LDPC benchmarking for DVB-H

Subject	Description and discussion of the benchmarking of our LDPC soluAL-FEC for DVB-H
Category	Report
Revision	1.0.1
Authors	STM/AST and INRIA

Table of contents

Document history

Revision	Author	
1.0	STM/AST- and INRIA	

1. Introduction

This document introduces and discusses the LDPC-Staircase AL-FEC benchmarking results.

The LDPC-Staircase benchmarking has been carried out ***following the guidelines defined in TM-CBMS1361***, and relies on the associated set of traces (from T-Systems International GmbH).

The LDPC-Staircase AL-FEC codes considered have been defined in [4] and [5].

The performance measurements reported in this document take advantage of the ***public LDPC codec*** available at URL: http://www.inrialpes.fr/planete/people/roca/mcl/ under a GNU/LGPL license. Please note that some mechanisms used during this work are not necessarily reflected in the public codec. Note also that the public codec is a generic codec, meant to be useable in many diverse environments, rather than being a codec specialized for one particular use case.

Parts of the results have been carried out on an ***STM embedded platform, Nomadik® Mobile Multimedia Processor***. These results are not, in this version of the document, finalized and should be considered appropriately. In particular the memory management aspects of the FEC codec has not been specialized for this embedded platform and will be reported in a future version of this document.

2. Test environment

2.1. Target devices

The tests were performed on two different systems, having significantly different constraints:

- **Workstation**: this terminal provides a large amount of resources, in particular CPU, memory and power consumption.
- **Embedded environment**: the Nomadik® Mobile Multimedia Processor platform is constrained in terms of CPU, memory and power consumption. More precisely, this is a mobile phone development platform based on ARM926EJ-S host-CPU, a powerful 32-bit RISC core that can typically run at 350 MHz in ST's 0.13-micron CMOS process. The core includes a memory management unit (MMU), 32 Kbytes of instruction cache, 16 Kbytes of data cache, a 16 x 32-bit multiplier capable of single-cycle MAC operations, and strong real-time debug support. It is therefore representative of the target DVB-H enabled devices.

2.2. Test descriptions

2.2.1. Methodology

The tests carried out strictly follow the TM-CMBS 1361 document of June 2005. They rely on the associated trace files, from T-Systems International GmbH, in the “ALG Second Trial.zip” archive.

IP datagrams use a maximum of 512 byte payload, resulting in a maximum total IP datagram size, including the IP/UDP/FLUTE headers, of 556 bytes. The effects of MPE-FEC erasure recovery are simulated.

Case A: Single file download

The users in case A are simulated in such a way that they don’t share the same portions of the error trace. More precisely, receiver *i* receives data until it is able to decode (or until the trace stops). Then the end of the current MPE Frame is skipped, and the following receiver, *i+1*, starts receiving at the next MPE Frame. Users are therefore synchronized with the MPE-FEC frames, as requested in TM-CMBS 1361, but experience different losses from one another.

In case A, transmissions are repeated (simulating a static carousel with a single file). Source and parity symbols are sent in a random order in each repetition.

Case B: Static file carousel

In case B, a user waits a random time before starting to listen to the channel. Each carousel cycle begins at a new time slice.

