On the use of packet-level FEC and data carousels for the delivery of broadcast/multicast services to mobile terminals

M. Chipeta, M. Karaliopoulos, B.G. Evans and R. Tafazolli
Mobile Communications Research Group,
Center for Communications Systems Research, University of Surrey
Guildford, GU2 7XH, United Kingdom
Email: m.chipeta@surrey.ac.uk

Abstract—In broadcast wireless networks, the options for reliable delivery are limited when there is no return link or a return link is not deemed cost-efficient due to the system resource requirements it introduces. In this paper we focus our attention on two reliable transport mechanisms that become relevant for the non real time delivery of files: packet-level Forward Error Correction (FEC) and data carousels. Both techniques perform error recovery at the expense of redundant data transmission and content repetition respectively. We demonstrate that their joint design may lead to significant resource savings.

Keywords—packet-level FEC; data carousels; reliable multicast

I. INTRODUCTION

Reliable transport is amongst the fundamental requirements for the delivery of point-to-multipoint services in wireless environments, where transmission errors are more frequent than in wired networks. The mobility of the users exacerbates the situation: signal reception quality is then affected by fading effects [1], while in cellular networks, cell handover may lead to temporary loss of connectivity. Data loss may become unacceptable for the application requirements, so that some form of error recovery is mandatory. In many cases, a return path does not exist or is not efficient to provide, thus limiting the available alternatives for error recovery. Packet-level Forward Error Correction (FEC) [2]-[4] and data carousels [5] [6] are possible mechanisms in this context since they can work in push-type delivery mode without any user feedback.

The value of the two techniques is reflected in their broad use in several wireless systems and in ongoing standardization work. The Internet Engineering Task Force (IETF) has specified a FEC building block for use in reliable multicast protocols [7] [8], whilst in the European Digital Video Broadcasting (DVB) standard for broadcasting to mobile handheld terminals (DVB-H or DVB-Handheld [9]) packet-level FEC is adopted in the Multi-Protocol Encapsulation layer, an adaptation layer lying above the radio transmission layers of the MPEG2/DVB transmission scheme. Data carousels, on the other hand, are one of the standard ways for DVB implementation over several media (cable, satellite, terrestrial radio) [10]. In Terrestrial Universal Mobile Telecommunications System (T-UMTS), the standard for third generation mobile cellular networks adopted in Europe and Asia, including Korea and Japan, the use of packet-level FEC and data carousels is considered in the context of Multimedia Broadcast Multicast Service (MBMS) framework [11]. The support of point-to-multipoint service over cdma2000, the third generation radio interface adopted in North America, envisages the encoding of radio access layer frames before forwarding them to the physical layer [12]. Finally, both techniques are investigated for the design of the reliable transport layer in the emerging Satellite Digital Multimedia Broadcasting system [13], a hybrid satellite-terrestrial system that deploys a unidirectional satellite component for the delivery of point-to-multipoint services to large, geographically dispersed audiences, in close synergy with terrestrial mobile cellular networks.

In the following, we are not going to restrict our discussion to any of the aforementioned systems. We will rather investigate the two techniques, specifically packet-level FEC and data carousels, resorting to analytical approximations widely used in the literature. We review some of this work in the following section before presenting our contribution—an investigation of encoded data broadcasting— in section II. Section III presents numerical examples demonstrating the benefits of superimposing packet-level coding on data carousels. The responsiveness of the data broadcast system increases and experiences a maximum for certain values of coding overhead, beyond which performance deteriorates. We conclude the paper in section IV.

II. PROBLEM FORMULATION

A. Broadcast scheduling and data carousels

In a so-called ‘push’ system, the broadcast network transmits data towards the users not as a result of an explicit request on behalf of one or more users, as in a ‘pull’ system, but rather under the expectation that the transmitted content will appeal to some of the target users. The network cannot know which item each user is interested in; nevertheless, on the basis of user subscription profiles or historical data, it may know what is the relative demand for various data at several levels of detail: for example if audio is more popular than video data but also whether video clip X is more popular than song Y.
Consider, therefore, that the network transmits $M$ different data items over a wireless broadcast link. An example may be a file but it may also be a set of files or some other way of data grouping.

