1

Reliable Wireless Video Streaming with Digital Fountain Codes

Raouf Hamzaoui*, Shakeel Ahmad**, Marwan Al-Akaidi*

* Faculty of Computing Sciences and Engineering, De Montfort University - UK

** Department of Computer and Information Science, University of Konstanz - Germany

Definition: Video streaming refers to a video transmission method that allows the receiver to view the video continuously after only a short delay.

Keywords: LT codes, Raptor codes, Wireless video streaming, 3GPP, Multimedia Broadcast/Multicast Service.

1.1 Introduction

Third generation cellular networks currently offer video streaming services to mobile users through unicast transmission in which a separate point-to-point connection with each recipient is established and maintained. However, due to limited server capacity and reduced available spectrum, the point-to-point approach does not scale well when the number of subscribers increases. Recently, the third Generation Partnership Project (3GPP) introduced the Multimedia Broadcast/Multicast Service (MBMS) [1], which allows efficient point-to-multipoint transmission of streaming video over the existing GPRS/EDGE and W-CDMA 3G networks. MBMS is an IP-based technology that uses forward error correction (FEC) with Raptor codes [2] at the application layer to protect the video bitstream against packet loss.

Raptor codes are a new class of erasure codes (that is, codes used against symbol erasure [3]) with many desirable features. Whereas traditional erasure codes have a fixed code rate that must be chosen before the encoding begins, Raptor codes are rateless as the encoder can generate on the fly as many encoded symbols as needed. This is an advantage when the channel conditions are unknown or varying because the use of a fixed channel code rate would lead to either bandwidth waste if the packet loss rate is overestimated or to poor performance if the packet loss rate is underestimated. Similarly, in broadcast and multicast systems where the same data is sent to many users over heterogeneous links, the choice of an appropriate fixed channel code rate is not obvious. Note that strategies based on retransmission of the lost packets would also be not appropriate for such applications because if too many receivers request retransmission of the data, the server will be overwhelmed. Another (and probably the most) attractive property of Raptor codes is their low encoding and decoding complexity as an encoded symbol is generated from k source symbols independently of the other encoded symbols in only $O(\log(1/\epsilon))$ time, and the k source symbols are recovered from $k(1+\epsilon)$ encoded symbols with high probability in $O(k\log(1/\epsilon))$ time for any positive number ϵ. This is a tremendous speed up over Reed-Solomon codes (the standard erasure codes), which typically have $O(k(n-k)q)$ encoding and decoding complexity if k source symbols are encoded into n codeword symbols for a symbol alphabet of size q, where

q should be larger than n.

Raptor codes are an extension and improvement to LT codes [4, 5, 6]. Both codes are known in the literature as digital fountain codes [7]. The goal of this chapter is to describe digital fountain codes and to explain how they are used in MBMS. The chapter is organized as follows. Section 1.2 introduces notations and terminology. Section 1.3 describes LT codes, while Section 1.4 discusses Raptor codes. Section 1.5 focuses on the MBMS video streaming framework. The last section overviews recent research on wireless video transmission with digital fountain codes.

1.2 Notations

In this section, we introduce the notations and terminology used in the chapter. Digital fountain codes are based on bipartite graphs, that is, graphs with two disjoint sets of vertices such that two vertices in the same set are not connected by an edge. The first set of vertices contains the source symbols, while the second set contains the encoded symbols. The symbols are binary vectors, and arithmetic on symbols is defined modulo 2. In particular, $\oplus$ denotes modulo 2 addition. If the number of source symbols is k, the degree of an encoded symbol is given by a degree distribution $\Omega(x) = \sum_{i=0}^{k} \Omega_i x^i$ on $\{1, \ldots, k\}$, where Ω_i is the probability that degree i is chosen. For example, suppose that

$$\Omega_i = \begin{cases} 0 & \text{if } i = 0; \\ \frac{1}{k} & \text{if } i = 1; \\ \frac{1}{i(i-1)} & \text{otherwise.} \end{cases}$$

