Performance Evaluation of Power Saving Strategies for DVB–H Services using adaptive MPE–FEC Decoding

Edith de Diego Balaguer, Frank H.P. Fitzek, Ole Olsen
Aalborg University, CTIF
Denmark
Email: {edith|ff|oo}@kom.aau.dk

Morten Gade
Freescale
Denmark
Email: mgade1@freescale.com

***Abstract*— DVB–H is an upcoming technology for service distribution on mobile and wireless handhelds. For a huge market success three main key feature are important such as the services provided, the costs involved and the stand–by time as a level of personal freedom. In this paper we investigate the power saving potentials for DVB–H services. By means of simulation we have shown that for different video sequences power saving potentials between 17% and 22% compared to the original transmission power can be achieved. The proposed mechanism is fully standard compliant.**

I. Introduction and Motivation

DVB–H stands for *Digital Video Broadcasting: Handhelds* and is an ETSI standard for Digital Video Broadcasting providing an efficient way to convey multimedia IP services over digital terrestrial broadcasting networks towards mobile wireless handhelds. DVB–H is based on, and expands, the terrestrial digital TV standard DVB–T, adding new features that fulfill the demands of mobile and battery–operating handhelds such as power–saving operation. Minimizing the battery consumption is generally a critical aspect for the wireless handhelds. Therefore, the aim of this work is to study the receiving processes and its impact on the handheld battery consumption for DVB–H services. Battery consumption was already taken into consideration when introducing time–slicing in the DVB–H standard. With time slicing, the data of the DVB–H service is not broadcasted continuously but data is bundled in ”bursts” at a high data rate. This enables the terminal to switch off the mobile receiver between data bursts. On top of the time slicing we introduce a novel power saving strategy based on the forward error correction part. By means of simulation we investigate the power savings for our new schemes for various DVB–H services.

II. DVB–H Insights

In DVB–H, the contents to be transmitted come in the form of IP datagrams, which are passed to the Multi–Protocol Encapsulation with Forward Error Correction (MPE–FEC) entity. It is a method to deliver Reed Solomon (RS) parity data for datagrams delivered on Multi–Protocol Encapsulation sections. This is accomplished through the introduction of an additional level of error correction at the MPE layer. By adding parity information calculated from the datagrams and sending this parity data in separate MPE–FEC sections, error-free datagrams can be achieved after the MPE–FEC decoding despite a very bad reception condition. The MPE–FEC frame is arranged as a matrix with 255 columns and a flexible number of rows as given in Figure 1. The maximum allowed value for the number of rows is 1024, which makes the total MPE–FEC frame size almost 2 Mbits as each position in the matrix hosts one information byte. The possible number of rows is 256, 512, 768 and 1024 [1], [2]. The left part of the MPE–FEC frame, consisting of the 191 leftmost columns, is dedicated for IP datagrams and possible padding, and is called the application data table. The right part of the MPE–FEC frame, consisting of the 64 rightmost columns, is dedicated for the parity information of the FEC code and is called the RS data table.

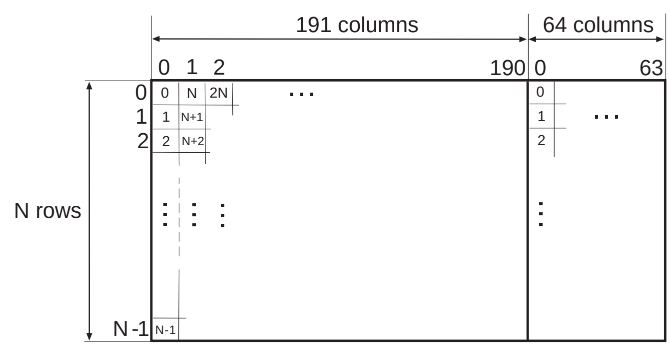

Fig. 1. Addressing of byte positions with 256 columns and N rows.