Each item $i$ is characterized by its length $L_i$, measured in packets, and a demand probability $q_i$, with $\sum q_i = 1$, reflecting the interest of users in this particular content. With broadcast scheduling, these $M$ items are organized in a schedule, which is repeated continuously over the air, with the aim to minimize the mean response time $S$ over all items. The response time for item $i$ is defined as the time that elapses between time instant $t_i$ when the user expresses his desire to access a certain item, to time $t_{i1}$ when the item is retrieved from its schedule and is stored at his terminal (Fig. 1).

In fact, at time $t_i$, the user will interact with his terminal via the Man-Machine-Interface (MMI) and ask for the particular data item. The user equipment will then read a predefined radio channel where the announcements of broadcast data are made and will tune to the correct radio channel to retrieve the requested data. How quickly this happens is a function of the transmission medium quality and how frequently each data item is requested data. How quickly this happens is a function of the broadcast medium quality and how frequently each data item appears in the schedule; in an optimum schedule design, the item transmission medium quality and how frequently each data item appears in the schedule is closely related to the item demand probabilities $\{q_i\}$ [14]. In particular, it is shown in [15] that under the optimum broadcast schedule design strategy, the average response time is given by:

$$ S_{av} = \frac{1}{2} \left( \sum_{i=1}^{M} \sqrt{q_i \cdot L_i \cdot (1 + 2r)} \right)^2, $$

where $r_j$ is the mean number of required re-appearances of data item $i$ in the schedule after its first appearance, so that it is fully retrieved in the presence of data loss. In general, data loss may be due either to congestion or transmission errors. $\{L_i\}$ is the set of item lengths normalized with respect to the minimum item length

$$ L_i = \frac{L_i}{\min L_j}. $$

Note that, using (2), the response time is measured in time units equal to the transmission time $T$ of the minimum-length item

$$ T = \frac{\min L_i}{C}. $$

where $C$ is the link capacity in packets per second.

At the user terminal, there are two possibilities for retrieving an item:

1) One-shot retrieval of data items: upon the appearance of each item in the schedule, the receiver will either retrieve correctly the whole item or not. When part of the item is not retrieved, correctly retrieved item parts are discarded rather than stored and the retrieval procedure begins from scratch upon the next item appearance in the schedule. If $p$ denotes the uniform packet error rate and $N_r$ the number of item packets, the probability $p_i$ that an item is not fully retrieved within one appearance in the schedule becomes

$$ p_i = 1 - (1-p)^{N_r}. $$

and the probability that the full item will be correctly retrieved by its $n$th appearance in the schedule after the user ‘requests’ it becomes

$$ P_i^C(n) = \sum_{k=1}^{n} (1-p_i) \cdot p_i^{k-1} \quad n \geq 1 $$

(5)

to yield an $r_j$ equal to

$$ r_j(p, N_r) = \frac{p_i}{1 - p_i}. $$

(6)

2) Cumulative retrieval of data items: in contrast to the previous scenario, the receiver stores that part of the item that was received correctly, even if the whole item is not retrieved correctly. This requires appropriate data packaging that will allow the identification of correctly received data parts and later, when all item parts are correctly retrieved, the reassembly of the item.

Now, the probability that the item will have been reassembled by its $n$th appearance in the schedule is

$$ p_i^C(n) = (1-p^n)^{N_r}, \quad n \geq 1 $$

(7)

while $r_j$ becomes

$$ r_j(p, N_r) = \sum_{k=0}^{n} \left[ (1-p^{k+1})^{N_r} \right] $$

(8)
Parity Packets

resilient to loss. The original data are structured into blocks of information is added to the original data to make them more robust, with FEC redundant transmission, in order to avoid excessive overheads in terms of encoding/decoding speed.

In the following, we only consider systematic small block erasure codes, such as Reed-Solomon (RS) codes, where a receiver needs to collect at least \( k \) packets out of the \( n \) of the FEC block to recover the original \( k \) packets. In contrast, large codes require more than \( k \) packets to recover a given FEC block [7]. We can define the code rate \( R = k/n \) and its inverse, the stretch factor \( SF = n/k \), which is a direct measure of the redundancy - hence, the additional capacity requirements introduced by FEC.

