

A SIMULATION FRAMEWORK FOR THE EVALUATION OF THE FILE REPAIR MECHANISM USED IN IPDC/DVB-H

Bernhard Hechenleitner

*Salzburg University of Applied Sciences
Urstein Süd 1, 5412 Puch / Salzburg, Austria
bernhard.hechenleitner@fh-salzburg.ac.at*

ABSTRACT

The rollout of digital television systems is taking place in many countries worldwide. Utilizing the advantages of digital technologies, the special characteristics of mobile devices – such as small screen sizes and low power supplies – can be addressed, thus enabling mobile television and data services. IP Datacast (IPDC) based on Digital Video Broadcasting – Handhelds (DVB-H) is one of the technologies which provide the broadcasting of audio and video streams as well as the distribution of arbitrary binary data objects. For the lossless transmission of binary data objects, the IPDC/DVB-H specifications define a mechanism for the indication and correction of erroneous broadcast data, called file repair mechanism. This study evaluates the performance of this file repair mechanism utilizing a simulation framework which integrates all relevant models and mechanisms including a state-of-the-art error model for the transmission channel. Based on the setting of different parameters such as the use of forward error correction and the length of encoding symbols, performance evaluations of different file repair strategies have been accomplished, resulting in some recommendations for the setting of critical transmission parameters.

KEYWORDS

IPDC; DVB-H; FLUTE; MPE-FEC; File repair.

1. INTRODUCTION

Digital Video Broadcasting – Terrestrial (DVB-T) (Reimers 2005) has been rolled out in many parts of the world as the technological successor of analog television. As this standard is directed to the broadcast of continuous high bit rate data corresponding to the screen sizes of home television sets, adoptions were necessary for transferring this technology towards mobile devices, which are restricted by certain limitations. Therefore, Digital Video Broadcasting – Handhelds (DVB-H) (ETSI 2004, Faria 2006) was specified as an extension of DVB-T focused on small mobile devices with their specific properties such as small screen sizes and low power supplies. Around this specification, a set of mechanisms and protocols was defined for enabling mobile television services, including their composition, distribution and charging. This set of mechanisms and protocols is defined as IP Datacast (IPDC) over DVB-H (ETSI 2006a) and it comprises specifications defining architectures and protocols for the distribution of video, audio and data services built upon DVB-H. One of these specifications defines the Content Delivery Protocols (CDP) (ETSI 2006b) necessary for the transmission of time-critical streaming sessions such as live mobile television. Furthermore, CDP defines protocols for the distribution of arbitrary binary data objects as well as a mechanism for the indication and correction of erroneous broadcast data, called file repair mechanism. This study focuses on the broadcast transmission of time-uncritical binary data objects, such as applications, images, ring tones, or other data using an existing IPDC/DVB-H infrastructure. These data objects need to be delivered lossless, therefore the transmission relies on the IPDC/DVB-H file repair mechanism. The purpose of this study is to implement a simulation framework concerning the file repair mechanism and evaluate its effectiveness with respect to several transmission parameters.

Each IPDC service is broadcast to a specific IP multicast address via DVB-H by multiplexing the data into the MPEG-2 Transport Stream (TS) of the underlying DVB-T infrastructure (Reimers 2005). For a

reliable transmission of this IP multicast data, the Asynchronous Layered Coding (ALC) (Luby 2002) protocol – one of two IETF specifications dealing with reliable multicast transport – has been chosen. The transmission of binary data objects and their metadata descriptions is conducted by the File Delivery over Unidirectional Transport (FLUTE) (Paila 2004) protocol, which is built upon ALC and uses UDP as transport protocol. Each FLUTE session consists of transmitting some metadata about the objects to be distributed – called File Delivery Table (FTD) – and the transmission of the binary objects themselves. Examples for binary objects to be delivered within a FLUTE session via IPDC/DVB-H are the Electronic Service Guide (ESG), applications, images, ring tones, or other data. As an additional level of transmission reliability, a file repair mechanism for IPDC/DVB-H systems has been defined at the application level, which offers receivers the possibility to inform the sender about lost or not recoverable data fragments (ETSI 2006b, DVB Project 2007). Using this file repair mechanism, receivers inform the sender about erroneous (i.e. lost or unrecoverable) data fragments. Based on this feedback, the sender can decide to either repeat the erroneous fragments individually for each receiver – which is called a point-to-point (ptp) repair session – or to use the broadcast channel for sending the requested fragments to all receivers, conducting a point-to-multipoint (ptm) repair session.