Then $\Omega(x)$ is called the ideal soliton distribution [6]. A more practical distribution is the robust soliton distribution $\Delta(x)$ [6] given by $\Delta_i = \frac{\Omega_i + \Gamma_i}{d}$, where $\Omega(x)$ is an ideal soliton distribution, $\Gamma(x)$ is given by

$$\Gamma_i = \begin{cases} \frac{s}{ki} & \text{if } i = 1, \ldots, \frac{k}{s} - 1; \\ \frac{s}{k} \ln(\frac{s}{\delta}) & \text{if } i = \frac{k}{s}; \\ 0 & \text{otherwise,} \end{cases}$$

$d = \sum_{i=1}^{k} \Omega_i + \Gamma_i$, and $s = C \ln \frac{k}{\delta} \sqrt{k}$. Here C and δ are parameters (see [6] for an interpretation of these parameters).

Any distribution $\Omega(x)$ on $\{1, \ldots, k\}$ induces a distribution on F_2^k, the set of binary vectors of length k, by $Prob(v) = \frac{\Omega_{w(v)}}{\binom{k}{w(v)}}$, where $v \in F_2^k$ and $w(v)$ is the weight of v (that is, the number of nonzero components of v).

A reliable algorithm for a digital fountain code is an algorithm that ensures that all k source symbols can be recovered with probability $1 - \frac{1}{k^c}$ for a positive constant c.

1.3 LT-Codes

The LT encoder takes a set of k source symbols and generates a potentially infinite sequence of encoded symbols of the same alphabet. The symbol alphabet may consist of any set of l-bit symbols, for example, bits or bytes. Each encoded symbol is computed independently of the other encoded symbols. The LT decoder takes a little more than k encoded symbols and recovers all source symbols with probability $1 - \epsilon$. Here ϵ is a positive number, which gives the tradeoff between the loss recovery property of the code and the encoding and decoding complexity. Luby [6] proves that for the robust soliton distribution, an encoded symbol can be computed in $O(\log \frac{k}{\epsilon})$ time and all source symbols can be recovered from $k + O(\sqrt{k} \log^2(k/\epsilon))$ encoded symbols with probability $1 - \epsilon$ in $O(k \log(k/\epsilon))$ time. The following subsection gives the details of the encoding.

1.3.1 Encoding

Given k source symbols $s_1, \ldots, s_k$, and a suitable degree distribution $\Omega(x)$ on $\{1, \ldots, k\}$, a sequence of encoded symbols $c_i,\ i \geq 1, \ldots,$ is generated as follows. For each $i \geq 1$

1. Select randomly a degree $d_i \in \{1, \ldots, k\}$ according to the degree distribution $\Omega(x)$.
2. Select uniformly at random d_i distinct source symbols and set c_i equal to their modulo 2 bitwise sum.

Figure 1.1 illustrates the encoding procedure.

In the following subsection, we describe the decoding algorithm.

1.3.2 Decoding

We assume that a transmitted encoded symbol is either lost or received correctly. We also assume that the decoder can determine both the degree of a received encoded symbol and the source symbols connected to this encoded symbol. This can be done, for example, by using a pseudo-random generator with the same seed as the one used in the encoding. Suppose now that n encoded symbols have been received. Then the decoder proceeds as follows.

1. Find an encoded symbol c_i that is connected to only one source symbol s_j. If there is no such encoded symbol (that is, an encoded symbol with degree one), stop the decoding and wait until more encoded symbols are received before proceeding with the decoding.
   (a) Set $s_j = c_i$.
   (b) Set $c_x := c_x \oplus s_j$ for all indices $x \neq i$ such that c_x is connected to s_j.

