

# Repair Costs of the IPDC/DVB-H File Repair Mechanism

Bernhard Hechenleitner  
Salzburg University of Applied Sciences  
bernhard.hechenleitner@fh-salzburg.ac.at

## Abstract

*IP Datacast (IPDC) based on Digital Video Broadcasting – Handhelds (DVB-H) is one of several technology candidates for mobile television. Besides the broadcasting of audio and video streams, IPDC/DVB-H also offers the distribution of arbitrary binary data objects. This paper examines the IPDC/DVB-H specific file repair mechanism, which deals with the correction of erroneous fragments of transmitted data objects, and gives recommendations for providers concerning the setting of several transmission parameters to achieve minimal file repair costs. For this purpose, a simulation framework which integrates all relevant models and mechanisms including a state-of-the-art error model for the transmission channel has been designed and implemented. Based on the setting of different parameters such as error correction and the length of encoding symbols, comparisons of resulting financial costs of different file repair strategies have been accomplished.*

## 1. Introduction

Digital Video Broadcasting – Handhelds (DVB-H) [1,2] is an extension of the DVB – Terrestrial (DVB-T) [3] standard focused on small mobile devices with specific properties such as small screen sizes and low power supplies. IP Datacast (IPDC) [4] comprises a set of 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) [5] necessary for the transmission of time-critical streaming sessions and for the distribution of binary data objects. IPDC systems based on DVB-H are called IPDC/DVB-H systems.

Each IPDC service is broadcast to a specific IP multicast address via DVB-H. For a reliable transmission of this IP multicast data, the

Asynchronous Layered Coding (ALC) [6] 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) [7] protocol, which is built upon ALC and uses UDP as transport protocol. Each FLUTE session consists of the transmission of 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), images, ring tones, applications or other data.

As an additional level of transmission reliability, a file repair mechanism has been defined at the application level, which offers receivers the possibility to inform the sender about lost or not recoverable data fragments. This kind of feedback implies the existence of a feedback channel such as a 3G or WLAN data connection.

This study examines the file repair mechanism defined for IPDC/DVB-H systems [5,8] on the basis of a simulation framework. Using this file repair mechanism, receivers inform the sender about erroneous (i.e. lost or unrecoverable) data fragments. Based on this feedback and on file repair cost calculations, the sender can decide to either retransmit 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. Using the developed simulation framework, this study compares the costs of ptp and ptm file repair sessions depending on the number of receivers, the file size, and several other parameters like Forward Error Correction (FEC) measures, the size of the single data fragments (encoding symbols), and others. The simulation framework developed is based on the open source simulation tool OMNeT++ [9]. Current error models

for the wireless transmission channel [10,11] were considered for the modeling of the framework. The formulas in [5] constitute the basis for the cost comparisons.

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 results of specific simulation experiments are presented in chapter 4, and chapter 5 concludes this study.

## 2. Relevant mechanisms and models

The following mechanisms and models are the basis for the design of the simulation framework:

- IPDC/DVB-H file repair mechanism,
- file repair cost calculation formulas,
- MPE-FEC, and
- error model for the transmission channel.

### 2.1. File repair mechanism

The file repair mechanism belongs to the group of so-called associated delivery procedures, defined for the purpose of delivering data in DVB-H based IPDC systems [5]. The associated delivery procedures specify the processes of reporting reception of data fragments and conducting file repairs. The basic principle of the file repair mechanism is the correction of lost or unrecoverable data fragments. For accomplishing this task, one or more file repair servers receive feedback of the receivers of the data transmission session and either repair the erroneous data individually via ptp repair using existing 3G or WLAN connections or use the IPDC/DVB-H infrastructure for a ptm repair.

The simplified procedure of the file repair mechanism is as follows [5]:

1. The IPDC/DVB-H receiver identifies erroneous data fragments.
2. The IPDC/DVB-H receiver calculates a random time value (back-off time).
3. After completion of the file transmission and expiration of the back-off time the IPDC/DVB-H receiver sends a repair request message 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 data necessary for accessing a ptm repair session.

The back-off time is the sum of a predefined offset value and a random value, individually calculated for each receiver. Generally, for each transmission a time

window, starting after the transmission is finished, is defined. During this repair request window, the receivers send their file repair requests to the file repair server. As each receiver calculates a random time value for sending the requests, the reception of these requests should be distributed across the whole window.