For systematic small block codes \((k, n)\), the FEC block error rate \( FBLER \), namely the probability not to decode correctly a single FEC block, is

\[
FBLER = \sum_{j=0}^{k} \frac{k!}{j!(k-j)!} p^j (1-p)^{k-j}
\]

where \( p \), as in (4) is the residual uniform packet error rate. Therefore, the probability to correctly retrieve a file consisting of \( N_B \) FEC blocks, becomes

\[
P_F;100 = \left( 1 - \sum_{j=0}^{k-1} \frac{n!}{j!(n-j)!} p^j (1-p)^{n-j} \right)^{N_B}
\]

The erasure correcting capability of the codes is a function of both the SF and the size of FEC block \( k \). Imagine, for example, that we want to encode a file of 200kB into blocks of 500-byte packets. Fig. 5 plots \( P_F;100 \) under four different possible combinations of \( \{k, SF, N_B\} \). It is obvious that, for a given item size, the probability to retrieve correctly the full item increases with both \( k \) and \( SF \).

**B. Packet-level FEC**

Packet-level FEC (hereafter called FEC) has been one of the fundamental building blocks for the support of reliable multicasting in wired networks [7]. Contrary to data carousels, where data transmissions are fully repeated, with FEC redundant information is added to the original data to make them more resilient to loss. The original data are structured into blocks of \( k \) packets and are used to produce the \( n \) packets of the FEC block, which will finally be transmitted to the network. The block can be decoded correctly at the receiver as long as less than \( n-k \) packets are lost (Fig. 4). In general, codes can be classified as large or small [7]. Large codes can support large \( k \) and \( n \), without a severe penalty in the processing complexity and encoding/decoding speeds. With large codes, the size of the original data in a FEC block may coincide with the whole file. On the contrary, small codes are limited to small values of \( k \) and a large file will have to be split into several FEC blocks for transmission, in order to avoid excessive overheads in terms of encoding/decoding speed.
C. Combined use of FEC and broadcast scheduling

We now consider an integrated carousel-FEC transmission scheme: files are organized in broadcast schedules for repeated transmission after being FEC-encoded. In the general case, each file/item may feature its own FEC parameters \((k_i, n_i)\) leading to different FBLERs; moreover, each item is of different length \(l_i\), eventually corresponding to different number of FEC blocks per item \(N_p\). Then, the minimum mean response time \(S\) can be derived as

\[
S_{\text{opt}}(k, n) = \frac{1}{2} \left( \sum_{i=1}^{M} \frac{n_i}{k_i} \left( [1 + 2 r_i(k, n)] \right) \right)^{1/22}
\]  

(11)

where \(r_i\) can still be derived from (6) and (8) for the one-shot item retrieval and cumulative item retrieval respectively, by substituting:

\[
p \rightarrow \text{FBLER}
\]

\[
p_i \rightarrow 1 - P_{F, 100}
\]

\[
N_p \rightarrow N_p
\]

Equation (12) suggests that the optimum response time now becomes a function of the vectors \(k, n\) employed for the encoding of the different items.

Increase of the FEC redundancy has a two-fold impact: on the one hand, for given link capacity, it increases the equivalent item sizes; hence, it increases the average response time in (12). On the other hand, increase of the redundancy increases the probability of one-shot retrieval of the whole file \(P_{F, 100}\). We speculate that the choice of FEC parameters will not be made independently for each item but will rather be rule based. We consider two such rules in the following:

1) Fixed redundancy for all items: the FEC block size and the stretch factor are the same for all items, independent of their size, namely \(k_i = k_j\) and \(n_i = n_j\) for all \(i, j\).

Note that the resulting probability of correct item reception varies according to the item size, i.e.,

\[
P_{F, 100}^i \neq P_{F, 100}^j
\]

In this case, the objective that drives the design, together with the minimization of the mean response time, is the preservation of a target link capacity overhead due to the use of FEC.

2) Fixed probability of one-shot full item retrieval: the stretch factor varies from item to item but not randomly; instead, for a given FEC block size of the \(i\)th item \(n_i\) the size of FEC block \(n_i\) for item \(j\) is adjusted so that

\[
k_i = k_j
\]

and \(P_{F, 100}^i = P_{F, 100}^j\) for all \(i, j\) and \(i \neq j\).

This rule gives higher priority to the fair treatment of the different types of content transmitted over the air, since it guarantees to all items equal probability of full retrieval upon each appearance in the schedule.