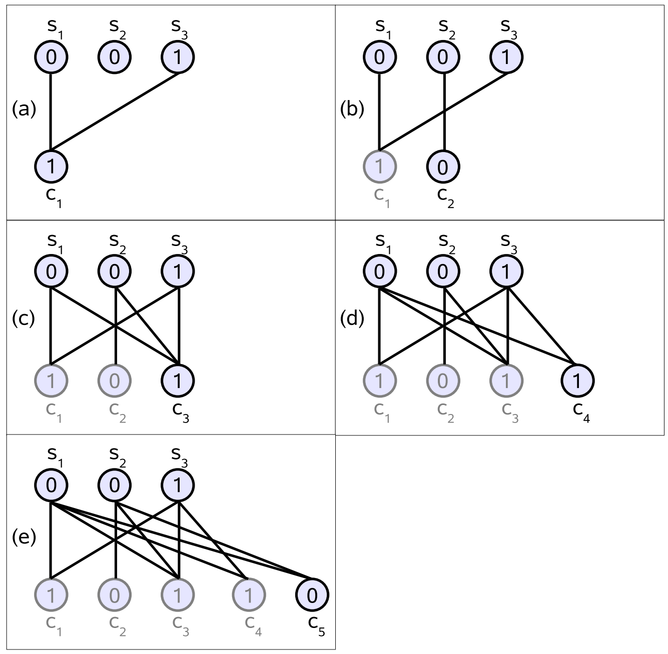


Figure 1.1: LT encoding of $k = 3$ source symbols (here bits) $s_1 = 0, s_2 = 0$, and $s_3 = 1$ to $n = 5$ encoded symbols $c_1, \ldots, c_5$. Each step shows a new encoded symbol. (a) $d_1 = 2$, $c_1 = s_1 \oplus s_3$. (b) $d_2 = 1$, $c_2 = s_2$. (c) $d_3 = 3$, $c_3 = s_1 \oplus s_2 \oplus s_3$. (d) $d_4 = 2$, $c_4 = s_1 \oplus s_3$. (e) $d_5 = 2$, $c_5 = s_1 \oplus s_2$.

(c) Remove all edges connected to s_j.

2. Repeat Step 1 until all k source symbols are recovered.

Figure 1.2 illustrates the decoding procedure. Figure 1.3 shows the performance of an LT code for the robust soliton distribution.

1.4 Raptor Codes

Raptor codes [2] are an extension of LT-codes that allow faster encoding and decoding. The complexity improvement is obtained by reducing the number of edges in the code graph. Since the number of edges in an LT code

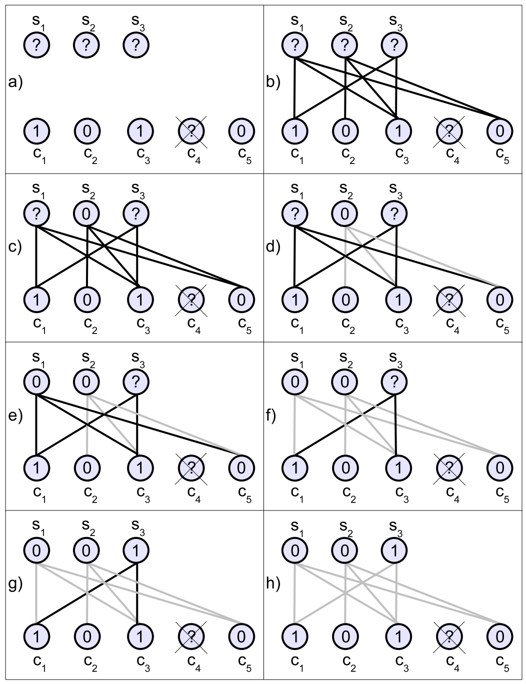
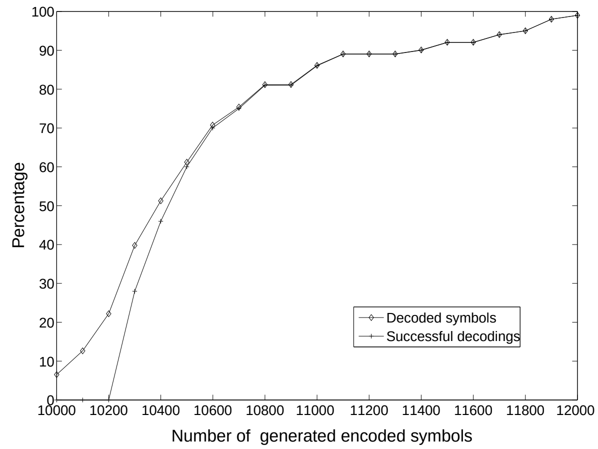
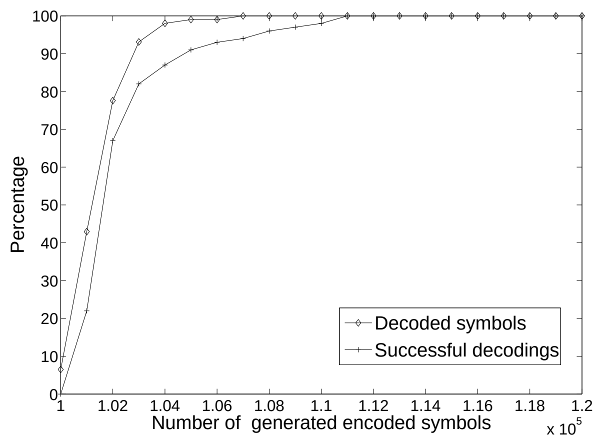