As given in Figure 2, IP datagrams are introduced one by one, starting with the first byte of the first datagram in the upper left corner of the matrix and going downwards to the first column. The length of the datagrams may vary arbitrarily from datagram to datagram. The maximum datagram size is 4080 bytes (including the header) [3].

Immediately after the end of one datagram the following datagram starts. If a datagram does not end precisely at the end of a column, it continues at the top of the following column.

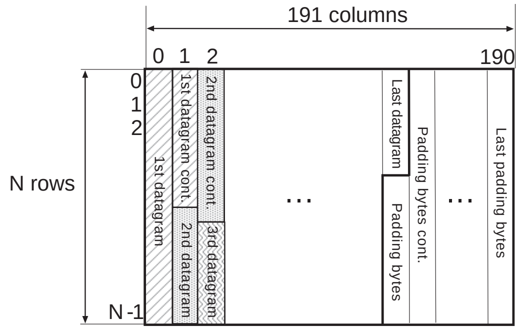

Fig. 2. The layout of the application data table

When all datagrams have been entered in the application data table, all unfilled byte positions are padded with zero bytes, which makes the leftmost 191 columns completely filled. These padding columns are used only for calculation of parity bytes and will not be transmitted later over the air interface.

Each row of the RS data table contains one RS codeword generated using Reed-Solomon code (255,191,64) consisting out of 64 parity bytes generated with all the leftmost 191 columns filled. For every row of 191 bytes of the filled application data table, the Reed–Solomon code generates one row of 64 bytes that is written in an horizontal way in the RS data table until filling the N rows as given in Figure 3.

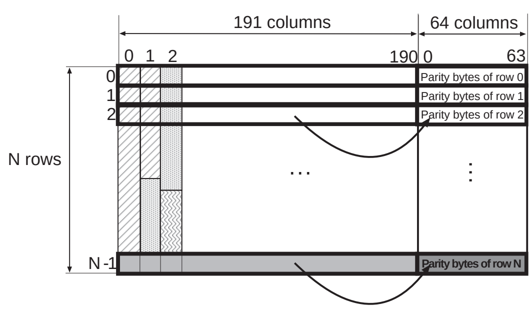

Fig. 3. The MPE–FEC frame with the RS data table layout

For the transmission over the air interface the MPE–FEC frame will be encapsulated in MPE and MPE–FEC sections (see Figure 4). The sections will be carried in Transport Stream (TS) packets (see Figure 5). The IP datagrams are carried in MPE sections in compliance with the DVB–H standard. Each delivered MPE section shall contain one complete valid network datagram with a valid IP header. For each MPE–FEC frame, at least one MPE section shall be delivered. For encapsulated datagrams, the maximum section size is 4096 bytes (4080 bytes of datagram + 16 bytes of section overhead), matching with the maximum number of bytes in an MPEG–2 private section [4]. The last section with application data table information contains in its header a table_boundary flag which indicates the end of the IP datagrams within the application data table. The RS data table information is carried in MPE–FEC sections. Each MPE–FEC section carries exactly one column of RS data table. The number of application data padding columns is signalled with 8 bits in the header of the MPE–FEC sections; the allowed range is 0 to 190. This value is needed only for the decoding. Each section (MPE of MPE–FEC) carries in the section header a start address for the payload: the byte position in the application data table or in the RS data table of the first byte of the section payload. At the end of every MPE and MPE–FEC sections, there is a CRC–32 field calculated over the entire section to detect the erroneous sections at the receiver side.