For the examination of different aspects of this file repair mechanism, a simulation framework has been developed. The simulation framework is based on the open source simulation tool OMNeT++ (Varga 2008) and implements current error models for the wireless transmission channel (Poikonen 2006, Poikonen 2005) as well as procedures concerning additional forward error correction measures as defined in (ETSI 2005). The formulas in (ETSI 2006b) constitute the basis for ptp and ptm repair cost comparisons. Based on this simulation framework, the following research questions have been examined:

- what are the transmission overheads caused by repair sessions for different numbers of receivers?
- what are the differences between ptp and ptm repair sessions concerning transmission overheads?
- what are the effects of additional forward error correction measures concerning transmission overheads?
- what are the resulting repair costs?
- what are the influences of additional forward error correction measures and different encoding symbol sizes on the repair costs?

The document is structured as follows. Chapter 2 describes the most important mechanisms and models relevant for the simulation framework. In chapter 3, the developed simulation framework is described in detail. The simulation parameters used for the different simulation runs are shown in chapter 4. The results of specific simulation experiments are presented in chapter 5, and chapter 6 concludes this study.

2. RELEVANT MECHANISMS AND MODELS

The mechanisms and models described in this chapter represent the technological basis for the simulation framework. These include the IPDC/DVB-H file repair mechanism including file repair cost calculation, the error model for the transmission channel, and MPE-FEC as a means for forward error correction.

2.1 File repair mechanism

The purpose of the IPDC/DVB-H file repair mechanism is the correction of lost or unrecoverable data fragments of a data transmission session. To fulfill this task, at least one file repair server is responsible for accepting file repair requests of the single receivers of the session. The indicated fragments can be sent to the requesters either directly via ptp repair using existing 3G or WLAN connections or via ptm repair using the same IPDC/DVB-H infrastructure which was used for the primary data transmission session.

2.1.1 Basic Procedure

The basic procedure of the IPDC/DVB-H file repair mechanism is as follows (ETSI 2006b):

1. The receiver identifies erroneous data fragments.
2. Concerning the transmission of file repair requests a time window is defined, which begins after the data session has ended. During this window the receivers send their requests at random times. Therefore, each receiver calculates a random time value (back-off time).

3. After completion of the file transmission and expiration of the back-off time the receiver sends its file repair requests to the file repair server.
4. The file repair server replies with a repair response message, which either contains the requested data fragments (ptp) or the session description for the ptm repair session.

All file repair request messages should be transmitted via a HTTP 1.1 GET request (Fielding 1999). If more than one GET request is necessary, they should be sent without intermediary waiting times. Erroneous data fragments are specified by their Source Block Numbers (SBN) and Encoding Symbol IDs (ESI). In case of a ptp file repair session, all requested fragments are sent to the receivers via HTTP responses using existing 3G or WLAN connections. In case of a ptm file repair session, all requested fragments are sent via a FLUTE file delivery session based on IPDC/DVB-H.

2.1.2 Cost calculation

In the context of this study the costs for the first rounds of file repair sessions are compared. For ptp repair sessions one round should be sufficient. For ptm sessions further rounds could be necessary until all receivers have received complete and error-free data.

