Receiver Coding Gain in DVB-H Terminals Using Application Layer FEC Codes

Sébastien Lafond, Kristian Nybom, Jerker Björkqvist, Johan Lilius
Åbo Akademi University
Department of Information Technologies
sebastien.lafond@abo.fi, kristian.nybom@abo.fi, jerker.bjorkqvist@abo.fi, johan.lilius@abo.fi

Abstract

DVB-H is targeted for broadcasting digital content to handheld devices. The content is generally divided into streaming media or file downloading. In file downloading scenarios there is typically a requirement on a zero error ratio which can be met using either a data carousel or additional forward error correction codes. Both methods however induce a higher energy consumption in the receiver. This paper analyses the energy consumption needed for two different forward error correction codes based on emulator results for typical hardware in a handheld device. The energy used for error correction is compared to the energy used when receiving more carousel rounds in order to meet the zero error ratio requirement. This difference is denoted as receiver coding gain. Additionally, error correction also leads to a reduction in the reception time.

1. Introduction

DVB-H is a relatively new standard in the set of standards developed by the DVB Project. DVB-H is mainly targeted for handheld devices, but is also intended for mobile usage in for instance cars or buses. The main use case for DVB-H is watching television broadcasts, but file downloading enables receiving digital content (MP3's, videos, etc.) for storing and later use.

DVB-H is based on the physical layer of DVB-T. DVB-T is a Coded Orthogonal Frequency Division Multiplexing (COFDM) system, where the basic data item is Transport Stream (TS) packets of size 188 bytes. In broadcast systems, such as DVB-T, errors turn up in the stream, even while using good physical layer Forward Error Correction (FEC) codes. DVB-T was not designed for mobile usage, and therefore DVB-H includes an optional FEC code at the link layer, embedded in the Multi Protocol Encapsulator (MPE or MPE-FEC) to compensate for the performance degradations due to fast fading effects in mobile channels. These additions make the delivery of standard IP packets over the DVB-T network possible, which increases the performance in mobile usage environments. Furthermore, the MPE-FEC layer adds time interleaving to the system, making it more resistant to slow fading effects (e.g. temporal obstacles). While the MPE-FEC provides an adequate performance improvement for video streaming services, where errors lead to frame losses or pixelation, it does not add a sufficient performance improvement for file downloading scenarios.

In file downloading services, an additional layer of error correction at the application layer (AL-FEC) is used, in order to deal with lost IP packets. The DVB-H standard [3] specifies a Raptor code [7] for file delivery scenarios. Although Raptor codes have several attractive properties, they may not be the best choice for application layer coding. One of the most notable reasons, which will be shown in this paper, is that their energy consumption may be too large in mobile devices for providing cost effective downloading, compared to that of other codes. In this paper we present simulation results which indicate that Hyper Low-Density Parity-Check (HLDPC) codes provide similar error correction performance as the Raptor code, but at a lower energy consumption in the host processor.

The main contributions of this paper are: (a) a comparative study between two AL-FEC codes (the HLDPC and Raptor code), (b) an analysis of the receiver coding gain in terms of energy consumption when application layer coding is used, and (c) a theoretical analysis for receiver coding gain.

2. Receiver coding gain

Coding gain for a transmitter is traditionally defined for a specified error probability as the reduction in required energy per transmitted information bit $E_{b,c}^T$ when using error correction coding compared to the energy per information bit $E_{b,0}^T$ when error correction coding is not used. The coding gain from the transmitter point of view is given by

$$G^T = \frac{E_{b,0}^T}{E_{b,c}^T} \quad (1)$$

and is often given in the logarithmic scale dB as

$$G_{dB}^T = 10 \log_{10} \frac{E_{b,0}^T}{E_{b,c}^T} \quad (2)$$

In satellite communications the coding gain is essential, where the energy budget in transmitting satellites is limited. In terrestrial systems, the availability of energy at transmitter stations is not that big an issue, rather a network planning or a regulatory matter limiting the power available at the transmitter. In handheld devices, on the other hand, the total energy budget is limited. Using the same logic as defining the transmitter side coding gain, we can define the receiver coding gain. Receiver coding gain is defined as the reduction of energy per information bit $E_{b,c}$ needed while using coding compared to the energy per information bit $E_{b,0}$ needed without coding, where the same error rate is achieved:

$$G_{dB} = 10 \log_{10} \frac{E_{b,0}}{E_{b,c}} \quad (3)$$

The use of AL-FEC in downloading should provide receiver coding gain. Furthermore, additional error correction coding should provide other benefits, which typically could be reduced downloading time for the required objects. The rest of this paper describes the codes used for additional coding, the emulations performed, and theoretical analysis for obtaining figures on typically achievable advantages of using coding.

3. Application Layer Codes Used for Simulations

In this paper, we compare the energy consumption of the Raptor code [7], the HLDPC code [5], and a system without AL-FEC. In [5], the erasure correction and overhead performances of these codes have been compared. We therefore limit ourselves to investigating the energy consumed in the receiving device by using the AL-FEC codes. The Raptor code is standardized for IP-datacasting (IPDC) services in DVB-H, but due to its license fees, other codes that can achieve similar performances in IPDC services are of great interest.

Raptor codes are rateless codes that belong to the class of concatenated LDPC/rateless LDGM codes, which achieve a good performance in terms of erasure correction and reception overhead performances. However, the decoding algorithm given in the DVB-H standard [3] for the Raptor code has a high computational complexity. As will be shown in this paper the HLDPC code shows a significantly better performance than the Raptor code, even when using less complex decoding algorithms for both codes. The HLDPC code is a fixed-rate code, which has a similar erasure correcting performance as the Raptor code [5], but as will be shown in section 6, the energy consumption of the HLDPC code is significantly lower than that of the Raptor code.

The Raptor decoding algorithm relies on Gaussian elimination of the parity-check matrix and is capable of yielding reception overhead performances in the order of 1–2 %. Because of the algorithm's high complexity we therefore compare the Raptor code with the HLDPC code, using a computationally simpler algorithm for both codes, namely the greedy iterative Belief Propagation algorithm. Using this algorithm, the erasure correcion and reception overhead performance of the Raptor code is degraded at the gain of decreasing the computational complexity in the decoder. The greedy iterative Belief Propagation algorithm works as follows:

Algorithm 3.1 *Given the value of a parity symbol and all but one of the information symbols on which it depends, set the missing information symbol to be the XOR of the parity symbol and its known information symbols.*

Clearly, this algorithm only works on erasure channels where the decoder knows which symbols are correct and which are not. Since the IP layer in a network protocol can be viewed as a packet erasure channel, where IP packets are either received without errors or corrupted and therefore discarded, the greedy iterative Belief Propagation algorithm is usable at the application layer.

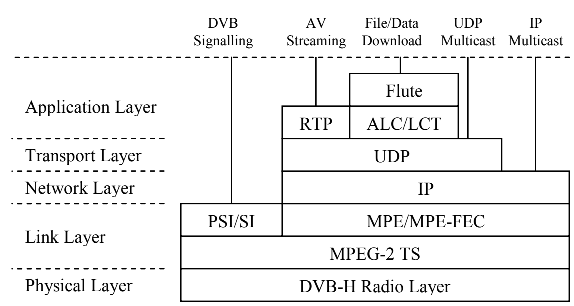

Figure 1. A overview of the system layers in a DVB-H receiver.