Figure 1.2: (a) An LT decoder trying to recover three source symbols (here bits): s_1, s_2, and s_3 from four received encoded symbols c_1, c_2, c_3, and c_5, while c_4 is lost. (b) Determine the degree and the source symbols associated to each received encoded symbol. (c) Find an encoded symbol with degree one. The only one here is c_2. Set $s_2 = c_2 = 0$. Since s_2 is also connected to c_3 and c_5, set $c_3 := c_3 \oplus s_2$ and $c_5 := c_5 \oplus s_2$. (d) Remove all edges connected to s_2. (e) The next encoded symbol with degree one is c_5, which is connected to s_1, so set $s_1 = c_5$. Since s_1 is also connected to c_1 and c_3, set $c_1 := c_1 \oplus s_1$ and $c_3 := c_3 \oplus s_1$. (f) Remove all edges connected to s_1. (g) There are two symbols with degree one: c_1 and c_3. Both are connected to s_3. Hence s_3 can be decoded with either of them. Set s_3 equal to either c_1 or c_3. (h) All source symbols have been recovered. The decoding stops.

is determined by the underlying distribution $\Omega(x)$, the idea is to choose a distribution that generates a small number of edges. However, by using such a distribution, the recovery performance of the code is worsened. In particular, it is shown in [2] that if an LT code with distribution $\Omega(x)$ on $\{1, \ldots, k\}$ is reliable, then its code graph must have at least $ck \log k$ edges,



(a)



(b)

Figure 1.3: Performance of an LT code for a robust soliton distribution with parameters $C = 0.1$ and $\delta = 0.5$. The number of source symbols is 10^4 in (a) and 10^5 in (b). The curve titled "Decoded Symbols" shows the average percentage of recovered source symbols as a function of the number of encoded symbols for 100 simulations. The curve titled "Successful decodings" shows the percentage of simulations where all source symbols were recovered.

where c is a positive number. To tackle this problem a traditional block code called precode is introduced before the LT code. In this way, a Raptor code consists of a concatenation of two codes: a precode and an LT code (Figure 1.4). More precisely, a Raptor code with parameters $(k, n, \mathcal{C}, \Omega(x))$ is the concatenation of a traditional erasure code $\mathcal{C}$ of dimension k and length n and an LT-code with distribution $\Omega(x)$ on n symbols. The precode considers all codeword symbols of the precode that were not successfully decoded by the LT code as erasures.

Examples. 1) An LT code with distribution $\Omega(x)$ on k source symbols can be seen as a $(k, k, F_2^k, \Omega(x))$ Raptor code.

2) Precode only Raptor codes. These Raptor codes have parameters $(k, n, \mathcal{C}, x)$.