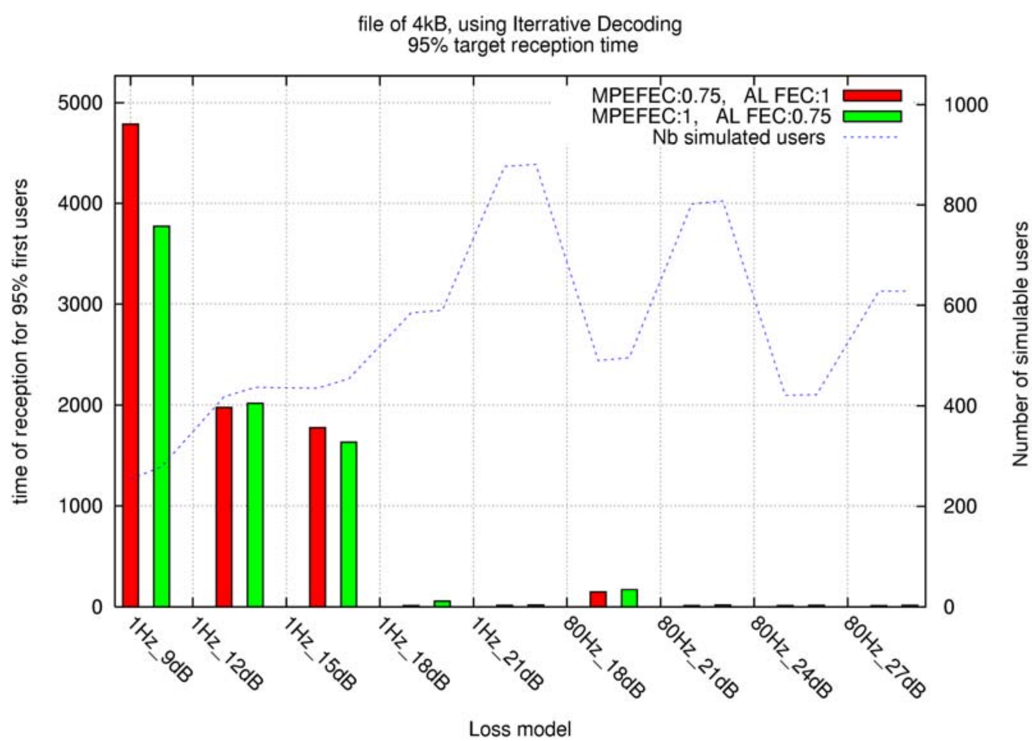
2.2.2. Limitations

Due to the insufficient length of the TS packets error trace files, **it was not possible to use the time slicing feature** of DVB-H in case A and B. Moreover, the simulations for big files must be taken with care. Indeed it is not always possible to provide significant results (i.e. simulate a sufficient number of users to provide accurate 95% reception times) because **some trace files turned out to be too short**. Finally the trace file for 15dB at 80Hz was not available.