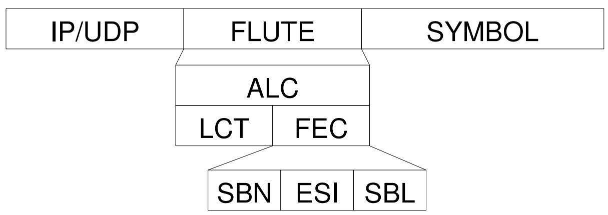

Figure 2. Packet structure for the encoded data

4. Protocols used in file downloading

The system layers in a DVB-H receiver are illustrated in figure 1, [2]. For the scope of this paper, only the application layer is of interest. In DVB-H IPDC services, the FLUTE protocol [4] is used for delivering objects to the receiving terminals. The FLUTE protocol is built on top of the Asynchronous Layered Coding (ALC) protocol, which combines the Layered Coding Transport (LCT) building block, a congestion control building block, and the FEC building block. However, the congestion control building block is not used in DVB-H IPDC services. The ALC and LCT building blocks contain relevant information for the file delivery, while the FEC building block is used by the FEC decoder. The FEC building block is comprised of three fields: the Source Block Number (SBN), the Encoding Symbol ID (ESI), and the Source Block Length (SBL). This gives the IP packet structure that is shown in figure 2.

The information obtained from the FEC building block is used in the following manner. The SBN signifies to which FEC block the received symbol belongs. The ESI is the encoding symbol index of the received symbol with the rule that if ESI $\geq$ SBL the received symbol is a parity symbol, otherwise it is an information symbol. Additionally, the Raptor code uses the ESI value as a seed to its random number generator, to create the degree and edge distributions of the symbol (see [3] for details). The SBL is the number of information symbols in the FEC block to which the received symbol belongs to.

5. Measurement Framework

A measurement framework was created in order to evaluate the extra costs created by the HLDPC and Raptor decoders. Figure 3 presents the full measurement framework used for evaluating the HLDPC and Raptor codes. The file containing the data to be transmitted over the DVB-H network was first encoded by the HLDPC or the Raptor encoder. For the measurements the IP/UDP and LCT headers were not included in the packets because the decoders only require the FEC headers for reconstructing the received object. The obtained encoded file was thus composed of packets and each packet contained one FEC header and one symbol.

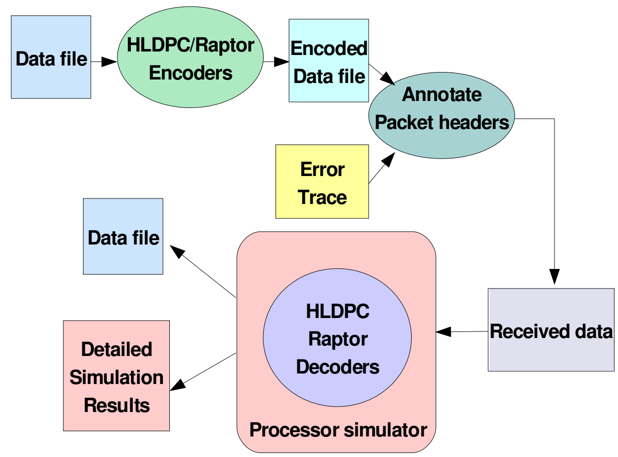

Figure 3. Measurement system - General view

The error trace was a binary map specifying whether each transmitted packet was correctly received or an erasure. In this work a Binary Erasure Channel (BEC) was used, i.e. erasures were distributed uniformly at random. Using the error trace each packet was tagged with erasure information by setting a flag in the packet header. Figure 4 shows the packet structure for the received data containing the error flag. The received data was then read by a software implementation of the HLDPC or Raptor decoder which tried to reconstruct the received object.

The Sim-Panalyzer [8] processor simulator was used for evaluating the costs generated by the execution of the decoders. Sim-Panalyzer is based on the SimpleScalar [1] processor simulator and performs cycle accurate simulations of a strongARM SA-110 processor. It computes at every simulated cycle the energy consumption of each module constituting the ARM core (clock, alu, cache, etc.). RTEMS was chosen as the operating system for this study because RTEMS 4.6.2 is to the best of our knowledge the only OS ported onto SimpleScalar (ported by Jack Whitham [9]).