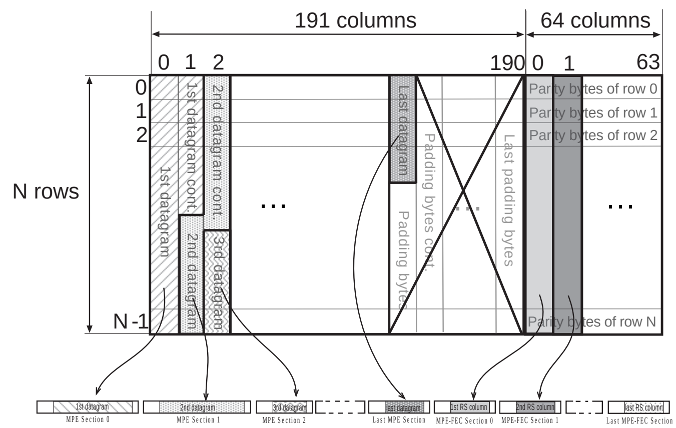

Fig. 4. Content layout of MPE and MPE–FEC sections

The resulting MPE–sections are sent in the same order as the datagrams were written in the table. For the MPE–FEC sections, they are sent analogously per column order, from the left column to the right column.

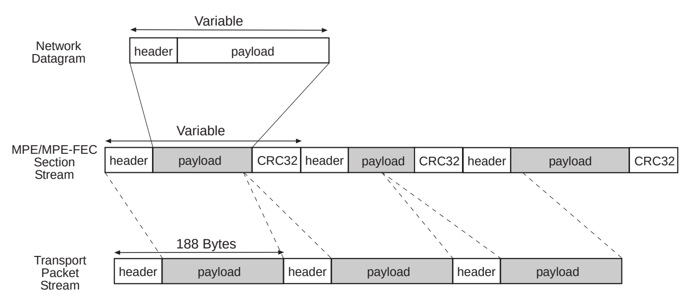

Fig. 5. Relation between Transport Packet Stream, MPE/MPE–FEC section stream and datagram

To accomplish the transmission in TS packets, the obtained stream of MPE and MPE–FEC sections is divided in groups of 184 bytes. The splitted sections are introduced, keeping their ordering, in MPEG–2 transport packets with 4 bytes for the transport packet stream header, constituting packets of 188 bytes length. Depending on the size of the sections, one transport packet payload may contain one or more sections, and one section may be fragmented into one or more transport packet payloads. A burst is defined as a sequence of MPEG–

2 transport stream packets containing in the payload bytes a complete MPE–FEC frame.

When the decoder begins to receive data, it has an empty MPE–FEC frame structure with all its byte positions marked as ”unreliable”. With the byte position address indicated in the header of every received section, the receiver will then be able to put the received datagram in the correct byte positions of the application data table or RS data table. The CRC–32, which detects erroneous sections, will allow to mark these positions as ”reliable” or ”unreliable”. All the empty holes of the MPE–FEC frame will remain as ”unreliable” bytes. If MPE–FEC sections are received, the receiver introduces the number of padding bytes in the application data table, as indicated in the MPE–FEC sections. It marks these padding bytes as ”reliable”. If the receiver did not receive the last MPE section correctly it will have to assume that all byte positions after the last correctly received section until the first padding column is lost data, and mark the corresponding bytes as ”unreliable”. After this procedure, all byte positions within the MPE–FEC frame are marked as ”reliable” or ”unreliable”. With this reliability information, also called *erasure information*, the RS decoder is able to correct up to 64 such bytes per 255-byte codeword. It should be noted that any 64 bytes can be corrected, i.e. it does not matter whether 1 bit or all 8 bits are in error. If there are more than 64 unreliable byte positions in a row, the RS decoder will not be able to correct anything and will therefore just output the bytes without error correction. It will have anyway the knowledge about the positions of any remaining byte errors. If a datagram is only partly corrected, the receiver will be able to detect this and (optionally) discard this datagram. If erasure information is not used, the RS decoder will allow correcting up to 32 random erroneous bytes in a received word of 255 bytes.

III. Proposal of Novel Power Saving Strategy

Even though DVB–H offers a good power consumption improvement in relation to DVB-T, further power saving improvement will be vital for the device battery life. There may be two main ways to improve the power consumption:

- Omitting RS columns in reception processing
- Carrying out the half RS decoding in post-processing

The erasure information will be fundamental for any of the quoted power saving proposals. In this paper, we will introduce the basis of this two ways to save power. We will also relate them to the erasure information that is the system feature that will allow us to have a decision criterion to apply them.