A file repair request message should be transmitted via a HTTP 1.1 GET request [12]. 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). As an example, the request shown in figure 1 tells the file repair server that encoding symbol number 5 from source block 7 and encoding symbol number 3 from source block 15 of the transmitted file “news.3gp” hosted at “example.com” were erroneous.

```
GET /file_repair?fileURI=example.com/news.3gp&SBN=7;ESI=5
&SBN=15;ESI=3 HTTP/1.1
```

**Figure 1. File repair request message**

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. Figure 2 shows a schematic example of a ptp file repair response message.

| <i>HTTP header</i>                  |   |                    |
|-------------------------------------|---|--------------------|
| Length of group (number of symbols) | SBN and ESI of first symbol of this group | Encoding symbol(s) |
| Length of group (number of symbols) | SBN and ESI of first symbol of this group | Encoding symbol(s) |
| ...                                 | ...                                       | ...                |
| Length of group (number of symbols) | SBN and ESI of first symbol of this group | Encoding symbol(s) |

**Figure 2. File repair response message (ptp)**

In case of a ptm file repair session, all requested fragments are sent via a FLUTE file delivery session based on IPDC/DVB-H.

### 2.2. 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. The formulas described in this section are based on [5] and they are valid for setups using one file repair server only.

During the time window for file repair requests, the file repair server can estimate the expected repair costs. Formula (1) defines the point in time this estimation is calculated. The parameter  $t_{end}$  defines the point in time when the file transmission is complete and  $t_{offsetTime}$  is a fixed parameter which specifies a waiting time after the transmission completion. After this waiting time the request window starts with a maximum duration of  $T_{maxBackOff}$ . The calculation of the expected repair costs is done after the waiting time  $t_{offsetTime}$  at a fraction of the repair request window, defined by the parameter  $\alpha$ .

$$t = t_{end} + t_{offsetTime} + \alpha \cdot T_{maxBackOff} \quad (1)$$

For a ptp repair session the estimated costs are defined by formula (2), with  $c_u$  defining the cost of the transmission of a single byte via the used ptp network,  $N_{sym}$  specifying the expected number of requested symbols to repeat,  $s_{sym}$  defining the average size of a symbol in byte,  $N_{req}$  defining the expected number of repair request messages, and  $s_{req}$  specifying the average size of a repair request message in byte. The expected numbers of repair requests and requested symbols are calculated by dividing the numbers of repair requests and requested symbols until time  $t$  by  $\alpha$ .

$$C_{ptp(expected)} = c_u \cdot N_{sym} \cdot s_{sym} + c_u \cdot N_{req} \cdot s_{req} \quad (2)$$

For a ptm repair session, which uses a broadcast channel for sending repair symbols, the number of distinctive requested symbols is the most relevant parameter concerning the resulting repair costs. Concerning the calculation of the ptm repair costs, a modified formula (3) is proposed, which better corresponds to the incurring costs than the formula specified in [5]. The cost of the transmission of a single byte via the used ptm network is given by  $c_m$ , the size of a ptm repair session announcement in byte is defined by  $s_{an}$ , and the number of distinctive requested symbols until  $t$  (see formula (1)) is specified by  $n_{dsym}$ .

$$C_{ptm(expected)} = c_m \cdot s_{an} + c_u \cdot N_{req} \cdot s_{req} + c_m \cdot n_{dsym} \cdot s_{sym} \quad (3)$$

When the time window for file repair requests has expired (see formula (4)), the total cost for the first round of a file repair session can be calculated. The

total cost for a ptp repair session can be calculated according to formula (5). For a ptm repair session formula (6) can be used respectively. The total number of requested symbols and the number of distinctive requested symbols is specified by  $n_{sym}$  and  $n_{dsym}$  respectively. The total number of file repair requests is defined by  $n_{req}$ .

$$t_{final} = t_{end} + t_{offsetTime} + T_{maxBackOff} \quad (4)$$

$$C_{ptp(final)} = c_u \cdot n_{sym} \cdot s_{sym} + c_u \cdot n_{req} \cdot s_{req} \quad (5)$$

$$C_{ptm(final)} = c_m \cdot s_{an} + c_u \cdot N_{req} \cdot s_{req} + c_m \cdot n_{dsym} \cdot s_{sym} \quad (6)$$

### 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 and broadcasting the data via the IPDC/DVB-H infrastructure is as follows [1,5]. First, apply a blocking algorithm to split the binary object into source blocks and encoding symbols. The blocking algorithm used depends on the used forward error correction 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. 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 (TS) provided for this service on the DVB-H 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 the 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).