Table 1. Source and code lengths in number of symbols

Code length	Source length	Code rate
4000	3000	0.75

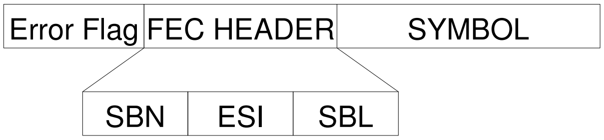

Figure 4. Packet structure for the received data

5.1. Simulation parameters

Tables 1 and 2 present the parameters used by the HLDPC and Raptor codecs. For all measurements, the number of information symbols were set to 3000 symbols and the codeword lengths were set to 4000, hence giving code rate $R = 3/4$. The Raptor code was a non-systematic code, i.e. 4000 rateless symbols were transmitted. The FEC building block used 12 bytes for the SBN, ESI, and SBL header fields and the symbol sizes were set to 1432 bytes, thereby giving IP packet payloads of 1444 bytes.

The processor parameters and the configuration of the caches must be defined in Sim-Panalyzer. For this study the processor speed was set to 233 MHz. The configuration for the level 1 instruction cache, level 1 data cache and the unified level 2 cache is presented in table 3. Table 4 shows the different latencies for each memory level. This configuration targets the average performance of the host processor in a multimedia handheld device. All other parameters used by Sim-Panalyzer were set to their default values.

6. Results

The measurement framework for the HLDPC and Raptor codes was run on the BEC with probabilities of erasures ranging from 0% to 20%. All data was transmitted in a carousel-like manner. The Raptor code was able to decode within one carousel round the received data containing up to 16% of erasures, while the HLDPC code was able to decode data containing up to 14% of erasures.

Figure 5 presents for each code the required time in clock cycles for decoding the received data depending on the erasure rate. We observe that increasing the erasure rate does not affect the execution time of the Raptor decoder while the execution time for the HLDPC decoder slightly increases

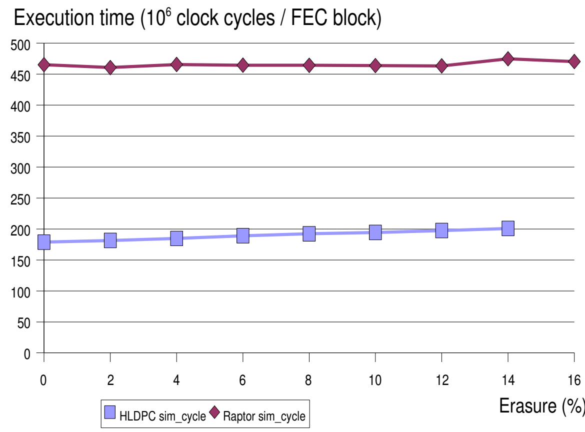

Figure 5. Clock cycles needed for complete reconstruction of received data

with the erasure rate. This is due to the Raptor code being unaffected by the erasure rate in the channel in terms of reception overhead performance, i.e. the Raptor code has an approximately constant reception overhead. In other words, almost every symbol that the Raptor code receives can be used for decoding, while the HLDPC code may receive symbols that have already been received or reconstructed and therefore are useless for the decoding procedure. For the HLDPC code, this fact reflects itself as increased execution time. The results indicate that the raptor decoder requires about $465 \cdot 10^6$ processor cycles regardless of the erasure rate. On the other hand, depending of the erasure rate the HLDPC code requires about $175 \cdot 10^6$ to $200 \cdot 10^6$ processor cycles in order to reconstruct the object.

6.1. Energy Budget Analysis

In this section we evaluate the cost of using AL-FEC in terms of energy consumption. For all the results presented in this section, we assume a data transmission rate T_b of 5 Mbits/s.

Based on Monte Carlo simulations (a similar approach as in [6]), table 5 presents for a receiver not using AL-FEC the minimum, maximum and average number of required carousel rounds on the BEC for downloading the uncoded object without errors. The given values are obtained based

Table 2. Symbol, Packet and Data file sizes in bytes

Symbol	Packet	Data file	Encoded data file
1432	1444	4 296 000	5 776 000

Table 3. Caches configuration

Caches	Associativity	Size	# blocks
il1	direct mapped	4 Kb	128
dl1	direct mapped	4 Kb	128
ul2	4-way	8 Kb	256

on 1000 experiments.