As we have seen in the previous section, if all the application data table sections have been received correctly the receiver does not need to receive any MPE–FEC sections and can be switched off. If the application data table has any error, then all the MPE–FEC sections are received to accomplish the error correction. As time between bursts (off time) is directly related to power saving, receiving only the necessary RS columns and switching off the receiver without receiving the remaining RS columns would result in a power consumption improvement. The number of columns that may be omitted will depend on the number of errors/erasures in the MPE–FEC frame and the forward error correction capability of the decoder.

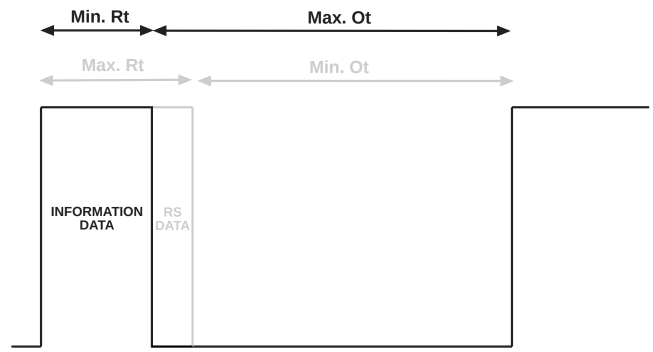

Fig. 6. Maximum and minimum Receipt time (Rt) and Off-time (Ot)

Figure 6 shows the possible minimum and maximum Off-times that depends on the quantity of data received from the burst. The maximum Off-time and maximum power saving will occur when only the information data (the data corresponding to the application Data Table) is received and the receiver switches off without receiving any RS columns. No power saving improvement will be accomplished when the Off-time will be minimum, that is when all the complete burst is received (Information data + RS data). In that case, the Receipt time (Rt) will be maxim and will be the same as the Burst duration (Bd). Data and signalling are mixed in the same burst. But since not the full signalling data is needed, it will not be a problem to power down before.

Considering the MPE–FEC frame architecture, the parity information constitute the 25% of its content. As a result, omitting one RS column would suppose omitting the 0,39% of the MPE–FEC frame content. Since the number of overhead bytes of the sent stream vary depending on the size of the included datagrams, it is difficult to determine the relation between omitted columns and exact power saving in terms of percentage with respect to standard DVB–H. If we do not consider any signalling in the burst, the maximum power saving may vary between 19% and 25,4%. To simplify, we can approximate the power saving percentage as the represented percentage of the RS columns in the MPE–FEC frame architecture.

In another way, to give an absolute value that would allow us to know how much power saving consumption is achieved, the duration of RS column transmission may be calculated as the ratio of the number of TS packets with complete omitted RS information (multiplied with 188 byte) divided by the burst bandwidth. Depending on the characteristics of the terminal device, a certain duration will be translated in a specific power consumption.

At the decoder side, omitting one RS column is equivalent to adding an error in each omitted byte. As seen previously, the RS decoder is able to correct up to 32 random erroneous bytes and 64 random erroneous bytes when reliable erasure information is used. So, to attain an output without any errors, the sum of the errors/erasures plus the number of omitted rows will not have to exceed 32 and 64 bytes respectively.

The Reed-Solomon decoding process is constituted by the following steps [8]:

- *Syndrome Calculation*. A similar calculation to parity calculation.
- *Finding the Symbol Error Location*. This involves finding an error locator polynomial and finding the roots of this polynomial. Simultaneous equations with 32 unknowns have to be solved.
- *Finding the Symbol Error Values*.