During the time window for file repair requests, the file repair server can estimate the expected repair costs. The request window consists of a predefined, fixed value and a random part, the maximum back-off time. The calculation of the estimated costs is done at a fraction of the maximum random part of the repair request window, defined by the parameter α . Based on the repair requests received until this point in time, the file repair server projects the expected ptp and ptm repair costs. After expiration of the time window for file repair requests, the total cost for the first round of a file repair session can be calculated. The costs for a ptp repair session are specified by the number of file repair requests, number of requested symbols, symbol size, and transmission cost per byte. For a ptm repair session, which uses a broadcast channel for sending repair symbols, the number of distinctively requested symbols is the most relevant parameter concerning the resulting repair costs.

$$C_{ptm} = c_m \cdot s_{an} + c_u \cdot N_{req} \cdot s_{req} + c_m \cdot n_{dsym} \cdot s_{sym} \quad (1)$$

All formulas necessary for calculating the expected and final ptp and ptm repair costs are defined in (ETSI 2006b). Concerning the calculation of the expected and final ptm repair costs, a modified formula (1) is proposed, which focuses on the characteristic that only distinctive erroneous symbols have to be retransmitted in a ptm repair session. The cost of the transmission of a single byte via the used ptm and ptp networks are given by c_m and c_u . The size of a ptm repair session announcement is defined by s_{an} , the number of repair request messages is given by N_{req} , and s_{req} specifies the average size of a repair request message in byte. The number of distinctive requested symbols is specified by n_{dsym} . The average size of a symbol in byte is defined by s_{sym} .

2.2 Error model for the transmission channel

The error model of the transmission channel used for this study is based on the four-state run length model described in (Poikonen 2006) and (Poikonen 2005). This model operates on the resulting MPEG-2 TS of the DVB-T data transmission and produces streams of erroneous TS packets by using the four states *Good (short)*, *Bad (short)*, *Good (long)* and *Bad (long)*. If the model is in the states *Bad (short)* or *Bad (long)*, erroneous TS packets are generated, and if the model operates in the other two states, correct TS packets are produced.

According to the state diagram of this model (Poikonen 2006), the probabilities for remaining in a state denoted as *long* are very high. Therefore, whenever the error model switches over to one of these states, it will remain there for a rather long period of time compared to the states denoted as *short*, leading to long sequences of erroneous or error-free TS packets. With these settings, the model has shown to produce a good approximation to error streams measured using the COST 207 Six-tap Typical Urban (TU6) multi-path channel model (COST 1989). See (Poikonen 2006) for a comparison.

2.3 Multi-protocol encapsulation forward error correction

The procedure for splitting a binary data object such as a file into data fragments for transmission via the IPDC/DVB-H infrastructure is as follows (ETSI 2004, ETSI 2006b). First, a blocking algorithm to split the binary object into source blocks and encoding symbols is applied. The blocking algorithm used depends on the used forward error correction mechanism at the application layer (AL-FEC). If no AL-FEC is used (as was done within this study), the binary object is simply split into several source blocks, with each source block consisting of several encoding symbols. The granularity of this splitting depends on the chosen encoding symbol size. Next, each encoding symbol is encapsulated in a FLUTE/UDP segment, and the resulting FLUTE/UDP segments are encapsulated in IP multicast packets. Then, these IP packets are encapsulated in Multi-protocol Encapsulation (MPE) sections and further mapped to the corresponding MPEG-2 transport stream provided for this service on the DVB-T infrastructure. Finally, if additional Multi-protocol Encapsulation Forward Error Correction (MPE-FEC) is applied, an internal buffer called MPE-FEC frame is used to calculate some correction data for IP packets buffered column by column in the Application Data Table (ADT) of the MPE-FEC frame. A MPE-FEC frame consists of 191 columns for the ADT, 64 columns for the calculated parity data, and either 256, 512, 768, or 1024 rows. Therefore, the size of a MPE-FEC frame either is 47.75 KB, 95.5 KB, 143.25 KB, or 191 KB. The error correction code used by MPE-FEC is the reed solomon block code RS(255,191,32). For a more detailed explanation of the used reed solomon block code see (ETSI 2005) and (Reimers 2005).

