

Sliding-window Raptor codes for Efficient Scalable Wireless Video Broadcasting with Unequal Loss Protection

Pasquale Cataldi, *Member, IEEE*, Marco Grangetto, *Senior Member, IEEE*, Tammam Tillo, *Member, IEEE*, Enrico Magli, *Senior Member, IEEE*, Gabriella Olmo, *Senior Member, IEEE*

Abstract—Digital fountain codes have emerged as a low-complexity alternative to Reed-Solomon codes for erasure correction. The applications of these codes are relevant especially in the field of wireless video, where low encoding and decoding complexity is crucial.

In this paper we introduce a new class of digital fountain codes based on a sliding-window approach applied to Raptor codes. These codes have several properties useful for video applications, and provide better performance than classical digital fountains. Then, we propose an application of sliding-window Raptor codes to wireless video broadcasting using scalable video coding. The rates of the base and enhancement layers, as well as the number of coded packets generated for each layer, are optimized so as to yield the best possible expected quality at the receiver side, and providing unequal loss protection to the different layers according to their importance. The proposed system has been validated in a UMTS broadcast scenario, showing that it improves the end-to-end quality, and is robust towards fluctuations in the packet loss rate.

Index Terms—Application layer FEC codes, digital fountain codes, scalable video coding, H.264/SVC, MBMS video broadcasting.

I. INTRODUCTION

Multimedia streaming applications represent an emerging phenomenon. In a near future, users of the Internet are expected to be multimedia content producers, by publishing digital pictures, videos, home surveillance data; content mediators, by storing/forwarding streaming contents; content consumers of digital television, video on demand, mobile broadcasting and alike. Users will be interfaced to several content delivery networks: broadcasting (DVB-T, DVB-S/S2, DVB-C), bidirectional (xDSL, WiMAX), mobile (3G/4G, GERAN, UTRAN, DVB-H) and P2P logical overlays. Seamless transition from one network to another will be required. Finally, users will

access services using terminals with different capabilities in terms of computational power and screen resolution.

Such a complex scenario imposes several technological challenges. Scaling to a very large number of users may lead to scarcity in the aggregated bandwidth, and requires powerful multimedia compression tools. Several causes of unreliability, e.g. fading and shadowing on wireless links, congestion in the access segment or in the IP core network, churn induced instability in P2P overlays, demand for robust and error resilient encoding methods. The heterogeneity of user terminals requires clever strategies to avoid the simultaneous delivery of multiple versions of the same content.

In this context, scalable video coding (SVC) represents an innovative tool. The recent Annex G of H.264/AVC standard, known as H.264/SVC [1], provides layered temporal/spatial/quality scalability and any combination thereof. It yields significant gain with respect to transcoding or simulcasting of the individual layers, and limited compression inefficiency with respect to H.264/AVC.

Let us consider the situation where all coded video packets are subject to the same loss probability, and retransmission is unfeasible due to delay constraints or network flooding problems (such as in broadcast or multicast applications). In this situation, it is important that all the received packets can be exploited at the application layer. This is not the case of a layered source such as SVC, where the loss of base layer packets prevents one from exploiting the subsequent ones. A possible way to cope with this problem is using differentiated application-layer Forward Error Correction (FEC) codes, so as to guarantee the reception of the base layer with high probability. This provides graceful degradation, while keeping the overall coding redundancy limited; this principle is known as unequal loss protection (ULP). Popular algorithms reported in literature make use of Reed-Solomon (RS) codes to implement ULP. In [2] different RS codes are allocated to data segments of different importance. The code rates are obtained by means of an optimization procedure, aiming at maximizing the expected peak signal-to-noise ratio (PSNR) at the receiver side, given the packet loss rate and the source rate-distortion (RD) characteristics. An alternative algorithm has been proposed in [3], which implements a suboptimal yet more efficient optimization procedure. Differentiated RS codes have been applied to H.264 data partitions [4], and to H.264 slices exploiting their different impact on the data reconstruction [5]. Applications to SVC are reported in [6], [7], [8].