III. NUMERICAL EXAMPLES - DISCUSSION

We now present some numerical examples comparing the alternatives described in section II.A for the item retrieval technique and the rules introduced in section II.C for the FEC parameterization for different values of number of items and FEC block length. We consider two distributions for the item demand probability, the uniform and the Zipf with skewness parameter \(\theta\) equal to 1.25, while the item size may either be fixed for all items or vary uniformly between two bounds \(l_0\) and \(l_1\) (Table I).

In all the plots that follow, the average response time is measured in time units equaling the transmission time of the minimum FEC block size.

| TABLE I. DISTRIBUTIONS CONSIDERED FOR THE NUMERICAL EXAMPLES |
|------------------|------------------|
| Item demand probability | Zipf |
| \(q_l = \frac{1}{M}\) , \(1 \leq l \leq M\) | \(q_l = \frac{[([l]/\theta)]}{\sum_{i=1}^{M} [([l]/\theta)]}, 1 \leq l \leq M\) |
| Item length (FEC blocks) | Variable (increasing with \(i\)) |
| \(l_i = c, c \geq 1, 1 \leq i \leq M\) (default \(c=5\)) | \(l_i = \text{round} \left( \frac{[l_i] - l_{i-1}}{M-1} \right) \left([l_i] - l_{i-1}\right) + l_{i-1}\) |
| \(1 \leq l_i < l_{i+1}, 1 \leq i \leq M\) (default \(l_0=1, l_1=10\)) |

A. One-shot item retrieval versus cumulative item retrieval

It was shown earlier in section II.A that, in the absence of FEC, the cumulative item retrieval is clearly better than the one-shot retrieval, since it results in lower response times. This is also clearly shown in the leftmost part of the curves in Fig. 6 and Fig. 7 (\(SF = 1\), i.e., no FEC protection), which plot the mean response time under the fixed redundancy rule for the data item encoding. However, it is interesting to note that as more FEC redundancy is added, the curves corresponding to the two item retrieval methods converge.
B. FEC design options: fixed redundancy versus fixed probability of one-shot full item retrieval

The results shown so far have been obtained assuming fixed redundancy for all items, irrespective of their length. Fig. 8 compares the two alternatives for the FEC redundancy per item. For low redundancy levels, the second rule results in lower response times. Note, however, that this comes at a higher cost in terms of FEC overhead, since the SF value on the x-axis is only the FEC overhead of the smallest item when the second rule is considered; largest items bear higher stretch factor values. Therefore, the gain in response time has to be weighed against the additional redundancy. On the other hand, the optimum response times are practically independent of the followed rule when applying FEC to the items. Note that the respective rule has an impact on response items only when variable-length items are considered.

Table II and Table III list the optimum response times, together with the respective FEC redundancy levels, under the two item retrieval techniques and the two rules for selecting the code redundancy of data items.

Table II. One-shot retrieval \((M = 1000, k = 50)\)

<table>
<thead>
<tr>
<th>p</th>
<th>Uniform demand, fixed length</th>
<th>Zipf demand, increasing length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fixed SF</td>
<td>Fixed (P_{100})</td>
</tr>
<tr>
<td>(10^{-3})</td>
<td>SF</td>
<td>SF</td>
</tr>
<tr>
<td>2625.0</td>
<td>1.05</td>
<td>2625.0</td>
</tr>
<tr>
<td>2677.3</td>
<td>1.05</td>
<td>2677.3</td>
</tr>
<tr>
<td>3228.7</td>
<td>1.25</td>
<td>3228.7</td>
</tr>
</tbody>
</table>

Table III. Cumulative retrieval \((M = 1000, k = 50)\)

<table>
<thead>
<tr>
<th>p</th>
<th>Uniform demand, fixed length</th>
<th>Zipf demand, increasing length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fixed SF</td>
<td>Fixed (P_{100})</td>
</tr>
<tr>
<td>(10^{-3})</td>
<td>SF</td>
<td>SF</td>
</tr>
<tr>
<td>2625.0</td>
<td>1.05</td>
<td>2625.0</td>
</tr>
<tr>
<td>2676.8</td>
<td>1.05</td>
<td>2676.8</td>
</tr>
<tr>
<td>3227.0</td>
<td>1.25</td>
<td>3227.0</td>
</tr>
</tbody>
</table>

In fact, there is an optimum SF value \(SF_{opt}\), which minimizes the average response time. Although the achieved minimum mean response times are higher when equally popular, fixed-length items are considered, the value where this happens, \(SF_{opt}\), is independent of the item retrieval method.

