unicast) respectively. Moreover, the switching algorithms guarantee a switch without distortions in the presented video by avoiding decoding useless P-frames. The search for the first I-frame is done quickly when the unicast switching algorithm is used, and it introduces no relevant delay regarding the channel switch.

The proposed archicture and the algorithms have been implemented on top of the Network-Integrated Multimedia Middleware (NMM) (Lohse, 2005), and will be evaluated. NMM is a framework for media processing building flow graphs through special nodes. Due to the object oriented design the nodes are extensible. To build our system we have added some functionality to the framework and especially to the Switching Node, which implements the different switching algorithms, the Media Source Node which feeds the video content into the graph, and the display node, that shows the result of the delivered video at the end of the switching process.

The delay of multicast-multicast switches is dominated by waiting for the first I-frame of the new channel. Therefore, a fast channel switch is possible if the I-frame is received early. The proposed multicast switching algorithm improves every case because I-frames are presented as fast as possible. The gain due to this immediate presentation roughly matches the buffering delay, in initial simulations we measured an average gain of 0.8 seconds, which is a noticeable improvement over the simple switching. Despite this improvement, a multicast switch takes at the average 2.8 seconds which is caused by the long GOF structure of 4 seconds for our simulations.

REFERENCES

3GPP (2004). *Multimedia Broadcast/Multicast Service; Stage 1 Release 6, 3GPP TS 22.146 V6.6.0*. 3GPP.

3GPP (2005). *Transparent End-To-End Packet-Switched Streaming Service (PSS): Protocols and Codecs (Release 6), 3GPP TS 26.234 V6.3.0*. 3GPP.

ETSI (2004). *Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television (DVB-T), EN 300 744 V1.5.1*. European Telecommunications Standards Institute.

Faria, G., Henriksson, J. A., Stare, E., and Talmola, P. (2006). DVB-H: Digital Broadcast Services to Handheld Devices. *Proceedings of the IEEE*, 94(1):194–209.

Knoche, H. and McCarthy, J. D. (2005). Design requirements for mobile TV. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 69–76, New York, NY, USA. ACM Press.

Lohse, M. (2005). *Network-Integrated Multimedia Middleware, Services, and Applications*. PhD thesis, Department of Computer Science, Saarland University, Germany.