3) Fast Raptor codes. These Raptor codes have parameters $(k, n, \mathcal{C}, \Omega_D(x))$ where $D = \lceil 4(1+\epsilon)/\epsilon \rceil$ and $\Omega_D(x) = \frac{1}{\mu+1}(\mu x + \sum_{i=2}^{D} \frac{x^i}{(i-1)i} + \frac{x^{D+1}}{D})$ with $\mu = \epsilon/2 + (\epsilon/2)^2$. One of the main results of [2] is that if $\mathcal{C}$ has code rate $R = \frac{1+\frac{\epsilon}{2}}{1+\epsilon}$ and $\mathcal{C}$ can be decoded on a binary erasure channel with erasure probability $(1-R)/2$ with $O(n \log \frac{1}{\epsilon})$ arithmetic operations, then the fast Raptor code can decode all k source symbols from $(1+\epsilon)k$ encoded symbols with high probability in $O(k \log(1/\epsilon))$ time. Examples of such precodes $\mathcal{C}$ are given in [2].

1.4.1 Systematic Raptor codes

The fountain codes described in the previous section do not provide systematic encoding, that is, if $s_1, \ldots, s_k$ are the source symbols and $c_1, \ldots, c_n$ are the encoded symbols, then there do not necessarily exist indices $i_1, \ldots, i_k$ such that $s_j = c_{i_j},\ j = 1, \ldots, k$ (the source symbols do not appear in the sequence of encoded symbols). This is a limitation because systematic encoding allows the decoder to immediately exploit any received symbol that

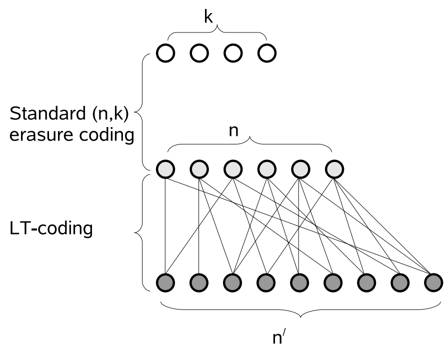


Figure 1.4: Raptor code.

corresponds to a source symbol. In the following subsection, we describe an algorithm [2] that provides systematic Raptor encoding.

Systematic encoding of Raptor codes

Let $(k, n, \mathcal{C}, \Omega(x))$ be a Raptor code. Let G be a generator matrix for $\mathcal{C}$. The encoding algorithm takes source symbols $x_1, x_2, \ldots, x_k$ and computes a set of k indices between 1 and $k(1+\epsilon)$ and a sequence of encoded symbols $z_1, z_2, \ldots$ satisfying $z_{i_1} = x_1, z_{i_2} = x_2, \ldots, z_{i_k} = x_k$. This is done as follows.

1. **Preprocessing Step**. Compute a matrix R and indices $i_1, \ldots, i_k$ in $\{1, \ldots, k(1+\epsilon)\}$ as follows.

   (a) Get $k(1+\epsilon)$ vectors $v_1, \ldots, v_{k(1+\epsilon)}$ in F_2^n by sampling $k(1+\epsilon)$ times independently from the distribution $\Omega(x)$ on F_2^n.

   (b) Compute a $k(1+\epsilon) \times n$ matrix S whose rows are the vectors $v_1, \ldots, v_{k(1+\epsilon)}$.

   (c) Compute the $k(1+\epsilon) \times k$ matrix $T = SG$. Use Gaussian elimination to find the rank of T. If the rank of T is less than k, output an eror message. Otherwise, find a submatrix R of T consisting of k rows $i_1, \ldots, i_k$.

2. Compute $y = (y_1, \ldots, y_k)$ with $y^T = R^{-1}x^T$, where $x = (x_1, \ldots, x_k)$.
3. Compute $u = (u_1, \ldots, u_n)$ with $u^T = Gy^T$.
4. Compute $z_i = v_i u^T,\ 1 \leq i \leq k(1+\epsilon)$.
5. Apply the LT code with distribution $\Omega(x)$ on u to generate the output symbols $z_i,\ i > k(1+\epsilon)$.