When we talk about ”Half RS Decoding”, we refer to the fact of using the erasure information provided by the CRC–32 and avoid the calculation of the error locator polynomial. In this way, power saving will be achieved in the post-processing stage. Thus, two RS decoding ways may be used:

a) *”Half RS Decoding” (Use of the erasures principle)*- In this case, the information provided by the CRC–32 will allow to mark the location of the erroneous bytes. Maximum 64 erasures will be able to be corrected.
   - Advantages: Faster processing since only half of the RS performance will be needed (it will not be necessary to calculate the error location polynomial or evaluate it). This characteristic will allow a supplementary power saving and cheaper decoders.
   - Drawbacks: An important number of correct bytes may have been marked as unreliable. So, the decoder may need much more RS columns than necessary or may not be able to correct a quantity of errors that could have been corrected with a full RS decoding.

b) *”Full RS Decoding”* - The RS decoding will be accomplished without any erasure information. Maximum 32 errors will be able to be corrected.
   - Advantages: No additional errors are added.
   - Drawbacks: The RS decoding process will be slower. It will need more power consumption and will be more expensive. If errors occur a minimum of 32 MPE–FEC RS columns should be received.

At the receiver side, the erasures are the only information that may allow us to deduce the minimum necessary number of RS columns we have to receive to accomplish the Forward Error Correction and help us to decide whether the ”Half” of ”Full” RS decoding is more suitable. Likewise, after the correction, the remaining ”unreliable” bytes will be the unique elements that may allow us to do a quality evaluation and help us to provide a threshold to attempt handover. If we base the decoding decisions on CRC–32 information, the most important expectation from this code is a very low probability for undetected errors. The probability of undetected errors also depends on the error distribution and the data length. The asymptotic limit could be assumed as $\frac{1}{2^{32}} = 2.3 \cdot 10^{-10}$ of failing to detect an error in the data [7]. The chances of an erroneous data going undetected can be supposed small but the RS decoder can also improve this reliability.

In addition to the CRC–32, which detects erroneous sections, the RS decoder also very reliably detects erroneous TS packets [1], [2] since the TS packet structure has its own CRC. If the MPEG–2 demultiplexer discards erroneous packets it could be designed not to build sections, which contain lost TS packets. In this way almost only correct sections would be built and the role of the CRC–32 would be to provide additional error detection functionality, which normally is not needed. In other words, sections can contain up to 4080 bytes whereas TS packets only 184. Therefore, if we know from the TS layer that a TS packet is wrong, we only need to mark 188 bytes as ”unreliable”, whereas up to 4080 bytes must be marked as ”unreliable” if we only rely on the section CRC–32.

As we have seen, incorrect sections may be discarded and not placed in the MPE–FEC frame. This could be because of the decoder configuration (to simplify the processing), due to irrecoverable errors in TS packets (such as wrong packet identification flag) or simply because of lost packets. In all these cases, the ”unreliable” bytes of the MPE–FEC frame will coincide with the real errors, or at least they will be very close to them. The erasure information and the output quality evaluation will be very precise. If the RS decoder does not discard erroneous sections, it will try to put the ”unreliable” data in the corresponding MPE–FEC frame position assuring to not overwrite any ”reliable” byte. Since the CRC–32 marks entire sections, all the bytes of one datagram may be marked as ”unreliable” even if there is only one unique erroneous byte. So, we will have to keep in mind that we may have more ”unreliable” bytes than real erroneous bytes. The erasure information will be far from the real errors and the output quality evaluation will be imprecise and will provide worst error rates than the reality.

IV. Simulations Description

In order to study the proposed topics for an improvement in power saving, simulations were developed using Matlab software tool. As MPEG–4 video is the main DVB–H application, the studies were developed based on this kind of data. Every IP datagram of the MPE–FEC frame consists out of 40 bytes overhead (20 bytes for IP header in IPv4 + 8 bytes for UDP header + 12 for RTP header) followed by the payload represented by MPEG–4 frames with the size given in [5]. MPEG–4 frames have variable length frames whose mean size will be very different depending on the quality specified by the encoding process, the height and width of the picture size and also in the amount of ”action” in the movie. To choose the MPEG–4 frame sizes, we used average results obtained in MPEG–4 traces study [6] for variable bit rate video streaming in CIF[1] and QCIF[2] format. These formats fit to small screens and they are the main candidates for DVB–H [9].