Figure 6 and Figure 7 present the energy consumption comparison for decoding the received data between a receiver without AL-FEC and a receiver using the HLDPC and the Raptor codes respectively for several average power dissipations while the data is received. The average power dissipated while receiving the data includes the power dissipated by the radio receiver and by other system units, like the screen, processor, possible speaker, etc. Figures 6 and 7 show that the energy consumption for a receiver without AL-FEC is increasing with the erasure rate, proportionally to the average power dissipated while receiving data, but at a faster pace than for a receiver using AL-FEC.

The energy consumption for a receiver without AL-FEC, illustrated in figures 6 and 7 with dashed lines, are calculated with the following equation as

$$E_{tot,0} = \epsilon_0 t_0 \bar{P} \quad (4)$$

where ϵ_0 denotes the transmission overhead for an uncoded transmission, t_0 is the time consumed for transmitting all information symbols during one carousel round, and $\bar{P}$ is the average power consumed by the receiver. If n' is the number of transmitted symbols at the time when the receiver has obtained the entire object and k is the number of information symbols in the object, then the transmission overhead is defined as $\epsilon = \frac{n'}{k}$, hence $\epsilon \geq 1$. The energy consumption for a receiver using AL-FEC is calculated in a similar manner as

$$E_{tot,c} = \epsilon_c t_0 \bar{P} + E_c \quad (5)$$

where $E_{tot,c}$ is the total energy used for receiving an object including decoding, ϵ_c is the transmission overhead for the encoded transmission, and E_c is the total energy used by the decoder on host processor, in our case the simulated strongARM SA-110 processor. Table 6 gives as example the energy consumed by the processor functional units for decoding an object with 6% of erasures. The values of $E_{tot,c}$ are illustrated in figures 6 and 7 with continuous lines.

When dividing the total energy required by the number of source bits L_0 we get the energy per bit as

$$E_{b,c} = \frac{E_{tot,c}}{L_0} \quad (6)$$

Table 4. Memory Latencies

	il1	dl1	ul2	RAM *first chunk access*	RAM *inter chunk access*
Latency in cycles	2	2	6	30	4

Table 5. Number of required carousel rounds on the BEC when no AL-FEC is used

Erasure rate (%)	Avg	Min	Max
0	1	1	1
2	2.71	2	5
4	3.17	2	5
6	3.54	3	6
8	3.91	3	8
10	4.26	3	7
12	4.55	3	9
14	4.86	4	8
16	5.16	4	9

$$E_{b,0} = \frac{E_{tot,0}}{L_0} \quad (7)$$

Substituting the above expressions into equation 3 gives

$$G_{dB} = 10 \log_{10} \frac{\epsilon_0 t_0 \bar{P} L_0^{-1}}{\epsilon_c t_o \bar{P} L_0^{-1} + E_c L_0^{-1}} \quad (8)$$

Simplifying equation 8 and taking into consideration that $t_0 = \frac{L_0}{T_b}$, where T_b is the transmission rate in the network, the final expression for the receiver coding gain is obtained as

$$G_{dB} = 10 \log_{10} \frac{\epsilon_0}{\epsilon_c + \frac{TbE_c}{L_0 \bar{P}}} \quad (9)$$

Note that by using probability theory, the expected value on the transmission overhead for the uncoded transmission $E\{\epsilon_0\}$ (abusing notation) over a BEC can be calculated as

$$E\{\epsilon_0\} = \sum_{\epsilon_0=1}^{\infty} \epsilon_0 \left[\left(1 - p_e^{\epsilon_0}\right)^k - \left(1 - p_e^{\epsilon_0 - 1}\right)^k \right] \quad (10)$$

Table 6. Energy consumption in Joule for the processor functional units for an erasure rate of 6%

	HLDPC	Raptor
instruction cache level 1	0.362	0.860
data cache level 1	0.204	0.483
unified cache level 2	1.18	3.41
clock	0.207	0.509
μ architecture	0.749	1.80
ALU	0.000526	0.00132
Total (E_c)	2.70	7.07