3. SIMULATION FRAMEWORK

The simulation framework designed and implemented uses the discrete, event-based simulation tool OMNeT++ (Varga 2008). Using this simulation tool, components and topologies are described using the programming language NED (Varga 2008). Simulation components have to be implemented in C++.

Figure 1 shows the general topology used for conducting the simulations. As an example, a topology with four receivers is shown. Two of them send file repair requests to the file repair server. The component *Sender* represents the IPDC/DVB-H sender, which broadcasts binary data objects to the receivers. The broadcast transmission is represented by the component *PacketBroadcast*. This component simply replicates a packet sent by the sender for each receiver. The component *Receiver* incorporates the error model for the transmission channel and the optional MPE-FEC error correction. Finally, the component *FileRepairServer* receives file repair requests from the receivers and conducts cost calculations (see 2.1.2).

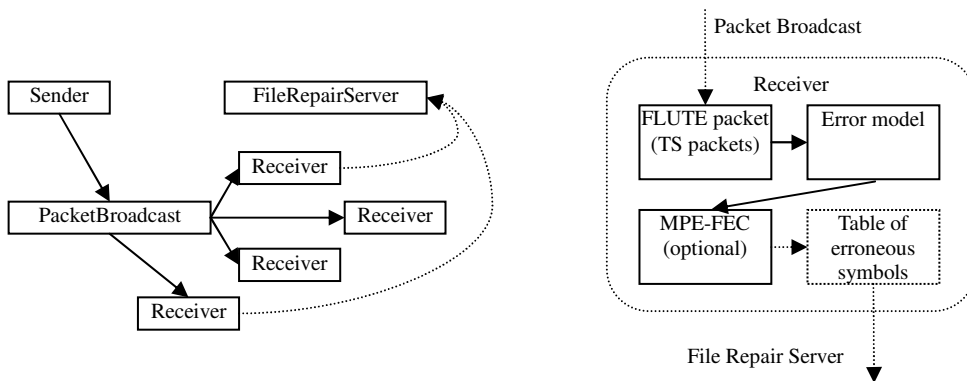


Figure 1. Simulation topology

Figure 2. Details of the component *Receiver*

The following parameters can be set for the component *Sender*: size of the data object to be transmitted (file size), size of an encoding symbol (each encoding symbol is the payload of a FLUTE packet), number of encoding symbols per source block, sum of additional protocol overheads (FLUTE/ALC, UDP, IP), and MPEG-2 TS data rate.

The component *Receiver* allows the configuration of the following parameters: file size, encoding symbol length, number of encoding symbols per source block, number of rows of the MPE-FEC frame (optional), DVB-T transmission mode (2K, 4K or 8K), minimal distance from sender in km, offset time and random time period for sending file repair requests after the transmission has finished. Figure 2 shows the most important parts of the component *Receiver* in more detail. Each encoding symbol is transported within a FLUTE packet, which itself is mapped to TS packets of the used MPEG-2 TS. The error model described in chapter 2.2 is applied to the received stream of TS packets. Depending on the states of the error model, this results in no or several erroneous TS packets. If MPE-FEC is used, some or all of these erroneous TS packets can be recovered, depending on the size of the MPE-FEC frame. Unrecoverable encoding symbols are listed in the table of erroneous symbols and can be requested at the file repair server after the transmission of the file has finished.

The following parameters can be set for the component *FileRepairServer*: number of receivers, cost for the transmission of a single byte via the ptp network, cost for the transmission of a single byte via the ptm network, offset time and maximum back-off time for receiving file repair requests after the transmission has finished, fraction of the maximum back-off time for the calculation of estimated repair costs (α), file size, average symbol size, average size of a file repair request, size of a ptp repair session announcement, estimated success rate for a receiver using a ptp repair session, and fraction of receivers without a ptp connection.