## 2.4. Error model for the transmission channel

The error model used for the transmission channel is based on the four-state run length model described in [10] and [11]. This model operates on the resulting MPEG-2 TS of the data transmission and produces streams of erroneous TS packets by using four states called *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. The corresponding transition probabilities are based on a set of formulas and constants as depicted in tables 1 and 2. As can be seen from these tables, 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.

**Table 1. Transition probabilities from states in the first column to states in the other columns [10]**

| To<br>From      | Good<br>(short)         | Bad<br>(short)          | Good<br>(long)              | Bad<br>(long)               |
|-----------------|-------------------------|-------------------------|-----------------------------|-----------------------------|
| Good<br>(short) | $\alpha_g$              | $\pi_1=(1-\alpha_g)p_b$ |                             | $\pi_2=(1-\alpha_g)(1-p_b)$ |
| Bad<br>(short)  | $\pi_5=(1-\alpha_b)p_g$ | $\alpha_b$              | $\pi_6=(1-\alpha_b)(1-p_g)$ |                             |
| Good<br>(long)  |                         | $\pi_4=(1-\beta_g)p_b$  | $\beta_g$                   | $\pi_3=(1-\beta_g)(1-p_b)$  |
| Bad<br>(long)   | $\pi_8=(1-\beta_b)p_g$  |                         | $\pi_7=(1-\beta_b)(1-p_g)$  | $\beta_b$                   |

**Table 2. Constants used for the calculation of transition probabilities [10]**

|            |       |
|------------|-------|
| $\alpha_g$ | 0.650 |
| $\beta_g$  | 0.999 |
| $p_g$      | 0.657 |
| $\alpha_b$ | 0.650 |
| $\beta_b$  | 0.982 |
| $p_b$      | 0.840 |

The TS packet error streams produced by this model represent a good approximation to error streams measured using the COST 207 Six-tap Typical Urban (TU6) multi-path channel model [13]. See [10] for a comparison.

## 3. Simulation framework

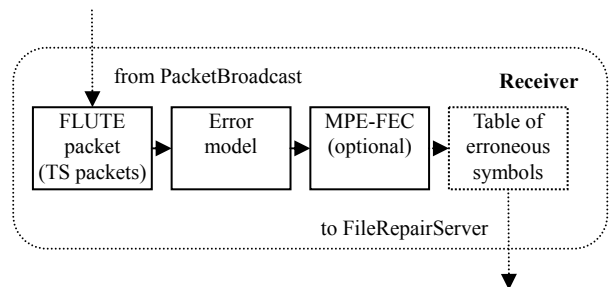
The simulation framework designed and implemented uses the discrete, event-based simulation tool OMNeT++ [9]. With this simulation tool, components and topologies are described using the programming language NED [9]. Simulation components have to be implemented in C++.

### 3.1. Simulation components

The simulation framework consists of the following general components: *Sender*, *PacketBroadcast*, *Receiver*, and *FileRepairServer*. 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 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 according to the formulas described in chapter 2.2.

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 *PacketBroadcast* is very simple and can only be parameterized with the number of receivers.



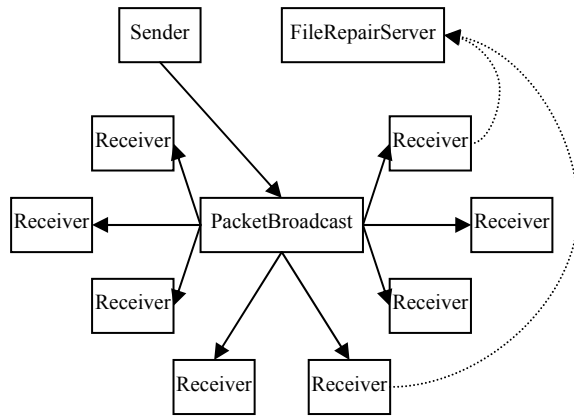
**Figure 3. Details of the component Receiver**

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 is finished.