However, since there is a wide variety of cases for MPEG–4 video services and in order to develop a general study, we applied fixed size datagrams of 113 bytes in our very first

[1]CIF (Common Intermediate Format): Video display standard with a resolution of 352 x 288 pixels. Suitable for PDAs.

[2]QCIF (Quarter Common Intermediate Format): Smaller variation of the CIF video standard. It yields a 176 x 144 resolution. Suitable for mobile phones.

investigation. The number of rows in the MPE–FEC frame is 1024.

The wireless errors were modelled by directly applying them in the MPE–FEC frame. This way, we did not consider physical layer complexity and we focused on MPE–FEC frame level. In order to add errors, we treated two cases based on the explanations. First, a uniform bit error random distribution that represents uncorrelated errors in the MPE–FEC frame. These uncorrelated errors at the MPE–FEC level could have been correlated errors at channel level that are afterwards spreaded because of interleaving. This distribution will cause a certain erasure pattern depending on the different datagram sizes. Next, we represented the case of errors due to packet loss or packet discard. All the bytes of the erroneous datagrams were marked as ”unreliable” and we obtained a kind of bursty error distributions. In that case, the real erroneous bytes matched with the ”unreliable” bytes.

When applying erasure principle, a random error distribution will represent the worst of the cases because the error distribution will be spreaded. Then, all the packet bytes will be marked as ”unreliable” even if only a few of its bytes are erroneous. The datagram size and the number of rows of the MPE–FEC frame will produce an effect on the erasure information accuracy. So, this factors will be important to bear in mind to obtain the best system behavior.

In order that the decoder receives the minimum necessary RS columns, we base the switch-off decision on the erasure information. It will be necessary to see the quantity of ”unreliable” bytes in a row and then go receiving the necessary RS columns to accomplish the forward error correction. Note that not receiving any RS column will be equivalent to 64 erroneous bytes per row. So, the number of necessary RS columns to receive to accomplish the complete correction are at least:

For the ”Full RS Decoding”: equal to 32 plus the maximum number of ”unreliable” bytes per row until the switching-off moment. This will assure that the maximum number of ”unreliable” bytes in a row will be at most equal to 32. It will be necessary to receive at least 32 RS columns in order to be able to perform the error correction.

For the ”Half RS Decoding”: equal to the maximum number of ”unreliable” bytes per row until the switching-off moment. This will assure that the maximum number of ”unreliable” bytes in a row will be at most equal to 64 and the error correction will be possible.

A. General Investigation

To illustrate the obtained results, we applied the ”full RS decoding” and the ”half RS decoding” to correct an MPE–FEC frame with the same erasure information in both of the cases. The following simulations were developed for an MPE–FEC frame of 1024 rows and datagrams of 113 bytes length. These sizes allowed us to show for a same MPE–FEC frame a wide range of possible results when different bit error rates were applied.

Figure 7 shows how the ”full” RS decoding needs to receive at least 32 columns to be able to accomplish a correction. In that case since the number of errors by row is very low, the decoder only needs a few more RS columns than 32. As far as the half RS decoding is concerned, it loses its correction capabilities as the number of ”unreliable” bytes increases. The transition between the capability to begin to correct the errors and have the complete correction accomplished (the transition between maximum BER and zero BER) bears a direct relation to the uniformity on the number of errors or ”unreliable” bytes between the rows. If ”full” RS decoding is applied, the relation is based on errors and if half RS decoding is applied, it is based on erasures. Therefore, the nearer the quantity of errors/erasures in the rows is, the more abrupt will be the transition. Notice that in the graphs the maximum BER may be slightly below the indicated BEP because the BEP refers to the entire MPE–FEC frame and the BER only to the application data table.