4. SIMULATION PARAMETERS

Several simulation runs have been executed. Table 1 shows the parameters of the different simulation runs. For all runs, the size of the transmitted object was 4 MB. This is for example the typical size of an mp3 sound file. The size of an encoding symbol either was 500 byte, 1400 byte or 100 byte. This allowed for the comparison of the effects of very small, medium-sized, and large FLUTE packets on the number of necessary repair requests. MPE-FEC was either disabled or enabled, using 256, 512 or 1024 rows for the corresponding MPE-FEC frames. With these settings, it was possible to compare the effectiveness of different MPE-FEC settings with regard to the resulting number of repair requests and therefore with regard to the resulting transmission overheads and repair costs.

Furthermore, the parameters shown in table 2 were set to fixed values. The values for the DVB transmission mode and TS data rate are typical, technical parameters of current DVB-H trial services (DVB Project 2008). The cost for the transmission of a single byte via the ptm network was based on the pricing published by a regional DVB-H provider that bills EUR 8300 excluding 20% VAT per 100 kbps stream per month. The cost for the transmission of a single byte via the ptp network was based on the pricing published by a regional mobile network provider that bills EUR 22 including 20% VAT per 3 GB per month. After one fifth of the maximum back-off time ($\alpha = 0.2$), a cost estimation concerning ptp and ptm file repair session costs was done by the file repair server. The remaining values of this table were chosen according to examples in (ETSI 2006b) and (DVB Project 2007). The number of receivers was the variable simulation parameter for each simulation run.

Table 1. Simulation parameters

File size [MB]	Symbol size [Byte]	MPE-FEC
4	500	disabled
		enabled, 256 rows
		enabled, 512 rows
1400	1400	enabled, 1024 rows
		disabled
		enabled, 512 rows
100	100	enabled, 1024 rows
		disabled
		enabled, 512 rows

Table 2. Fixed simulation parameters

Component	Parameter	Value
Sender	Maximum source block length	100 symbols
Sender	TS data rate	9.95 Mbps
Receivers	DVB transmission mode	8K
FileRepairServer	Transmission cost of 1 octet ptp	EUR 7.33e-9
FileRepairServer	Transmission cost of 1 octet ptm	EUR 3.07e-7
FileRepairServer	Offset time	50 s
FileRepairServer	Maximum back-off time	600 s
FileRepairServer	α	0.2

5. SIMULATION RESULTS

Using the simulation framework depicted in chapter 3, parameterized with the simulation parameter values as described in chapter 4, several simulation studies of the first rounds of ptp and ptm repair sessions have been conducted, depending on the number of receivers. The simulations focused on the research questions described in section 1. Exemplary simulation results are shown in this chapter.

Figure 3 shows the resulting transmission overhead relative to the transmitted file size of 4 MB, comparing ptp repair sessions with disabled MPE-FEC and MPE-FEC with 256 rows in figure 3a, and comparing ptm repair sessions with the same parameters in figure 3b. As can be seen, the ptp transmission overhead is several hundredfold higher than the ptm transmission overhead. Whereas the overhead for ptp sessions rises linearly with the number of receivers, the transmission overhead for ptm sessions quickly converges to a maximum. This maximum represents the correction of the whole file by simply sending it again using the ptm network.