P. Cataldi is with Mobile Communications Department of Eurecom Institute, 2229 route des crêtes - F-06560 Sophia-Antipolis - France - E-mail: pasquale.cataldi@eurecom.fr

M. Grangetto is with Computer Science Department of Università di Torino, Corso Svizzera 185 - 10149 Torino - Italy - E-mail: marco.grangetto@unito.it

T. Tillo is with Electrical and Electronic Engineering of Xi'an Jiaotong-Liverpool University - 111 Ren Ai Road - Suzhou 215123 - P.R. China - E-mail: tammam.tillo@xjtlu.edu.cn

E. Magli and G. Olmo are with the Department of Electronics, Politecnico di Torino, Corso Duca degli Abruzzi 24 - 10129 Torino - Italy - E-mails: {enrico.magli|gabriella.olmo}@polito.it

This work was supported in part by National Natural Science Foundation of China (No. 60972085, No. 60903066).

RS codes have the drawback of high co-decoding complexity. The complexity is highly dependent on the encoding parameters and the packet loss rates [9]. In particular, the decoding speed necessary to sustain IPTV applications can be achieved only with the help of dedicated hardware implementation. Therefore, other classes of erasure codes have been recently proposed. Such codes trade co-decoding complexity with a certain degree of suboptimality, expressed in terms of an amount of extra symbols to be received in order to assure proper decoding.

Digital fountain (DF) codes exhibit the property that, letting K the length of the source block, the number of coded symbols N is not fixed *a priori*. Thus, they are also known as *rateless codes*. The first practical DF codes are the so-called Luby-transform (LT) codes [10]. Although they exhibit excellent efficiency, the co-decoding cost of LT codes is still too high to be afforded in practical applications. Raptor codes [11] exhibit linear co-decoding time while still keeping low coding overhead. Raptor codes have been included in the DVB-H standard for IPTV applications [12], and standardized by 3GPP (Third Generation Partnership Project) in the context of multimedia broadcast multicast services (MBMS) [13].

DF codes have been employed in the context of video broadcasting or media streaming in several papers. Most of them address equal loss protection (ELP). For example, in [14] a protocol for asynchronous video multicast is proposed, employing Tornado codes [15] and LT codes. In [16] a protocol for live unicast video streaming based on rateless codes is introduced. Similar ideas are described in [17], where the asymptotic behaviour of an asynchronous media streaming system based on DF codes is investigated. In [18] Raptor codes are used for streaming of stereoscopic video. Even though, in their basic realization, DF codes are unable to provide ULP, some recent work attempts to design DF codes that feature ULP. In [19] an algorithm is proposed to implement scalable data multicast to different user groups, assuming that the data are ordered according to their importance. Embedded windows of increasing length are defined and used with different probabilities, so that the most important symbols are included in a high percentage of parity checks, thus providing ULP. The same principle is also reported in [20], where an expanding window of source symbols is defined, so that the most important source symbols are included in the innermost window and can be decoded with higher probability.

In [21] the principle of Dependency-aware (DA)-UEP is proposed, where the existing data dependencies among I, P and B video frames are exploited to achieve differentiated protection, and different erasure protection codes are allocated to according to their impact on the reconstruction quality. This principle is applied in [22], where a practical scheme using DA-UEP is proposed. An application of Raptor codes to SVC is reported in [23], where the redundancy symbols of a given layer l are calculated including source symbols of all layers on which l depends, so achieving across-layer protection.

It is worth noticing that the performance of random codes such as DF codes is asymptotically optimal for large data blocks. On the other hand, large block sizes can seldom be afforded at the application layer and especially in video

applications. Moreover, large data blocks usually imply a significant latency, in that it is necessary to receive a larger number of coded symbols to ensure the decoding of a data block. If a receiver is unable to collect enough coded symbols within the maximum tolerable latency, the data block will not be decoded. In real time applications, or when TV-like experience is required (including fast channel change and tune-in), latency must be kept limited, and this implies using shorter data blocks. A common and sensible approach is to separately encode each group of pictures (GOP), i.e. group of successive frames that represents an independent coding unit.