Figure 3 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 2.4 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 is 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 is finished, fraction of the maximum back-off time for the calculation of estimated repair costs ( $\alpha$ ), file size, average symbol size, average size of a file repair request, size of a ptm repair session announcement, estimated success rate for a receiver using a ptm repair session, and fraction of receivers without a ptp connection.



**Figure 4. Simulation topology**

Figure 4 shows the topology used for conducting the simulations. As an example, this figure shows two receivers sending file repair requests to the file repair server.

### 3.2. Simulation parameters

Several simulation runs have been executed. Table 3 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.

**Table 3. Parameters of the simulation runs**

| <i>File size [MB]</i> | <i>Encoding symbol length [Byte]</i> | <i>MPE-FEC</i>     |
|-----------------------|--------------------------------------|--------------------|
| 4                     | 500                                  | disabled           |
|                       |                                      | enabled, 256 rows  |
|                       |                                      | enabled, 512 rows  |
|                       | 1400                                 | enabled, 1024 rows |
|                       |                                      | disabled           |
|                       |                                      | enabled, 512 rows  |
|                       | 100                                  | enabled, 1024 rows |
|                       |                                      | enabled, 512 rows  |
|                       |                                      | enabled, 512 rows  |

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 repair costs.

**Table 4. Fixed simulation parameters**

| <i>Module</i>    | <i>Name</i>                      | <i>Value</i> |
|------------------|----------------------------------|--------------|
| 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 | $\alpha$                         | 0,2          |

Furthermore, the parameters shown in table 4 were set to fixed values. The values for the DVB transmission mode and TS data rate are typically

technical parameters of current DVB-H trial services [14]. 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, which 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, which 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 [5] and [8]. The number of receivers was the variable simulation parameter for each simulation run.

#### 4. Simulation results

Using the simulation framework depicted in chapter 3, cost comparison studies between ptp and ptm repair sessions have been conducted, depending on the number of receivers. The parameters used for the simulations are defined in chapter 3.2. Some exemplary results of these studies are shown in this chapter.

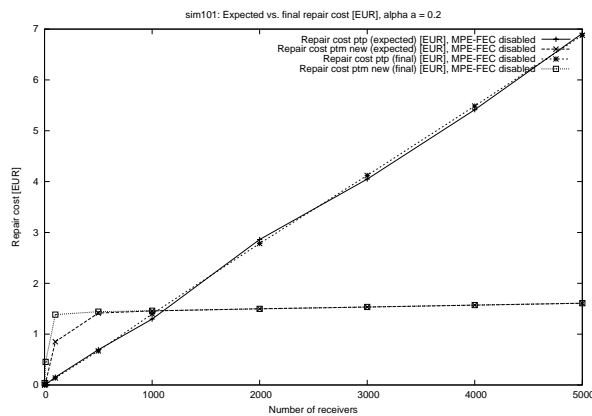


Figure 5. Repair costs of ptp and ptm sessions

Figure 5 compares the expected and final file repair costs for ptp and ptm repair sessions without MPE-FEC. Whereas the costs for ptp sessions rise almost linearly with the number of receivers, the costs for ptm sessions quickly converge to a maximum. This maximum represents the correction of the whole file. These different behaviors are due to the fact that in a ptp repair session each requested symbol has to be transmitted to the requester, whereas in a ptm session only distinctive requested symbols are repeated, independent of the number of different requests for one and the same symbol (cf. formulas (5) and (6) in

chapter 2.2). 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 ( $\alpha = 0.2$ ).

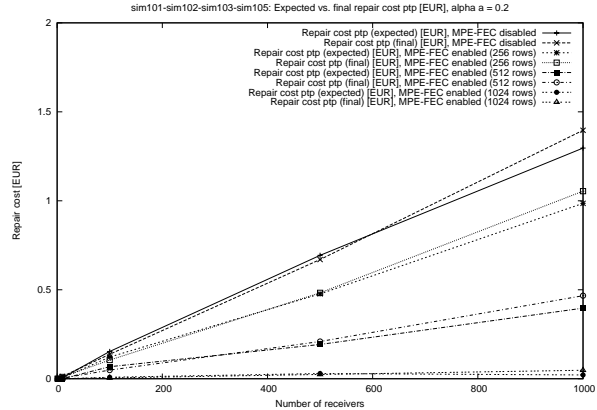


Figure 6. Costs with/without MPE-FEC (ptp)

For the results in figures 6 and 7, additional simulations with MPE-FEC as an additional level of error correction have been executed. The MPE-FEC frame sizes used were 256, 512 and 1024 rows. The results focus on broadcast networks of limited size with a maximum number of 1000 users.