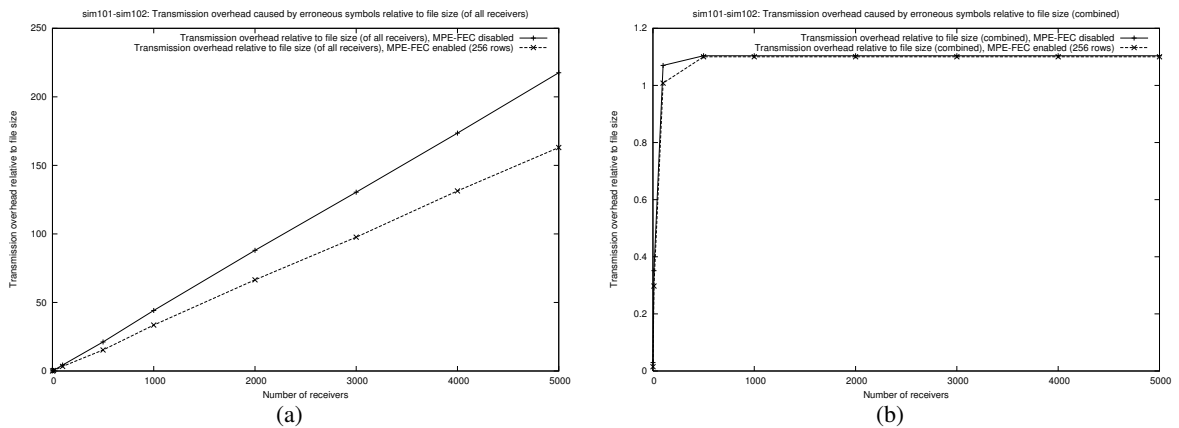


Figure 3. Transmission overhead

The results of the studies on the impact of MPE-FEC for forward error correction are shown in figure 4. Figure 4a compares the resulting amount of necessary repair data for ptp repair sessions depending on the used MPE-FEC frame size. As can be seen clearly, the bigger the MPE-FEC frame size, the lower the amount of erroneous data. Concerning ptm sessions, the amount of repair data converges to the file size including protocol overheads. The bigger the used MPE-FEC frames, the more slowly this convergence happens (see figure 4b).

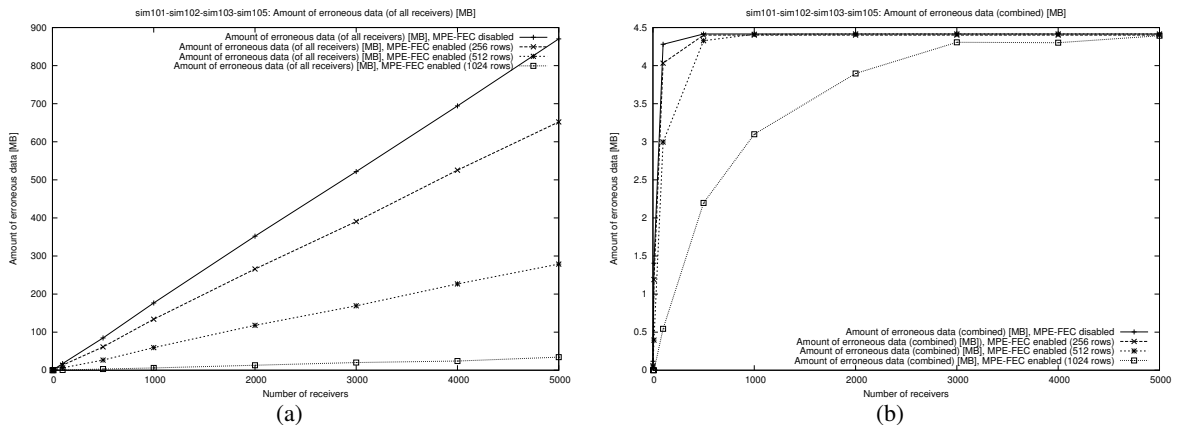


Figure 4. Effect of MPE-FEC

Figure 5a compares the expected and final file repair costs for ptp repair sessions with and without MPE-FEC. The costs for ptp sessions rise almost linearly with the number of receivers. The bigger the MPE-FEC frame size, the lower the repair costs. Furthermore, it can be seen that the expected costs are very close to the final costs, although the calculation of the expected costs was done at only a fifth of the repair request

window ($\alpha = 0.2$). Figure 5b shows the comparison of ptp repair costs using different symbol sizes. The usage of a symbol size of 500 byte yields better results than using a bigger symbol size of 1400 byte. The results in figure 6a show that also a very small symbol size of 100 byte increases the ptp repair costs, compared to a symbol size of 500 byte. Therefore, it is recommended to use medium-sized symbols. As can be seen in figure 6b, very small symbols result in higher ptm repair costs as well. This is due to the disproportion of data to be transmitted and protocol overheads.