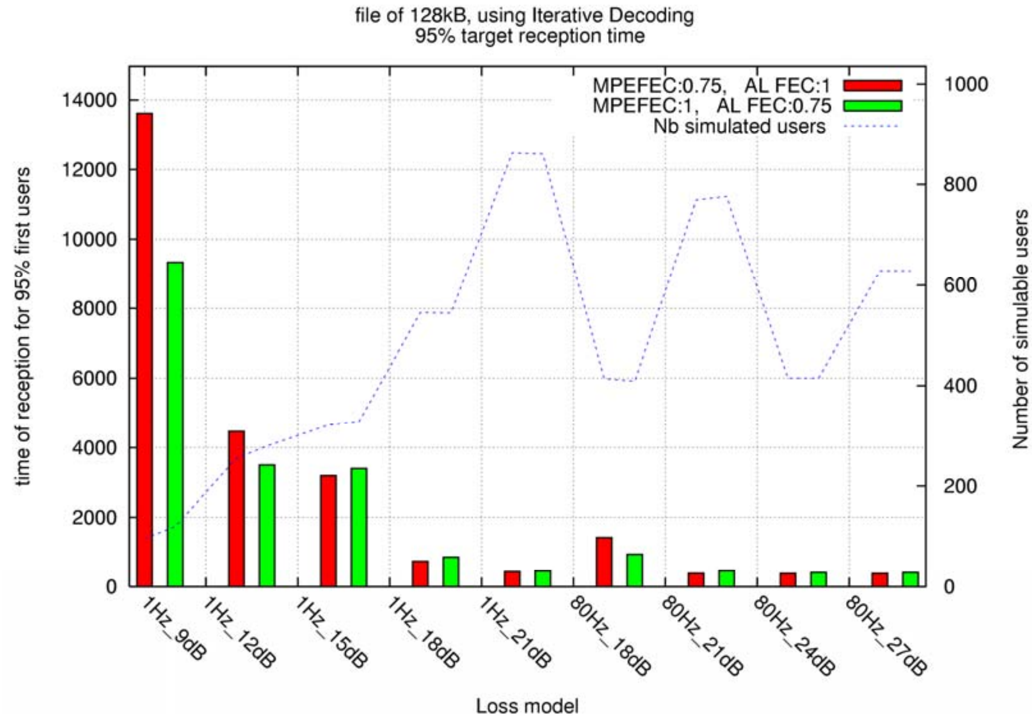
3. Results

3.1. Case A: Single file download

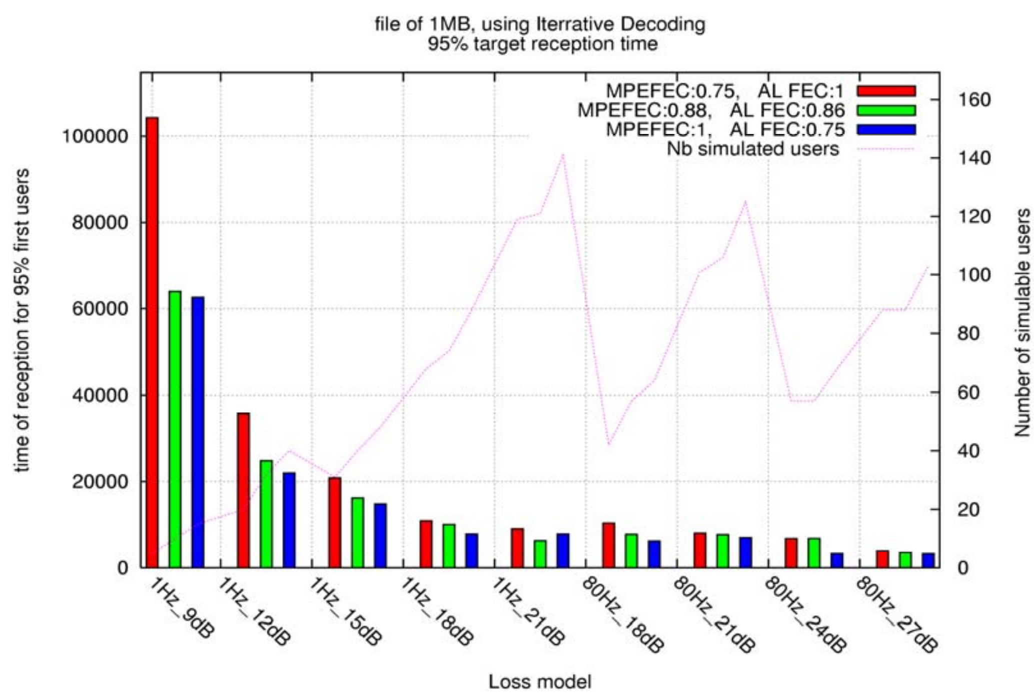
Results for 4k file size

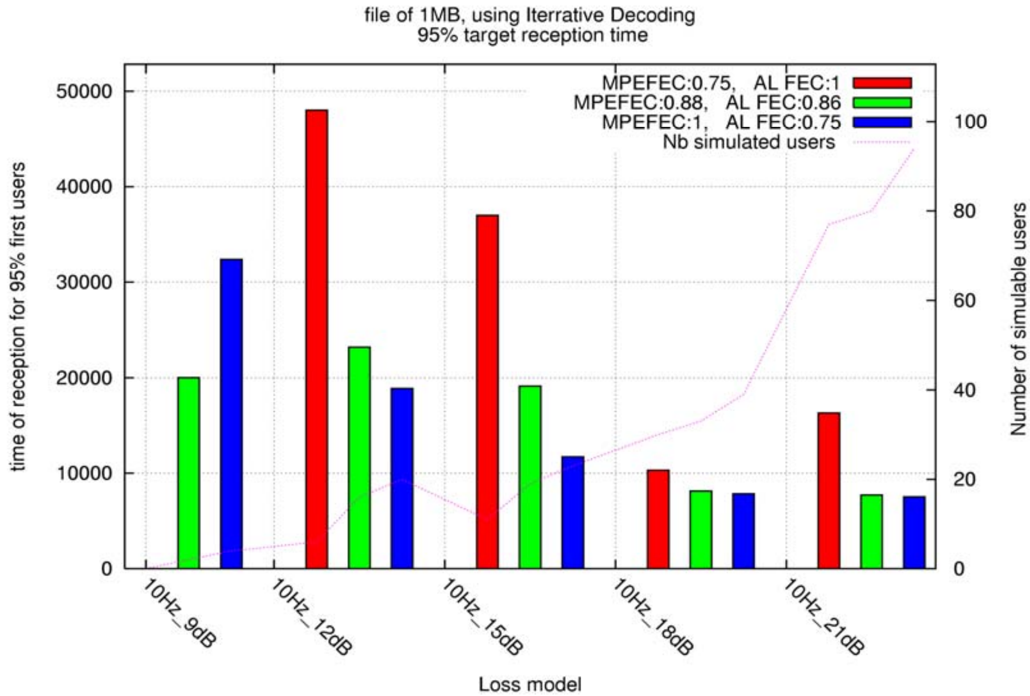


Results For 128k file size



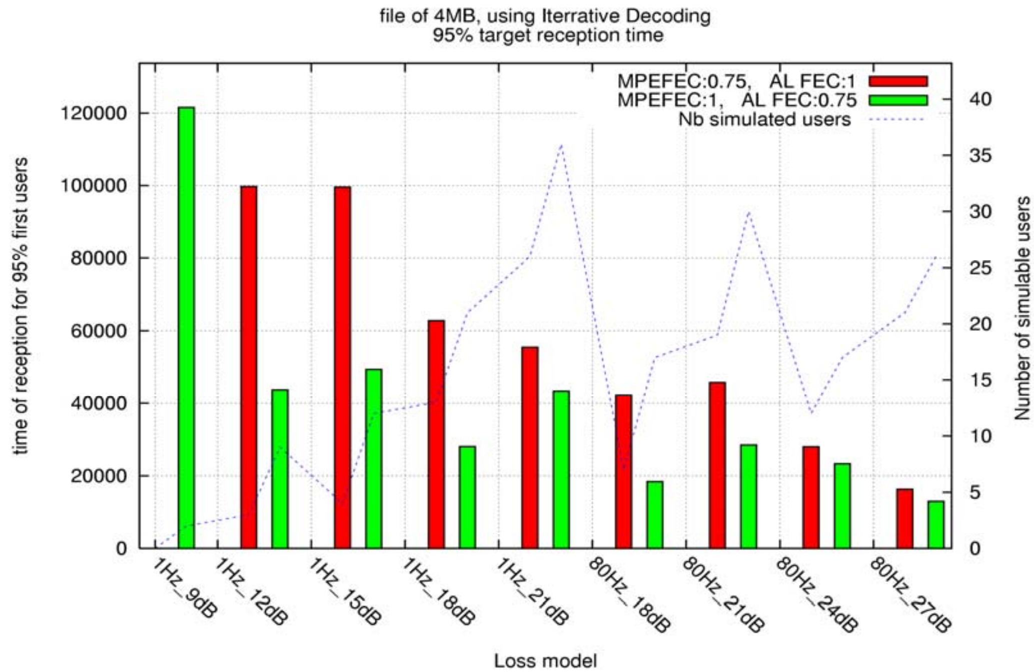
Results for 1MB file size





Please note that the missing bar in the figure above means that