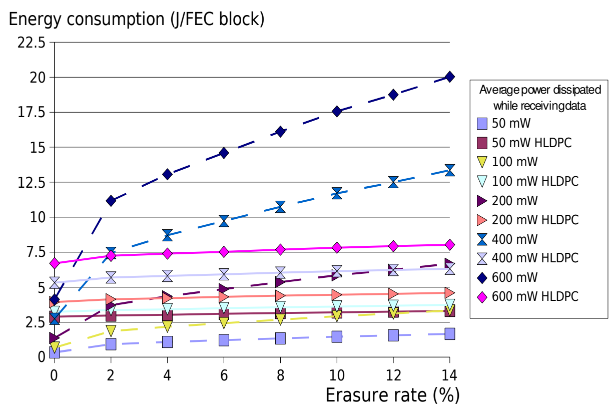

Figure 6. Energy consumption comparison between uncoded and HLDPC coded transmission

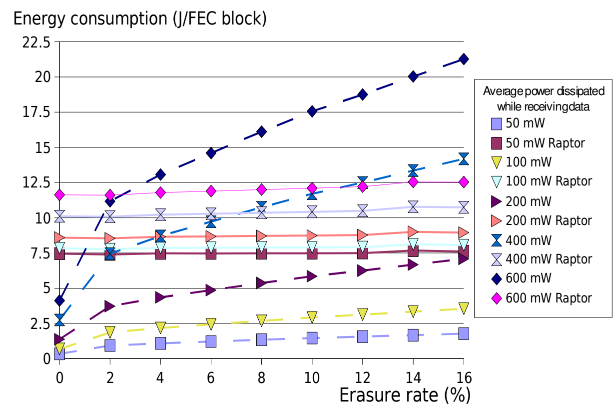

Figure 7. Energy consumption comparison between uncoded and Raptor coded transmission

where p_e is the probability of an erasure, and k is the number of information symbols in the transmitted object. To clarify the equation, the probability of all symbols being correct at a transmission overhead of ϵ_0 is $(1 - p_e^{\epsilon_0})^k$. Therefore, the expression inside the brackets signifies the probability of all symbols being correct at a transmission overhead of *exactly* ϵ_0.

Using equation 9 we can now calculate the receiver gain when AL-FEC is used. Figures 8 and 9 present the obtained receiver coding gain when the HLDPC and Raptor codes are used. As the use of AL-FEC is beneficial only when the receiver coding gain is positive, the comparison of figures 8 and 9 clearly shows better performance for the HLDPC code than the Raptor code. As an example we can see that for an average receiver power consumption $\bar{P}$ of 200mW, the HLDPC code is more efficient than a system without AL-FEC for erasure rates of 4% and upwards. On the other hand, for the same average receiver power consumption the Raptor code is unefficient compared to a system without AL-FEC for all the erasure rates.

It is also important to note that the simulated strongARM SA-110 processor is becoming an outdated processor. As technology evolution since the late 90's concentrated efforts in developing more energy efficient processors, we can expect that with modern processors the receiver coding gain when using AL-FEC could reach positive values for even smaller erasure rates than the one presented on figures 8 and 9.

7. Conclusions

In this paper, we evaluated the use of AL-FEC techniques for achieving error free delivery of data objects in a DVB-H system. The alternative to using AL-FEC is to retransmit the data in a data carousel thereby waiting, in the worst case, for several carousel rounds before all the data is received without errors. AL-FEC decoding is performed in the receiver general purpose host processor. In order to be efficient from an energy point of view, the energy consumed by the host processor for handling the application layer coding should be smaller than the energy consumed by the receiver device for receiving the extra carousel rounds.

Two AL-FEC codes, the HLDPC code and Raptor code, were run in an emulator system, from which detailed information on energy dissipation could be obtained. The energy used for the AL-FEC codes was compared to the energy needed for receiving additional carousel rounds. As the exact energy performance figures for the receiver equipment (frontend) was not known, a set of different average power dissipations were used for the simulations. We believe that this set of average power dissipations covers the range of most receiver equipment characteristics.