A possible solution to this issue is the sliding-window approach, introduced in [24] for LT codes, where coding is performed not on disjoint data blocks, but instead on a sliding window of source symbols. This amounts to virtually enlarging the block size.

In this paper we build on [24] and provide a twofold contribution. First, we extend the concept of sliding-window to Raptor codes, so as to provide an encoding tool able to yield good performance with low coding overhead and limited co-decoding complexity. A performance analysis of the proposed sliding-window Raptor code in standalone mode is provided. Second, we show how it is possible to exploit the sliding-window Raptor code to provide ULP in the context of digital broadcasting of H.264/SVC video. ULP is addressed not in the code design such as e.g. in [20], but rather at the system level, as different digital fountains are employed for the different layers. We provide a framework for performance optimization of the broadcasting system, which selects the source encoding rates of the various layers, as well as the number of coded packets for each layer, so as to maximize a metric of expected quality at the receiver side.

This paper is organized as follows. In Sect. II sliding-window Raptor codes are introduced, and their performance is investigated. In Sect. III the novel coding tool is applied to broadcasting of H.264/SVC video using MBMS over a UMTS mobile networks. The optimization of the SVC and DF code rate allocation is addressed in Sect. III-C, and validation of the proposed application is provided in Sect. IV. Finally, in Sect. V conclusions are drawn.

II. SLIDING-WINDOW RAPTOR CODES

A. Background: DF codes

DF codes are random, sparse-graph codes developed for erasure channels. DF codes are also *universal*, because the coding and decoding efficiency is not affected by the cardinality of the source symbols. Let K be the length of the source block to be encoded. Then, the block can be decoded if at least $N = (1+\epsilon)K$ coded symbols are received, with ϵ representing the coding overhead. A key property of DF codes is that the number of coded symbols generated from K source symbols is not predetermined, but can be chosen arbitrarily. This is the reason why these codes are also called *rateless codes*.

LT codes represent the first practical implementation of the DF concept. Each coded symbol is obtained from a bitwise exclusive-OR of a uniformly random selection of d source symbols, where the degree d is specified by a suitable

statistical distribution. A good design of the degree distribution is crucial for the code performance. The *Robust Soliton Distribution* (RSD) $\mu(\cdot)$, defined in [10], is characterized by two parameters: c , a suitable positive constant [10], and δ , representing the decoding failure probability when $N = (1 + \epsilon)K$ coded symbols are received. By decreasing δ , the average degree of the coded symbols increases and so does the complexity. It can be shown that LT codes are asymptotically optimal, i.e. $\epsilon \rightarrow 0$ when $K \rightarrow \infty$ [10]. Nevertheless, LT codes exhibit more-than-linear encoding/decoding complexity.

Raptor codes [11] represent an evolution of LT codes. They exhibit lower complexity and at the same time excellent performance. Raptor codes encompass a pre-coding stage (usually a high rate low-density parity-check code - LDPC), concatenated with an inner LT code, working on symbols obtained by properly grouping l bits at a time. The coding overhead of Raptor codes is lower bounded by the overhead of the pre-code, and does not vanish asymptotically as in the case of LT codes. The decoding algorithm is composed of two steps. The inner LT decoder returns a hard bit-reliability vector. This latter is processed by the outer LDPC decoder, using the belief propagation algorithm [25]. This allows one to loosen the constraints on the LT code degree distribution, as the pre-code is able to recover the source symbols possibly not retrieved by LT decoding.

Notwithstanding their considerable performance and promising applications in multimedia, DF codes also exhibit some weaknesses. Because of their random nature, DF codes achieve near-optimal performance with large data blocks [10], [11]. Unfortunately, for short blocks, a random approach can be inefficient in terms of overhead required to successfully decode the original source.