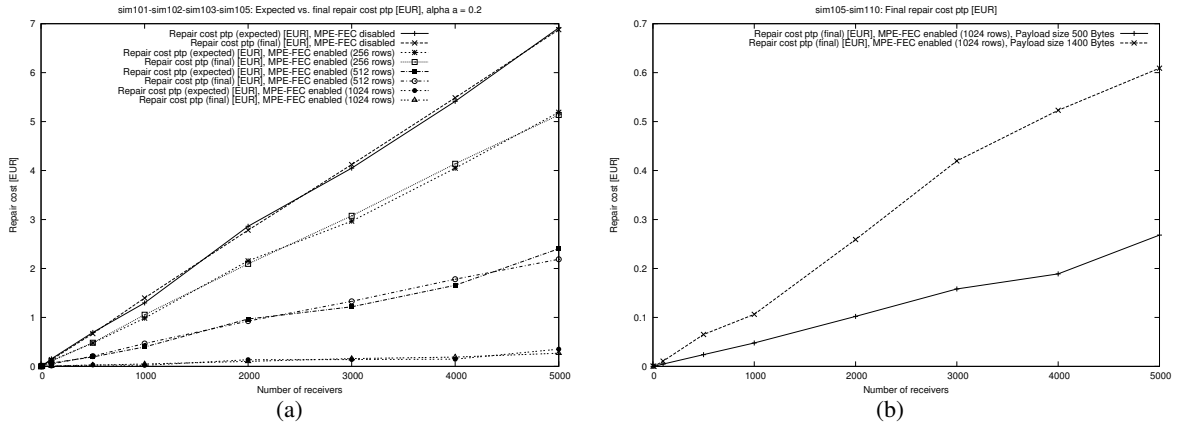


Figure 5. Repair costs ptp

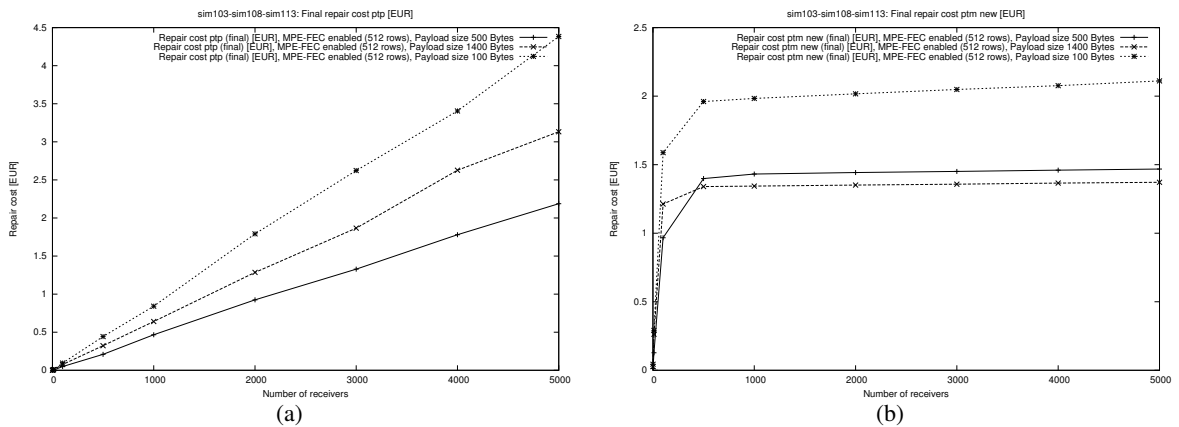


Figure 6. Repair costs depending on the symbol size

6. CONCLUSIONS

Within the course of this study, the IPDC/DVB-H file repair mechanism used for correcting erroneous (i.e. lost or unrecoverable) data fragments was examined. This examination was done using a simulation framework, which incorporates the basic IPDC/DVB-H transmission and file repair mechanisms as well as a current error model concerning the DVB-H transmission channel. Several simulation runs concerning the transmission of a file to a variable number of receivers were executed with varying values for several critical parameters such as the use of MPE-FEC for error correction, the size of MPE-FEC frames, or the size of single encoding symbols.