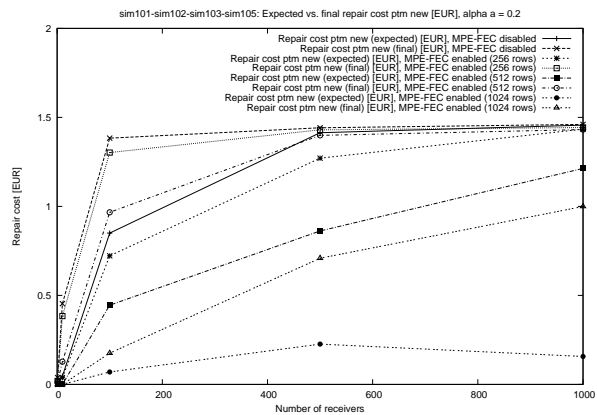
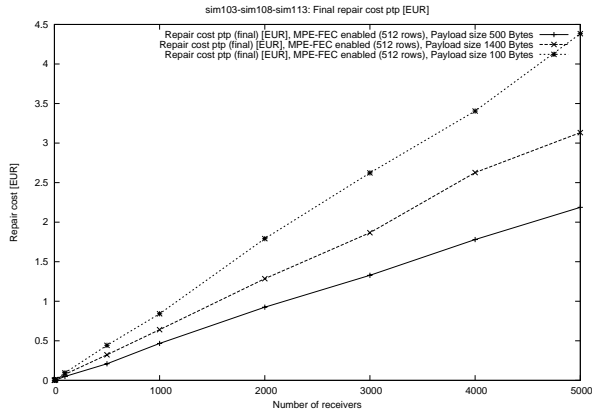


Figure 7. Costs with/without MPE-FEC (ptm)

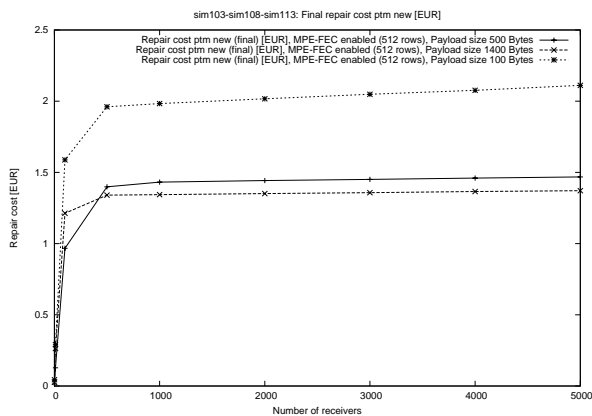
As can be seen clearly in figure 6, the bigger the MPE-FEC frame size, the lower the repair costs for ptp repair sessions. As the encapsulated symbols are inserted into the MPE-FEC frame column by column (see chapter 2.3), a bigger frame consisting of more rows raises the probability that a symbol doesn't span several columns. As the number of corrigible errors per row is limited by the used error correction code, a

symbol buffered in one column only will more likely be corrected in case of errors than a symbol which spans several rows.



**Figure 8. Costs using different symbol sizes (ptp)**

Concerning ptm sessions, the results in figure 7 show that when using bigger MPE-FEC frame sizes, the repair costs converge much more slowly to the maximum, which represents the retransmission of the whole file. This is again due to the more effective error correction capability of bigger MPE-FEC frames. Therefore, the bigger the MPE-FEC frames, the smaller the number of distinctive erroneous symbols. This also results in a more inaccurate estimation of the repair costs, expecting lower cost values than finally occur.