B. The sliding-window principle

The concept of Sliding-Window Digital Fountain (SW-DF) was first proposed in [24] for LT codes. The main idea is to apply the DF code not on disjoint blocks of K source symbols, but instead on a sliding window of length K . The presence of memory between adjacent blocks allows to virtually extend the block length, as is shown later in this section.

The traditional (fixed window - FW) coding approach, depicted in Fig. 1(a), can be compared to SW-DF reported in Fig. 1(b). Each block of K source symbols is processed by a DF code, generating a proper number N of coded symbols. After the current block has been coded, the window is shifted S symbols forward. Then, the DF code processes the new block of K input symbols, S of which are new entries, whereas $K - S$ are overlapped with the previous window.

For the sake of simplicity, let us assume that K is a multiple of S . In this way each source symbol enters $N_o = K/S$ successive windows. In other words, if K is kept constant, as the window shift S is decreased, the source symbols enter more and more successive windows. We can define a *virtual block length* as $K_v = 2K - S$ symbols, i.e. the length of the symbol sub-stream where a given source symbol enters the encoding process. Let us clarify this concept with an example. Let us assume $K = 6$ and $S = 3$ and focus on a given source

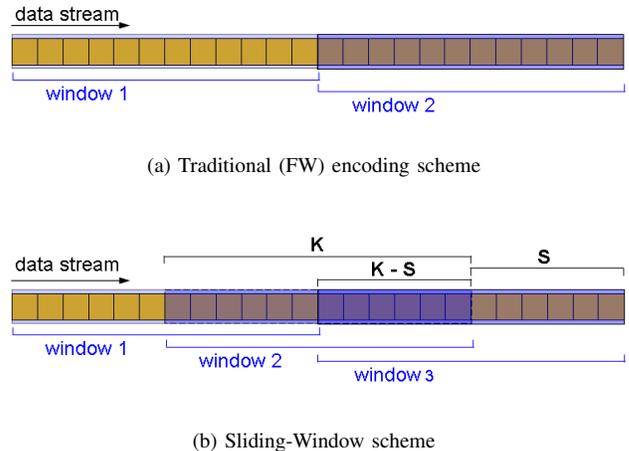


Fig. 1. Comparison between fixed window and sliding window schemes.

TABLE I
MAIN PARAMETERS OF SW-DF

Block length (symbols)	K
Window shift (symbols)	S
No. of windows a symbol belongs to	$N_o = K/S$
Virtual block length (symbols)	$K_v = 2K - S$
Encoded symbols per block (FW)	$N = (1 + \epsilon)K$
Encoded symbols per block (SW-DF)	$N_w = (1 + \epsilon) \cdot K^2 / (2K - S)$

symbol, e.g. symbol 13. Windows 1 to 6 span the following source symbols: (1–5), (4–8), (7–11), (10–14), (13–17). Symbol 13 is contained in $K/S = 2$ windows, i.e. 4 and 5. Hence, the interval in which symbol 13 can be picked up to form a coded symbol is (10-18), and the virtual block length is $K_v = 2K - S = 9$ symbols. We can notice that, by increasing the overlapping between adjacent windows, K_v increases tending to the limit value $2K$.

Let $N = (1 + \epsilon)K$ be the number of coded symbols output per each source block of length K in the FW scheme. If SW-DF is run with the same N , it will output a larger overall number of symbols, as it processes a larger number of length- K blocks due to the overlapping. In order to make SW-DF generate the same coded symbol rate as the FW scheme, a number of symbols per window smaller than N should be generated. In particular, the same number of coded symbols generated by FW for each block of length K , must be generated by SW-DF for each block of length K_v . Hence, the number of symbols generated by SW-DF for each window of length K is

$$N_w = N \cdot K / K_v = (1 + \epsilon) \cdot K^2 / (2K - S) \quad (1)$$

It is worth noticing that, if $S = K$ (i.e., no overlap), SW-DF reduces to the FW case. Moreover, number of blocks tending to infinity, the number of symbols generated becomes $N_w = (1 + \epsilon) \cdot S$. The main parameters that characterize SW-DF are summarized in Tab. I.