Increasing the FEC redundancy beyond \(SF_{opt}\), which varies with \(p\), has a steady adverse effect on the average response time. At these values, the parameter \(r_i\) in (12) tends to zero, and \(S_{opt}\) effectively increases linearly with the FEC stretch factor \(S_{opt}(k,n)\)

\[
S_{opt}(k,n) = \frac{1}{2} \left( \sum_{i=1}^{n} \sqrt{q_i \cdot SF \cdot f_i} \right)^2 = \frac{SF}{2} \left( \sum_{i=1}^{n} \sqrt{q_i \cdot f_i} \right)^2 \tag{13}
\]

In other words, the difference between the two techniques reduces with \(r_i\), which is their differentiation factor at lower FEC redundancy levels reflecting the advantage of the cumulative over the one-shot item retrieval.

![Figure 6. Cumulative (dashed line) retrieval vs one-shot retrieval (solid line) of items \(-M=1000, k=50\), fixed-length items with uniform demand distribution.](image)

![Figure 7. Cumulative (dashed line) retrieval vs one-shot retrieval (solid line) of items \(-M=1000, k=50\), variable-length items with Zipf demand distribution.](image)

![Figure 8. Fixed redundancy (solid line) versus fixed item success rate (dashed line) – one-shot item retrieval, \(M=1000, k = 50\), variable-length items with Zipf demand distribution.](image)
C. The impact of number of items, M, and FEC block size, k

The optimum SF value, though dependent on the packet error rate, remains the same as the number of items increases (see Table IV), whereas the minimum average response time increases. Longer broadcast schedules result in wider spacing of individual items in the schedule, hence higher response times for individual items.

Table V, on the other hand, indicates that increase of the FEC block size decreases the optimum response time and/or the $S_{F_{opt}}$. For the same stretch factor SF, larger FEC block sizes increase the correcting capability of FEC, as it has been shown in Fig. 5, and accelerate the retrieval of individual items. The response time and SF both approach their respective minimum points as k increases i.e., the benefits of large k diminish when k is very large.

<table>
<thead>
<tr>
<th>Table IV. Optimum Mean Response Times and FEC SF Values vs. Number of Items M (One-Shot Retrieval With Fixed Redundancy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>10^-7</td>
</tr>
<tr>
<td>10^-8</td>
</tr>
<tr>
<td>10^-9</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

In this paper we have performed a study of FEC-encoded data broadcasting. The joint use of packet level FEC and data carousels becomes highly relevant for multicasting in wireless environments, where a return path for user interactivity may either not exist or be cost-inefficient to use.

The average user response time, the key metric reflecting the responsiveness of data broadcasting systems, can be significantly reduced when data enjoy the additional protection of FEC at the expense of redundancy overhead that decreases the effective system capacity. Our analysis showed that for a given packet error rate, the FEC overhead may be tuned to minimize the average response time, whilst limiting the capacity wastage. This is a general trend that is independent of possible alternatives with respect to the way the application retrieves the data of interest or the way the FEC overhead per data item is determined.

The second interesting outcome is that the actual values of the minimum response time and the respective FEC settings are not affected by the way the item is retrieved. Although, in the absence of FEC, the cumulative item retrieval accelerates significantly the response time, when FEC is added to the broadcast items, its gain over the one-shot item retrieval technique fades out quickly and disappears when the average response time becomes minimum. Likewise, the optimum points are not sensitive to the way FEC is assigned to each data item. In fact, the mean response time is minimized when the FEC redundancy is just about enough to enable the full item retrieval upon their first appearance in the schedule.

More predictably, the optimum values of both FEC SF and response time are sensitive to the FEC block length, since higher block lengths improve the erasure correction capability of the considered block codes. The optimum response time is also sensitive to the number of items accommodated in the schedule, as longer schedules result in higher optimum values thereof.

In this work we limited ourselves to small systematic block codes. We are currently looking into ways to include other types of FEC codes in this study framework. The other direction for expanding this study is the relaxation of the assumption for uniform packet error rates. Bursts of lost packets may be experienced both in wireless environments, e.g., due to temporary loss of connectivity, but also in wired networks, e.g., due to router buffer management schemes that tend to discard packets in bursts. The analytical study of encoded data broadcasting under burst packet loss is the natural follow-up of this work.

REFERENCES