B. Application of MPEG–4 video traffic over DVB–H

In order to verify the theoretical results, we applied the explained power saving proposals to true videos encoded with different MPEG–4 video sequences. The results obtained for the movie ”Star Wars IV” are described below in more detail. The MPEG–4 traces of the movie ”Star Wars IV” [6] contain both ”flat” scenes and ”action” scenes: therefore a variability in the MPEG–4 frames size is provided. The evaluated data consisted of a QCIF medium quality video stream of 1 hour of duration. Following the standard, every MPEG–4 frame was carried in independent IP datagrams with 40 overhead bytes and MPEG–4 frames bigger than 4040 were fragmented. A total of 89998 MPEG–4 frames were received and corrected through 782 MPE–FEC frames of 256 rows. We applied a random error distribution with probability 10^{-5}. The resulting output did not contain any error after the forward error correction applying half RS decoding and provide a mean power saving values due to columns omission of $20,84\%$. In order to compare the results we apply exactly the same error pattern to an MPE–FEC frame with constant length datagrams equal to the mean frame size (388 bytes). In this case we obtain a power saving of $22,75\%$. The power saving in the later case are slightly higher than the mean value for the movie because there are no packets as big as in the movie case.

In Figure 8 the power savings for different video sequences versus the mean frame size in byte are given. We have chosen six different video sequences to analyze the potential power savings. The power saving values obtained are between 17% and 22% for the *Jurassic Park* video sequence and the *Star Wars IV* video sequences, respectively. Our first observation of the achieved results is that the larger the mean frame size becomes, the lower the power saving. This is based on the fact that for shorter frames the errors have less impact than for the longer frames, because less bytes are marked as unreliable. This is especially true if frames are larger than one MPE–FEC column. Note, that the two sequences *Silence of the lambs* and *Mr. Bean* have different power saving values even though