This principle has been applied to LT codes in [24] (SW-LT codes). The design of SW-LT codes is very similar to that of traditional LT codes, the main difference lying in the degree distribution, which takes into account the virtual data block

TABLE II
LDPC GENERATOR POLYNOMIALS.

ϵ_{LDPC}	$\lambda(x)$	$\rho(x)$
1.01%	$0.247354x^2 + 0.194184x^3 + 0.225899x^6 + 0.0340546x^7 + 0.101625x^{15} + 0.196883x^{16}$	x^{400}

extension. In fact, as a source data block has an actual span of K_v symbols as far as the generation of parity checks is concerned, it is appropriate to consider a degree distribution over a block length K_v . Although the degree distribution is designed for block length K_v , the symbols are picked only within the current window of size K . Hence, the computed degree distribution has to be re-scaled to have a maximum degree K . Details and performance of such codes can be found in [24].

C. Implementation of Sliding-Window Raptor Codes

As discussed in Sect. II-A, Raptor codes are the concatenation of a high performance binary block code, such as LDPC, and a weakened LT code. Hence, a sliding-window Raptor code (SW-Raptor) may be obtained replacing the LT code with the SW-LT scheme. However, some specific features of Raptor codes make this generalization not trivial.

Differently from LT codes, the degree distribution for Raptor codes does not depend on the source symbol block length [11]. Thus, in this situation it is not useful to modify the inner LT code degree distribution taking into account the virtual block length K_v . Instead, in this paper we propose a sliding-window approach that affects the precoding stage of Raptor codes. We impose that the block length of the pre-code be equal to the window shift S . The source is first encoded with a block code that spans S symbols. Then, the coded bits are grouped in l -bit symbols and input to SW-LT with degree distribution as in [11].

The SW-Raptor decoder performs the same operations in reverse order. The received symbols within each window of length K are LT-decoded. Due to the loosened degree distribution, not all symbols will be decoded. The external decoder is then used to recover from the residual erasures.

Such a decoding process can be implemented in different ways. The simplest approach, which has been followed in this paper, is to decode each data block using the SW-LT decoder, and then run the LDPC block decoder on those blocks whose symbols are not any longer used in subsequent parity check equations. It should be noted that, since the SW approach virtually enlarges the size of the block, the latency (i.e., the delay that the decoding process introduces in terms of source symbols) has to be at least K_v , as opposes to K symbols for the traditional scheme.

D. Performance of Sliding-Window Raptor Codes

In our simulations, we considered $4 \cdot 10^5$ samples of a memoryless binary source X with $P_0 = P(X = 0) = 0.5$. The source was encoded using a SW-Raptor with LDPC pre-code with fixed overhead ϵ_{LDPC} and generator polynomials shown in Tab. II.

TABLE III
LDPC BLOCK SIZES.

K	S	N_0	LDPC block
5000	5000	1	(4949, 5000)
	2500	2	(2474, 2500)
10000	10000	1	(9899, 10000)
	5000	2	(4949, 5000)
	2500	4	(2474, 2500)
40000	40000	1	(39599, 40000)
	20000	2	(19799, 20000)
	10000	4	(9899, 10000)
	5000	8	(4949, 5000)
	2500	16	(2474, 2500)

The performance results are expressed in terms of the undecoded symbol rate (USR), i.e. the percentage of undecoded symbols after Raptor decoding, and failed simulation rate (FSR), i.e. the number of simulations where the decoding process has not been successfully accomplished. Both metrics are expressed as a function of the encoding overhead ϵ in the range $[0, 10\%]$. Co-decoding complexity figures are provided as well. We present results for different block sizes, i.e. $K = 5000, 10000$ and 40000 bits, and different window shifts S . In particular, we have considered $N_o = K/S = 1, 2, 4, 8, 16$. The case $N_o = 1$ corresponds to the FW-Raptor code with the same LDPC and LT parameters. We recall that the LDPC block length is set equal to S in all simulations, as shown in Tab. III. The decoder is considered to have unlimited memory capability. All simulations have been performed on an Intel Pentium IV workstation at 3 GHz, running Linux Ubuntu with kernel version 2.6.15.