Depending on the AL-FEC code and the erasure rate, the receiver coding gain was in the region of -9 to 4 dB. This shows that the energy used by the AL-FEC codes is of the same magnitude as the energy needed for receiving additional carousel rounds. On the other hand, by using AL-FEC codes the transmission overhead is reduced, leading to faster downloading for the end-users. Moreover, the transmission bandwidth is reduced, because the number of required carousel rounds containing the same data is reduced. Thus, using AL-FEC codes is an appealing approach.

The HLDPC and Raptor codecs used in this work, were originally implemented in a PC environment in ANSI C++, using rather naïve software engineering. For example, dynamic memory allocations have been frequently used, spe-

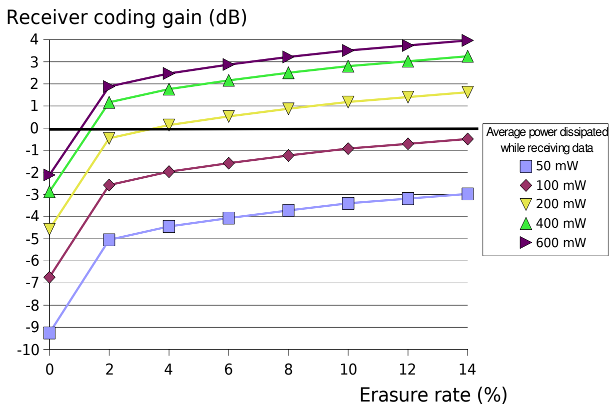

Figure 8. Receiver gain when HLDPC is used

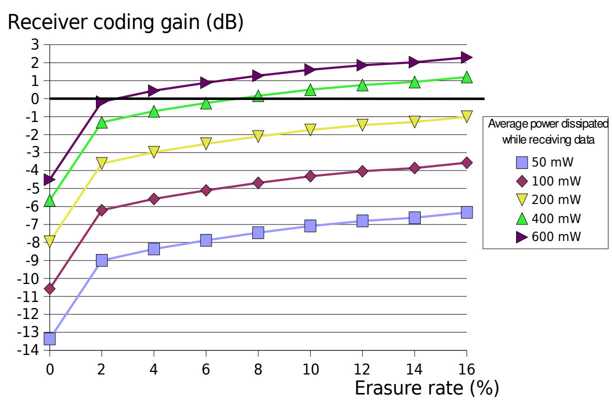

Figure 9. Receiver gain when Raptor code is used

cial processor instructions for optimizing performance have not been used. This code was then recompiled for the emulator framework. Optimizing the code in general, and specific optimizing for the target processor architecture would certainly give some additional gain.

Future host processor architectures in mobile handsets will furthermore be more energy efficient, hence increasing the receiver gain. Receiver chipsets for DVB-H will of course also be more energy efficient, but assuming that the host processor development will be faster the experiments presented in this paper shows that AL-FEC codes are already fully applicable technologies, providing time and transmission bandwidth savings at practically no extra cost in the receiver.

References

[1] D. C. Burger and T. M. Austin. The simplescalar tool set, version 2.0. Technical Report CS-TR-1997-1342, 1997.

[2] ETSI TR 102 469 V1.1.1. *Digital Video Broadcasting (DVB): IP Datacast over DVB-H: Architecture*, 2006.

[3] ETSI TS 102 472 V1.2.1. *Digital Video Broadcasting (DVB): IP Datacast over DVB-H: Content Delivery Protocols*, 2006.

[4] IETF RFC 3926. *FLUTE - File Delivery over Unidirectional Transport*, 2004.

[5] K. Nybom and J. Björkqvist. Hldpc codes - low density, low complexity, efficient erasure correcting codes. In *Proceeding of the 13th European Wireless Conference*, Apr 2007.

[6] J. Peltotalo, S. Peltotalo, J. Harju, and R. Walsh. Performance analysis of a file delivery system based on the flute protocol. *Int. J. Commun. Syst.*, 20:633–659, 2007.

[7] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6):2551–2567, June 2006.

[8] Sim-Panalyzer. http://www.eecs.umich.edu/˜panalyzer.

[9] J. Whitham. http://www.jwhitham.org.uk/simplescalar/.