The simulations focused on several issues of interest, such as the resulting transmission overhead of point-to-point and point-to-multipoint repair sessions, the effects of MPE-FEC as an additional level of error correction on the amount of necessary repair data, and the file repair costs using different symbol sizes and MPE-FEC frame sizes. The results of the simulations provided some interesting findings about the setting of

several transmission critical parameters. Some of the most important conclusions with respect to the research questions (see section 1) are as follows:

- The transmission overhead due to repair data is significantly higher for ptp repair sessions than it is for ptm sessions. Whereas this overhead rises linearly with the number of receivers for ptp sessions, it quickly converges to the retransmission of the sent data object for ptm sessions.
- The application of MPE-FEC drastically reduces the amount of repair data necessary for data recovery, especially concerning ptp repair sessions. The bigger the MPE-FEC frame size, the lower the amount of necessary repair data.
- The file repair costs of ptp repair sessions rise (almost) linearly with the number of receivers, whereas the file repair costs of ptm repair sessions rapidly converge to the costs of retransmitting the whole file.
- The bigger the used MPE-FEC frame, the lower the rising of file repair costs, especially concerning ptp repair sessions. The biggest MPE-FEC frame size (1024 rows) provides the best results. Depending on the number of receivers, ptp repair costs can be considerably lower than ptm repair costs, if MPE-FEC is used.
- The size of the encoding symbols has a strong impact on the repair costs. Neither a very big symbol size, nor a very small symbol size leads to optimal results. Medium-sized symbols (e.g. 500 byte) are recommended.

The IPDC/DVB-H file repair mechanism allows for the calculation of expected repair costs before the repair request window has expired. Simulations showed that the correctness of the expected costs was high, even when the calculation was done within only a small fraction of the whole repair request window.

ACKNOWLEDGEMENT

This study was supported by the Austrian RTR project “Mobile TV Austria”.

REFERENCES

- COST, 1989. *Digital land mobile radio communications*. Commission of the European Communities.
- DVB Project, 2007. *IPDC over DVB-H: Content Delivery Protocols (CDP) Implementation Guidelines* (DVB BlueBook A113).
- DVB Project, 2008. *DVB-H: Global Mobile TV*. [online]. [Accessed 11th April 2008]. Available at: <<http://dvb-h.org/>>.
- ETSI, 2004. *Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)* (ETSI EN 302 304 V1.1.1).
- ETSI, 2005. *Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines* (ETSI TR 102 377 V1.2.1).
- ETSI, 2006a. *Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Architecture* (ETSI TR 102 469 V1.1.1).
- ETSI, 2006b. *Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols* (ETSI TS 102 472 V1.2.1).
- Faria, G. et al., 2006. DVB-H: Digital Broadcast Services to Handheld Devices. *Proc. IEEE*, Vol. 94, No. 1, pp. 194-209.
- Fielding, R. et al., 1999. *Hypertext Transfer Protocol -- HTTP/1.1* (IETF RFC 2616). The Internet Society.
- Luby, M. et al., 2002. *Asynchron Layered Coding (ALC) Protocol Instantiation* (IETF RFC 3450). The Internet Society.
- Paila, T. et al., 2004. *FLUTE - File Delivery over Unidirectional Transport* (IETF RFC 3926). The Internet Society.
- Poikonen, J. and Paavola, J., 2005. *Comparison of Finite-State Models for Simulating the DVB-H Link Layer Performance*. Technical Report, University of Turku.
- Poikonen, J. and Paavola, J., 2006. Error Models for the Transport Stream Packet Channel in the DVB-H Link Layer. *Proceedings of the IEEE International Conference on Communications 2006 (ICC '06)*, Vol. 4, pp. 1861-1866.
- Reimers, U., 2005. *DVB – The Family of International Standards for Digital Video Broadcasting*. Springer, Berlin Heidelberg New York.
- Varga, A., 2008. *OMNeT++ Community Site*. [online]. [Accessed 11th Apr. 2008]. Available at: <<http://www.omnetpp.org/>>.