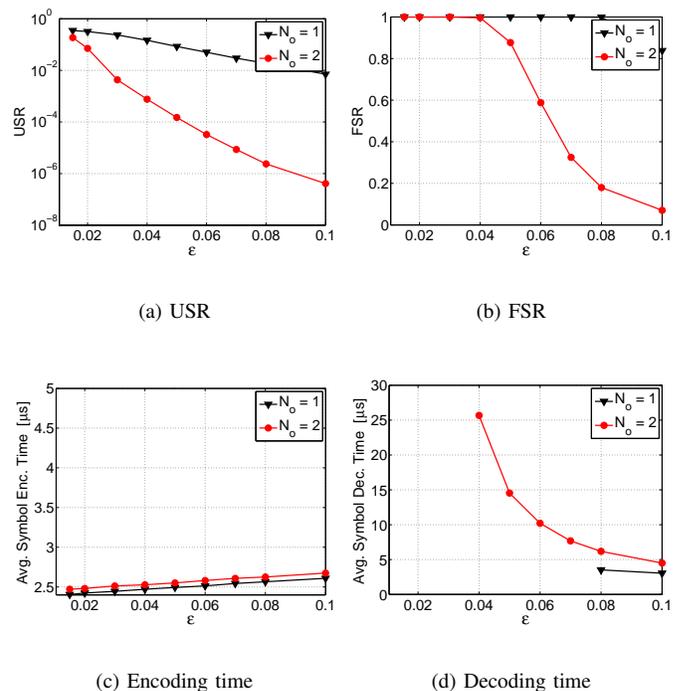


Fig. 2. Performance of SW-Raptor; $K = 5000$.

Fig. 2 shows USR and FSR versus ϵ in the case $K = 5000$, and $N_o = 1, 2$. We recall that $N_o = 1$ (triangle marker) refers

to FW-Raptor coding. It can be noticed that SW-Raptor code significantly outperforms FW coding in terms of USR. For example, for $\epsilon = 5\%$, SW-Raptor yields USR that is three orders of magnitude lower than FW. A significant performance improvement of SW-Raptor can be appreciated also in terms of the FSR (Fig. 2(b)).

In order to appreciate the SW-Raptor co-decoding complexity, Fig. 2(c) and Fig. 2(d) report the average encoding and decoding times per source symbol as functions of ϵ . It can be noticed that SW-Raptor exhibits a linear encoding complexity, with a negligible increase with respect to FW. The evaluation of the average decoding times only includes successful simulations; this is the reason why the FW curve is not plotted in the low ϵ region. We can observe that the decoding process of SW-Raptor is slower than FW. In fact, SW-Raptor computational complexity is dominated by the cost of keeping into account symbols, and the corresponding equations, which were only partially reduced during past windows. This amounts to managing a larger memory than the FW algorithm to store the undecoded symbols equations. Since the location of degree-one equations is not linear with the memory size, the decoding process turns out to be highly non linear, as well. In this paper all the undecoded symbols are stored in the extra memory so as to yield the best decoding performance, maximizing the probability to recover symbols from past windows. Nevertheless, one could decide to limit the amount of such memory, making the decoding process faster at the expenses of the decoding performance.