**Figure 9. Costs using different symbol sizes (ptm)**

Figure 8 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 or a very small symbol size of 100 byte. Big symbols tend to span

several columns in an MPE-FEC frame. Therefore, a big symbol is more likely to remain erroneous than a smaller symbol, increasing the repair costs. On the other hand, small symbols cause much more transmission overhead. The FLUTE/UDP/IP encapsulation overhead of 52 byte per symbol results in increased repair costs.

As can be seen in figure 9, very small symbols result in higher ptm repair costs as well. This is again an effect of the disproportion of data to be transmitted and protocol overheads. Therefore, it is recommended to use medium-sized symbols.

## 5. Conclusions

The IPDC/DVB-H file repair mechanism provides protocols and procedures for correcting erroneous (i.e. lost or unrecoverable) data fragments of broadcast transmissions of arbitrary binary data objects. Two modes have been defined: point-to-point (ptp) for retransmitting each erroneous fragment to its requester using a 3G or WLAN data connection and point-to-multipoint (ptm) for retransmitting requested fragments to all receivers using a DVB-H broadcast session. The focus of this study was the examination of resulting repair costs using these different methods with a variety of transmission parameter settings. The results can help providers choose the most cost-efficient strategies and parameter settings. The examination was done by using a simulation framework which incorporates the basic IPDC/DVB-H transmission and file repair mechanisms as well as a current error model of the DVB-H transmission channel. Several simulation runs concerning the transmission of a file to a variable number of receivers have been 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 the file repair costs of the first rounds of ptp and ptm repair sessions. The impact of MPE-FEC on the resulting file repair costs was of special interest. The results of the simulations provided some interesting findings about the setting of several critical transmission parameters. Some of the most interesting conclusions are as follows.

The file repair costs of ptp repair sessions rise (almost) linearly with the number of receivers.

The file repair costs of ptm repair sessions rapidly converge to the costs of retransmitting the whole file. Only a very small number of receivers (typically lower than 100) keep the ptm repair costs low.

The usage of MPE-FEC has a strong impact on the repair costs. The bigger the used MPE-FEC frame, the

lower the rise of file repair costs. Depending on the number of receivers, ptp repair costs can be considerably lower than ptm repair costs if MPE-FEC is enabled.

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 results in low repair costs. Medium-sized symbols (e.g. 500 byte) should be used for lowest repair costs.

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 for ptp sessions, even when the calculation was done within only a small fraction of the whole repair request window. Concerning ptm sessions, the accuracy of the calculation depends on the MPE-FEC frame size used. Bigger frame sizes result in more inaccurate values.

## 6. References

- [1] ETSI, *Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)* (ETSI EN 302 304 V1.1.1), Nov. 2004.
- [2] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital Broadcast Services to Handheld Devices", *Proc. IEEE*, vol. 94(1):194-209, Jan. 2006.
- [3] U. Reimers, *DVB – The Family of International Standards for Digital Video Broadcasting*, Springer, Berlin Heidelberg New York, 2005.
- [4] ETSI, *Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Architecture* (ETSI TR 102 469 V1.1.1), May 2006.
- [5] ETSI, *Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols* (ETSI TS 102 472 V1.2.1). Dec. 2006.
- [6] M. Luby et al., *Asynchron Layered Coding (ALC) Protocol Instantiation* (IETF RFC 3450), The Internet Society, 2002.
- [7] T. Paila et al., *FLUTE - File Delivery over Unidirectional Transport* (IETF RFC 3926), The Internet Society, 2004.
- [8] DVB Project, *IPDC over DVB-H: Content Delivery Protocols (CDP) Implementation Guidelines* (DVB BlueBook A113), Mar. 2007.
- [9] A. Varga, *OMNeT++ Community Site*, <http://www.omnetpp.org/>
- [10] J. Poikonen and J. Paavola, "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. 4:1861-1866, Jun. 2006.
- [11] J. Poikonen and J. Paavola, *Comparison of Finite-State Models for Simulating the DVB-H Link Layer Performance*, Technical Report, University of Turku, 2005.
- [12] R. Fielding et al., *Hypertext Transfer Protocol -- HTTP/1.1* (IETF RFC 2616), The Internet Society, 1999.
- [13] COST 207, *Digital land mobile radio communications*, Final Report, Commission of the European Communities, 1989.
- [14] DVB Project Office, *DVB-H: Global Mobile TV*, <http://dvb-h.org/>