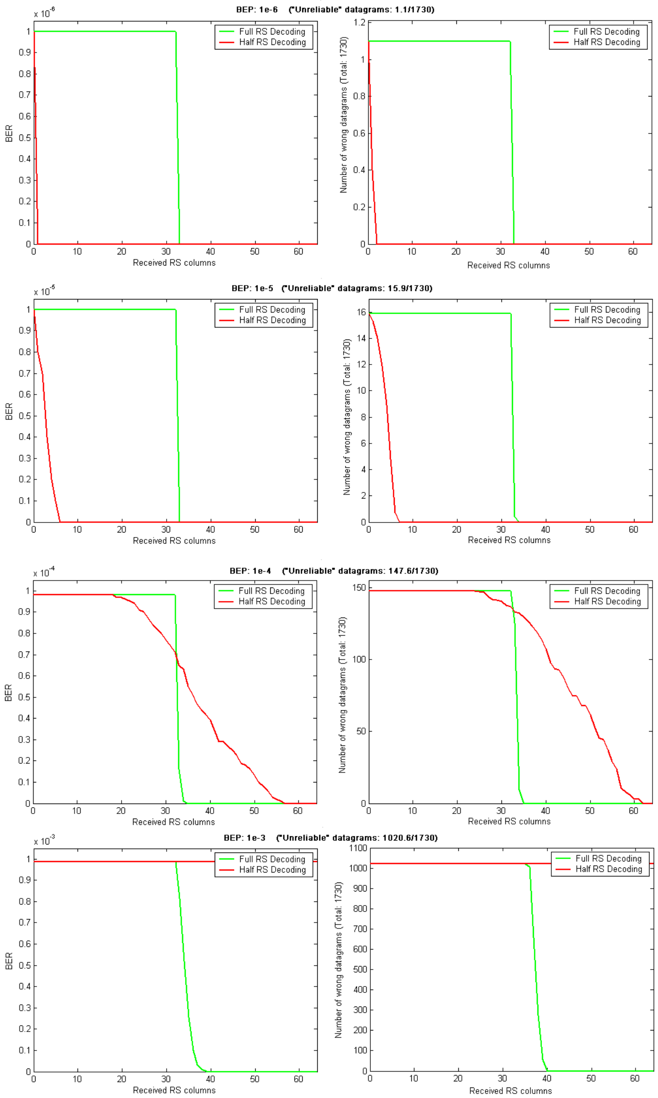

Fig. 7. Remaining BER and Remaining wrong datagrams in recovered application table - "Unreliable" bytes caused by uncorrelated errors (MPE–FEC frame: 1024 rows - Datagrams size: 113 bytes).

that they have more or less the same mean frame size. The differences is caused by the larger co–variance value of the *Silence of the lambs* than that of *Mr. Bean*, which is 1.21 and 0.97, respectively as given in [6]. This fact can be used to taken into consideration for the video editing process and encoding leading to an adaptive media approach for DVB–H. The goal should be to achieve small frames with low co-variance in the frame size. This simple encoding rules can be used for the encoding process of DVB–H video services.

V. Conclusion

In this paper we have shown a new strategy for power saving in DVB–H systems using the MPE–FEC coding. By column omission we can yield power saving values up to 25% in the best case. In this paper we have shown for six different video sequences realistic power saving values between 17% and

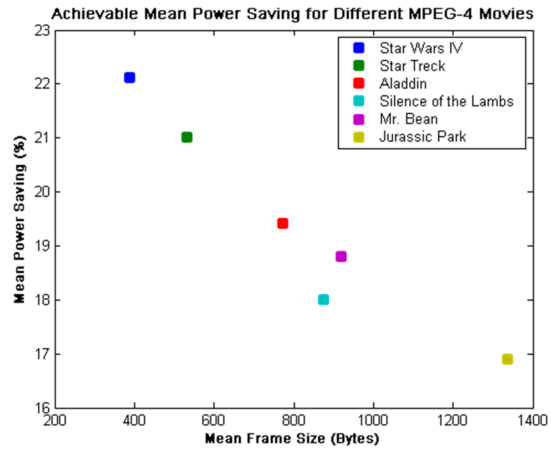

Fig. 8. Power savings for different video sequences versus the mean frame size in byte.

22%. The final power saving values depend on the channel errors and the traffic pattern of the served video sequence. Therefore *DVB–H adaptive media* can be one solution for larger power saving potentials. The proposed scheme is fully standard compliant and can therefore implemented in all DVB–H handhelds.

References

[1] ETSI EN 301 192 v1.4.1 (2004-06): *Digital Video Broadcasting (DVB)*; DVB specification for data broadcasting.

[2] ETSI EN 302 304 v1.1.1 (2004-06): *Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB–H).*

[3] DVB Document A079 (April 2004): *IP Datacast Baseline Specification, PSI/SI Guidelines for IPDC DVB-T/H Systems.*

[4] ISO/IEC 13818-1: *Information technology - Generic coding of moving pictures and associated audio information: Systems.*

[5] RFC 3016: *RTP Payload Format for MPEG–4 Audio/Visual Streams*, Nov. 2000.

[6] Seeling P., Reisslein M. and Kulapala B.: *Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial*, IEEE Communications Surveys and Tutorials Vol. 6, No. 2 Pages 58–78, Third Quarter 2004.

[7] Barr M.: *For the Love of the Game* - Embedded Systems Programming, December 1999 , pp. 47-54.

[8] Riley M. and Richardson I.: *Reed-Solomon Codes - An introduction to Reed-Solomon codes: principles, architecture and implementation.*

[9] Henriksson J.: *DVB–H Outline*, Dec. 2003.

[10] Gringeri S., Egorov R., Shuaib K., Lewis A. and Basch B.: *Robust Compression and Transmission of MPEG–4 Video*, 1999.