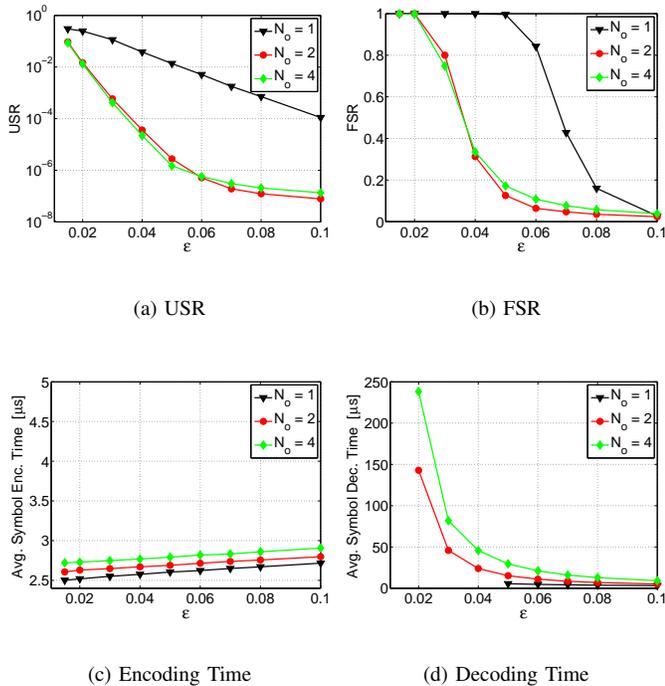


Fig. 3. Performance of SW-Raptor; $K = 10000$.

In Fig. 3 we present results for $K = 10000$ and different window shifts, i.e. $N_o = 1, 2, 4$. Again, it can be appreciated that SW-Raptor increases the reliability of the code when

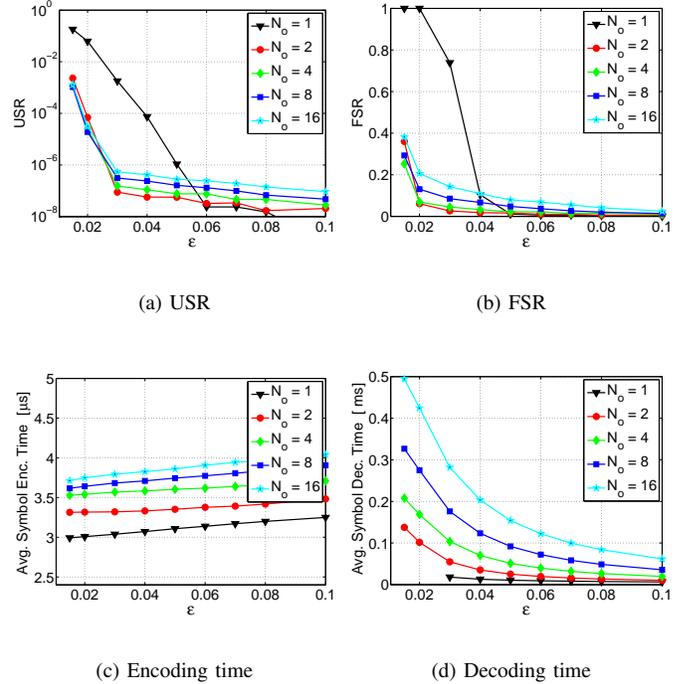


Fig. 4. Performance of SW-Raptor; $K = 40000$.

few excess symbols have been received, yielding significantly lower USR (Fig. 3(a)) and FSR (Fig. 3(b)) at parity of overhead ϵ . We can also appreciate that the performance of SW-Raptor with $N_o = 2$ and 4 are quite similar. This is due to the fact that in the first case we have $S = 5000$, whereas in the second case we have $S = 2500$. While increasing N_o improves the inner LT decoding, a reduced value of S impairs the external LDPC decoding performance. In the presented simulations the two effects compensate each other, thus yielding approximately the same overall performance. In particular, we can observe that when ϵ is small, the effect of the SW approach is dominant. However, when the number of encoded symbols is higher, the size of the LDPC block constrains the decoding performance.

The average co-decoding times are reported in Fig. 3(c) and 3(d) respectively. The encoding time, although being linear, increases with N_o . This is due to the fact that more windows are generated, while the total number of coded symbols remains fixed. Similarly, the complexity of the decoding process increases with N_o . Again, the complexity is mainly due to the memory management of the symbols belonging to the previous windows.