with MPE-FEC the trace file wasn't long enough to simulate one receiver as the line "Nb of simulated users" shows.

Results for 4MB file size



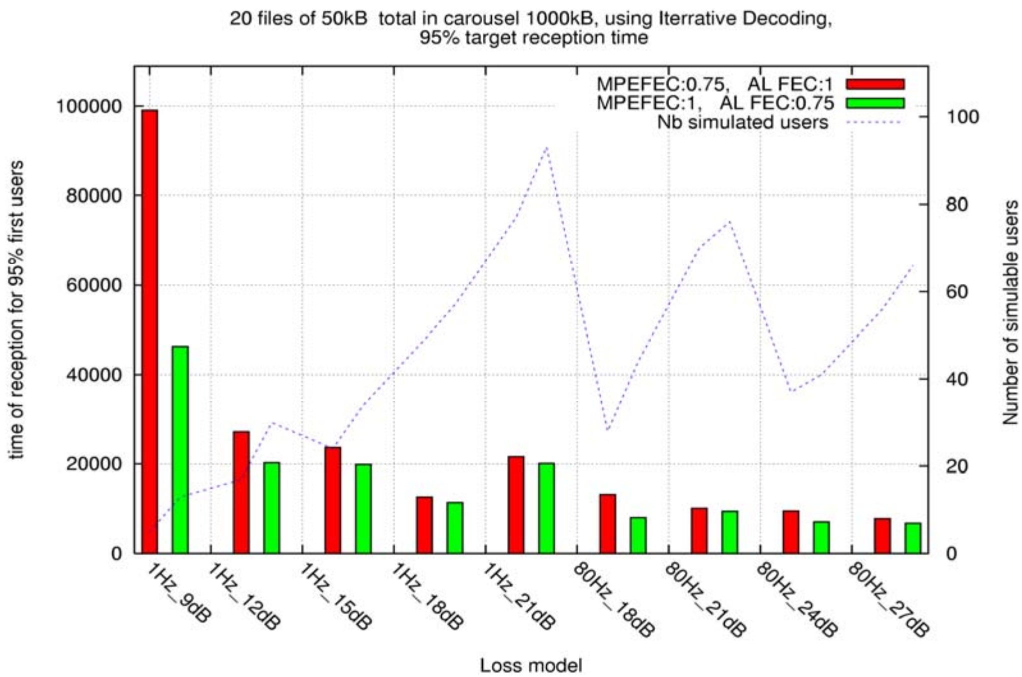
Please note that the missing bar in the figure above means that with MPE-FEC the trace file wasn't long enough to simulate one receiver as the "Nb of simulated users" curve shows.