It can be shown [2] that the algorithm produces output symbols z_i such that $z_{i_1} = x_1, z_{i_2} = x_2, \ldots, z_{i_k} = x_k$.

Decoding of systematic Raptor codes

Given the received encoded symbols $u_1, \ldots, u_{k(1+\epsilon)}$, the algorithm recovers the original source symbols $x_1, x_2, \ldots, x_k$ as follows.

1. Use the decoding algorithm of the Raptor code to obtain $y_1, \ldots, y_k$.
2. Recover the input symbols $x_1, \ldots, x_k$ from $x^T = Ry^T$.

1.5 Video Streaming with MBMS

3GPP defines three functional layers for the delivery of MBMS-based services. The first layer, called Bearers, provides a mechanism to transport data over IP. Bearers is based on point-to-multipoint data transmission (MBMS bearers), which can also be used in conjunction with point-to-point transmission. The second layer is called delivery method and offers two modes of content delivery: download and streaming. Delivery also provides reliability with FEC. The third layer (User service/Application) enables applications to the end-user and allows him to activate or deactivate the service. An MBMS session consists of the following three phases.

1. User Service Discovery Phase. MBMS services are announced to the end-user using either 2-way point-to-point TCP-IP-based communication or 1-way point-to-multipoint UDP-IP-based transmission.

2. Delivery Phase. Multimedia contents are delivered (in either the streaming or the download mode) using 1-way point-to-multipoint UDP-IP-based transmission.

3. Post-Delivery Phase. A user may report on the quality of the received contents or request a file repair service (if in the download delivery mode) using 2-way point-to-point TCP-IP-based communication.

During the delivery phase, a UDP packet may be discarded by the physical layer if the bit errors cannot be corrected, and it can be lost due to, e.g., network congestion or hardware failure. Because there is no feedback channel in the delivery phase, ARQ-based protocols cannot be used. Instead, the media data is protected at the application layer using FEC with systematic Raptor codes.

The MBMS technical specifications recommend the H.264/AVC baseline profile as a video coder. The primary unit generated by the H.264/AVC codec is called the Network Abstraction Layer (NAL) unit. At the transport level, Real-time Transport Protocol (RTP), as specified by RFC1889 of the Internet Engineering Task Force is used. In general, one NAL unit is encapsulated in a single RTP packet according to the RTP payload specification [8]. However, one NAL unit may also be fragmented into a number of RTP packets or one RTP packet may contain more than one NAL unit. FEC is applied to the incomming stream of RTP packets.

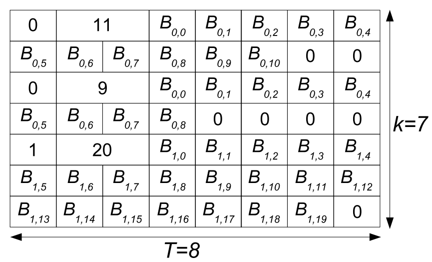


Figure 1.5: MBMS source block example. Three payloads of lengths 11, 9, and 20 bytes are placed in a source block of SBL $k = 7$ with symbol size $T = 8$ bytes. The first two payloads are from RTP flow $f = 0$, and the third one is from RTP flow $f = 1$. Each cell in the block is a byte. $B_{i,j}$ denotes the $(j + 1)th$ byte of the $(i + 1)th$ RTP flow.

A copy of each RTP packet is forwarded to the FEC encoder to construct a source block. A source block is a two-dimensional array of size $T \times k$, where the source block length (SBL) k is the number of symbols in the block, and T is the symbol size in bytes. To each incoming RTP packet, a 3-byte identifier is prepended, and the resulting block is inserted in the source block, starting from the first available empty row. The prepended identifier contains the RTP flow ID f and the length l of the RTP packet. The RTP flow ID f allows multiplexing several streams and protecting them together. If for an RTP packet, $l + 3$ is not a multiple of T, then the block must be padded with additional zeros. The padded zeros are not transmitted and can be inserted by the receiver to duplicate the original two-dimensional array. The source block is filled with incoming RTP packets until the number of source symbols reaches k. The value of k for a source block is flexible and computed dynamically during the source block construction. However, for any source block, the constraints $k \geq k_{\text{min}} = 1024$ (as the performance of Raptor codes is low for smaller k) and $k \leq k_{\text{max}} = 8192$ must be satisfied. Figure 1.5 shows an example of source block construction.

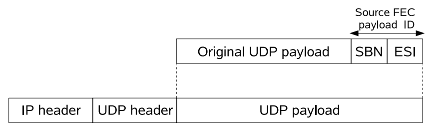


Figure 1.6: FEC source packet. SBN is a 16-bit integer that identifies the source block related to the RTP packet. ESI is a 16-bit integer that gives the index of the first source symbol in this packet.

Once a source block is completed, the FEC encoder generates $N - k$ repair (redundant) symbols, each of size T, by applying a systematic Raptor code on the k symbols of the source block. A pseudo-random number generator is used to generate the graph of the Raptor code. The pseudo-random number generator is based on a fixed set of 512 random numbers [1] that must be available to both sender and receivers. The value of N is not fixed and may vary with the source block.

Each symbol (source and repair) has two associated fields called source block number (SBN) and encoding symbol ID (ESI). The fields SBN and ESI, each of size two bytes, indicate the associated source block number and the position of the symbol within the block, respectively. ESI values of source symbols are in $\{0, \ldots, k-1\}$, while ESI values of repair symbols are larger than k. A source FEC payload ID is appended at the end of each RTP packet to create an FEC source packet, which is then encapsulated by UDP and sent to the receiver by the MBMS bearers (Figure 1.6). A number G of consecutive repair symbols are concatenated, and a repair FEC payload ID is prepended to the resulting block, yielding an FEC repair packet (Figure 1.7). Each FEC repair packet is encapsulated by UDP and sent to the recipients by the MBMS bearers.

At the receiver side, the received stream of source and repair packets is processed in blocks. If some of the source packets in a block are lost but sufficient repair packets from the same block are received, then the original

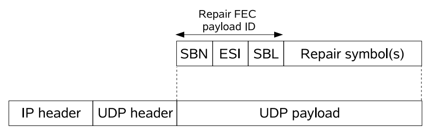


Figure 1.7: FEC repair packet. The packet contains G repair symbols of size T each. SBN is a 16-bit integer that identifies the source block related to the repair symbols. ESI is a 16-bit integer that gives the index of the first repair symbol in this packet. SBL is a 16-bit integer that gives the number of source symbols k in the source block.

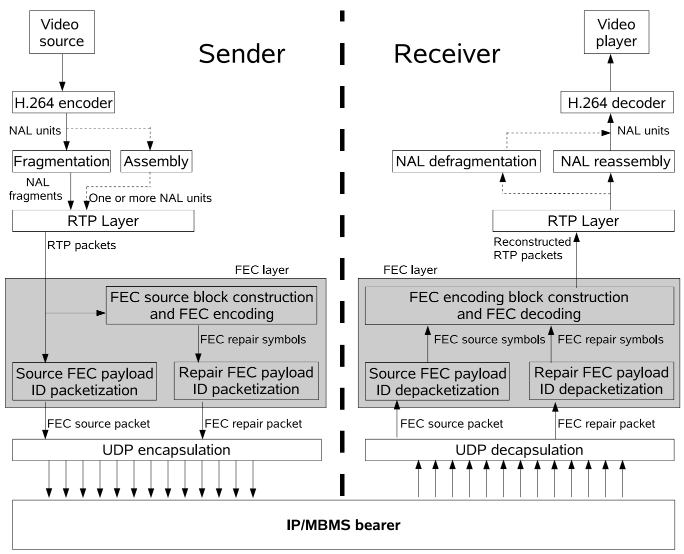


Figure 1.8: MBMS video streaming framework.

source block can be reconstructed by the Raptor decoder. The original RTP packets in individual streams can be recovered using f and l. These RTP packets are passed to the RTP layer, which extracts the NAL units and forwards them to the H.264 decoder.