3.2. Case B: Static File Carousel

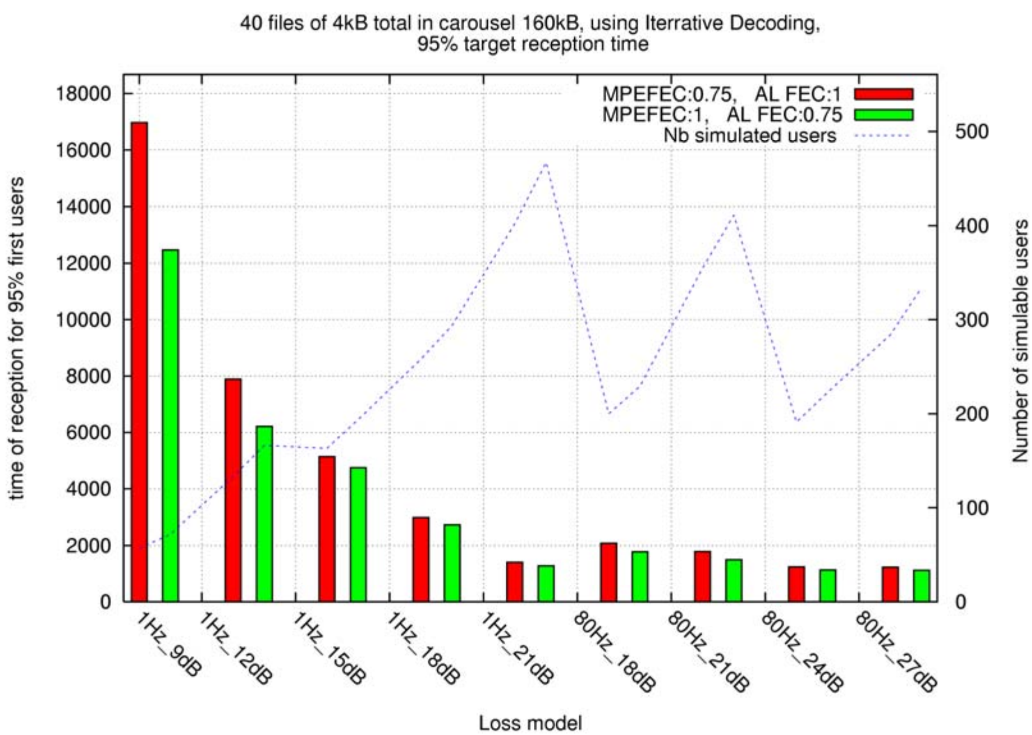
As requested in TM-CMBS 1361, the simulations use the 1Hz and 80Hz channel models.