Figure 1.8 illustrates the general framework of MBMS video streaming.

The MBMS specifications [1] provide recommendations for the values of T and G. These recommendations are based on the input parameters B

Table 1.1: MBMS recommendations for a maximum source block size B.

B	G	T
40KB	10	48
160KB	4	128
640KB	1	512

(maximum source block size in bytes), A (symbol alignment factor in bytes), P (the maximum repair packet payload size which is a multiple of A), $k_{\min}$, $k_{\max}$, and $G_{\max}$ (maximum number of repair symbols per repair packet), which should satisfy $\lceil(B/P)\rceil \leq k_{\max}$. The number of repair symbols per repair packet is estimated as $G = \min(\lceil(P \times k_{\min}/B)\rceil, P/A, G_{\max})$. The symbol size is estimated as $T = \lfloor(P/(A \times G))\rfloor \times A$. Table 1.1 shows some recommended settings for the input parameters $A = 4$, $k_{\min} = 1024$, $k_{\max} = 8192$, $G_{\max} = 10$, and $P = 512$.

1.6 Further reading

Apart from the 3GPP MBMS technical specifications [1] presented in this chapter, only a few works have considered the application of digital fountain codes to wireless video transmission. Afzal *et al.* [9] studied the effect of the MBMS video streaming parameters on the system performance. In particular, they suggest to determine the value of the source block length k by inserting as many RTP packets into the source block as possible subject to the constraint that the maximum end-to-end delay is not exceeded. Wagner, Chakareski, and Frossard [10] applied Raptor codes to efficiently stream scalable video data from multiple servers to a client. One limitation of the MBMS framework is that the information contained in packets that contain unrecoverable bit errors is ignored since these packets are discarded at the physical layer. Gasiba *et al.* [11] show that by modifying the receiver and the protocol stack, one can forward this information to the application layer and significantly improve the performance of MBMS video streaming at the cost of higher decoder complexity.

Links

1. Digital Fountain, Inc., `http://www.digitalfountain.com/`.
2. 3GPP Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs, `http://www.3gpp.org/ftp/Specs/html-info/26346.htm`.

References

[1] 3GPP TS 26.346 V6.6.0, "Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs", Oct. 2006.

[2] A. Shokrollahi, "Raptor Codes", IEEE Trans. Inf. Theory, Vol. 52, No. 6, June 2006, pp. 2551–2567.

[3] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", ACM Computer Communication Review, Vol. 27, no. 2, Apr. 1997, pp. 24–36.

[4] M. Luby, "Information Additive Code Generator and Decoder for Communication Systems", US Patent No. 6,307,487, Oct. 23, 2001.

[5] M. Luby, "Information Additive Code Generator and Decoder for Communication Systems", US Patent No. 6,373,406, Apr. 16, 2002.

[6] M. Luby, "LT-Codes", Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 271–280, 2002.

[7] J.W. Byers, M. Luby, and M. Mitzenmacher, "A Digital Fountain Approach to Asynchronous Reliable Multicast", IEEE Journal on Selected Areas in Communications, Vol. 20, N0. 8, Oct. 2002, pp. 1528–1540.

[8] S. Wenger, T. Stockhammer, M.M. Hannuksela, M. Westerlund, and D. Singer, "RTP payload format for H.264 video," RFC3984, IETF, Feb. 2005.

[9] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video Streaming over MBMS: A System Design Approach," Journal of Multimedia, Vol. 1, Issue 5, pp. 25–35, Aug. 2006.

[10] J. Wagner, J. Chakareski and P. Frossard, "Streaming of Scalable Video from Multiple Servers using Rateless Codes", Proc. IEEE International Conference on Multimedia and Expo 2006, Toronto, Ontario, July 2006.

[11] T. Gasiba, T. Stockhammer, J. Afzal, and W. Xu, "System Design and Advanced Receiver Techniques for MBMS Broadcast Services", Proc. IEEE International Conference on Communications 2006, Istanbul, Turkey, June 2006.