	Case B.1	**Case B.2**	**Case B.3**
Number of files	20	40	20
File size:	50KB	4KB	400 bytes
Target probability of reception	95%	95%	95%
MPE-FEC rates	3/4, 7/8, 1	3/4, 7/8, 1	3/4, 7/8, 1

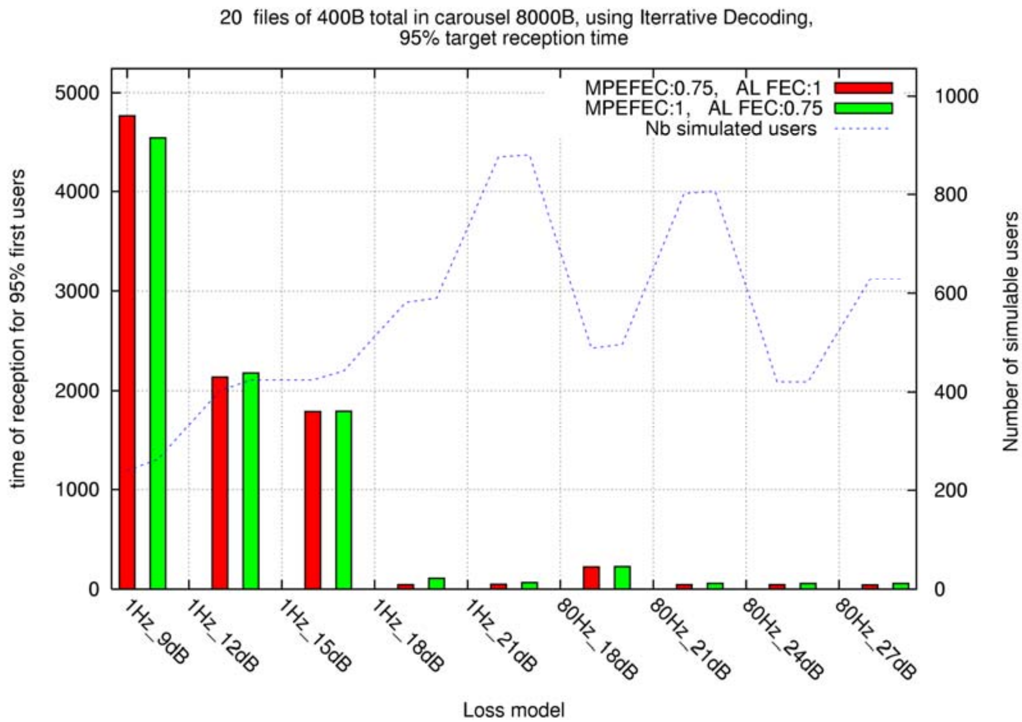
B 1


20 files of 50kB total in carousel 1000kB, using Iterrative Decoding,
95% target reception time
MPEFEC:0.75, AL FEC:1
MPEFEC:1, AL FEC:0.75
Nb simulated users
time of reception for 95% first users
Number of simulable users
Loss model
1Hz_9dB
1Hz_12dB
1Hz_15dB
1Hz_18dB
1Hz_21dB
80Hz_18dB
80Hz_21dB
80Hz_24dB
80Hz_27dB

B 2


40 files of 4kB total in carousel 160kB, using Iterrative Decoding,
95% target reception time
MPEFEC:0.75, AL FEC:1
MPEFEC:1, AL FEC:0.75
Nb simulated users
time of reception for 95% first users
Number of simulable users
Loss model
1Hz_9dB
1Hz_12dB
1Hz_15dB
1Hz_18dB
1Hz_21dB
80Hz_18dB
80Hz_21dB
80Hz_24dB
80Hz_27dB

B 3



3.3. Embedded Environment Results

On embedded platform only speed results are presented, decoding results are identical as the previously presented as the algorithms are similar on embedded platform.
Those results are preliminary. They show the average values from 10 simulations for each file size. Memory optimizations are still under progress and will be considered in the next version of this document.
Values for encoding and decoding speeds in the table are computed in the following way:

$$EncodingSpeed = \frac{(DataSymbols + FecSymbols) * SymbolSize * 8}{EncodingTime}$$

$$DecodingSpeed = \frac{(DataSymbools + FecSymbols) * SymbolsSize * 8}{DecodingTime}.$$

	file size	AL-FEC	iterative decoding algorithm					
	[byte]		inef_ratio	encoding time [sec]	encoding speed [bit/sec]	decoding time [sec]	decoding speed [bit/sec]
Staircase	4k	3/4	1,1375	0,001041	42.068.038	0,007111	4.607.813
	128k	3/4	1,0526	0,020342	68.739.191	0,094116	11.141.315
	1M	7/8	1,0294	0,092061	106.336.220	0,215071	39.003.937
	1M	3/4	1,0513	0,097972	114.177.508	0,243369	34.468.736
	4M	3/4	1,0521	0,400997	111.573.479	0,995955	33.690.704

4. Discussion about results

4.1. General conclusions

The following general conclusions can be made:

- The LDPC-Staircase AL-FEC is most of the time at least as efficient, in terms of erasure recovery capabilities, as the MPE-FEC scheme. The only exception is in case of 4kB objects and a very small erasure rates, or in case of static file carousel, case B3 where there is a large number of very small objects.
- In many situations, in particular when the object size increases, the LDPC-Staircase AL-FEC erasure capabilities are significantly higher than that of the MPE-FEC scheme. For instance with 1MB files, using LDPC only makes the 95% target reception probability time between 12% to 51% shorter than with MPE-FEC only.
- In most situations, using both MPE-FEC and LDPC-Staircase AL-FEC (for the same amount of MPE-FEC/AL-FEC parity overhead) is more interesting than using MPE-FEC only, but is sub-optimal compared to using AL-FEC only.
- This document therefore recommends to avoid the use of MPE-FEC (by setting a code rate of 1.0) and to rely only on LDPC-Staircase AL-FEC.
- For the reasons explained in section 4.3, no comparison can be made with other AL-FEC schemes currently.

4.2. Open issues

Several open issues exist. They will be corrected in the following version of this document. More precisely:

- Memory requirements in constrained embedded environments are not included in the present document which relies on the generic LDPC Staircase codec.
- The present work only considers an iterative decoding algorithm. Other decoding algorithms derived from Gauss triangulation can also be used in the context of DVB-H. This is left to future works.

4.3. Requests related to document TM-CBMS1361

This work has highlighted several shortcomings in the TM-CBMS1361 document:

- This document does not specify any IP datagram size. We chose 512 byte IP datagram payloads in our tests, yet other choices are possible and will significantly impact the final results. Generally speaking, using smaller IP datagram sizes will reduce the impacts of MPEG2-TS packet losses at the cost of a higher protocol header overhead, and vice-versa. **Recommendation:** specify an IP datagram payload size and IP/UDP/FLUTE header sizes for comparison purposes.
- This document should specify that the performance comparisons between several AL-FEC codes require that the performance results in case there is no AL-FEC (AL-FEC code rate = 1) match. **Recommendations:** mandate that before any performance comparison, the benchmarking environment accuracy be checked/calibrated with known results for the case where no AL-FEC is used.

- This document should define a minimum number of client reception statistics in order to have a valid 95% reception measurement. Indeed, with some loss traces, the results do not necessarily include a sufficiently high number of reception statistics, which makes the results inaccurate. **Recommendations:** specify a minimum number of client reception statistics for calculating the 95% reception threshold.

This work has highlighted several shortcomings in the associated set of traces (from T-Systems International GmbH):

- Traces are too small in some situations (in particular with large files). The 95% successful reception statistics can therefore be inaccurate. **Recommendations:** provide larger traces, or clearly identify situations where there could be problems.

All these topics should be addressed before comparison between several AL-FEC codes can be made.

5. Conclusion

We have shown that in general, using the LDPC AL-FEC scheme only is the most efficient solution. Unfortunately, at the present time, no comparison with other AL-FEC schemes is feasible, and several issues identified in section 4 must be fixed first. Besides, the benchmarking work presented in this document is the only one we are aware of, that complies with the guidelines defined in TM-CBMS1361, June 2005.

The present document is a first version that will be updated when newer results become available, in particular for embedded applications as notified in chapter 4.

6. Glossary

AL-FEC	Application Layer Forward Error Correction
LDPC	Low Density Parity Check
MPE-FEC	Multi Packet Encapsulation Forward Error Correction

7. References

[1] TM-CBMS1361 (Revision of 1323) 21 June 2005 “Proposal for simulations for evaluation of Application Layer FEC for file delivery”

[2] ETSI EN 301 192 V1.4.1 (2004-11) Digital Video Broadcasting (DVB); DVB specification for data broadcasting

[3] Trace files from T-Systems International GmbH

[4] V. Roca, C. Neumann, D. Furodet, “Low Density Parity Check (LDPC) Forward Error Correction”, IETF RMT Working Group, draft-ietf-rmt-fec-bb-ldpc-00.txt (Work in Progress), October 2005.

[5] STMicroelectronics, “LDPC Staircase FEC codes”, submitted to the DVB-H group, September 2005.