Fig. 4 reports performance results for SW-Raptor, $K = 40000$ and $N_o = 1, 2, 4, 8, 16$. In terms of the USR (Fig. 4(a)) and for redundancy values below 6%, SW-Raptor outperforms FW for any considered window shift. In this region, the performance is little dependent on the actual N_o value. On the other hand, for higher values of ϵ , FW tends to outperform SW-Raptor. This can be explained noticing that the performance of both FW and SW-Raptor is dependent on the LDPC pre-code block length. In fact, the larger is this parameter, the

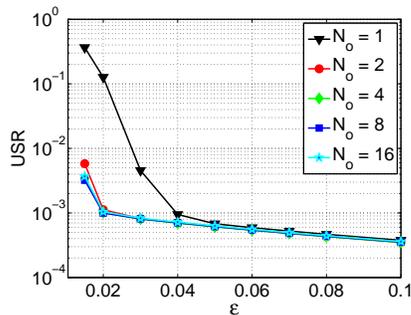


Fig. 5. Undecoded symbol rate after LT decoding; $K = 40000$.

more effective is the correction capability of the LDPC pre-code. When K is high, this effect is particularly relevant, as FW can exploit a very powerful pre-code with respect to SW-Raptor. However, this potential benefit is traded against the virtual block enlargement yielded by SW-Raptor, which turns out to be more effective in the low coding redundancy region. For the same reason, it is worth noticing that, among all the overlap strategies, the one that allows the longest pre-code block (i.e., $N_o = 2$) is to be preferred. Similar considerations can be drawn from Fig. 4(b) in terms of FSR.

The average co-decoding times for $K = 40000$ are presented in Fig. 4(c) and 4(d) respectively. Again, we can appreciate the fact that the encoding times are linear, and the complexity increases with N_o . The decoding times are approximatively linear for reasonable values of ϵ .

Finally, we analyze the impact of the LT code degree distribution on the overall SW code performance. Fig. 5 reports the percentage of undecoded symbols after the LT decoding stage versus ϵ , for $K = 40000$ and several values of N_o . As discussed in Sect. II-C, the design of SW-Raptor codes does not modify the LT degree distribution. However, we can notice that the SW scheme still achieves better performance than FW, and this improvement can be already appreciated for very low values of the overhead. This reveals that the overlapping strategy improves the performance independently of the adopted degree distribution. From the same figure we can also notice that it is not necessary to use a large overlap to significantly decrease the rate of non decoded symbols.

III. AN APPLICATION OF SW-RAPTOR TO SCALABLE VIDEO STREAMING

A. Streaming Model

In this section we propose an application of SW-Raptor codes to H.264/SVC video broadcasting. In order to profitably exploit the scalability provided by H.264/SVC, the different video layers are streamed via separate digital fountains, optimized in terms of the expected video quality offered to the end users. This scenario can be considered as an extension of the MBMS broadcasting service, which can be offered via the UMTS mobile network. Whereas the exploitation of SW-Raptor within MBMS would require to modify the erasure code adopted in the standard, the optimization strategy used to allocate the DF code rates can be employed within MBMS.

In the following it is assumed that the video server knows the number and the capabilities of the clients requiring service, so that it can classify them by their features (e.g. screen resolution, video quality). This information is usually exchanged between server and client when the multimedia session is set up. Let us consider a video source, encoded using H.264/SVC, and consisting of layers L_j , $j = 0, \dots, M$ with L_0 being the base layer. We define the i -th client class C_i , $i = 0, \dots, M$, as the set of clients that subscribe video layers from L_0 to L_i . The least demanding class C_0 is associated to the base layer L_0 , whereas class C_M demands for the maximum screen resolution by subscribing to all layers. Following [26], [27], we assume that the coded symbols are packetized using the RTP/UDP/IP protocol stack. Using the real-time transport control protocol (RTCP), it is possible to collect statistics about each RTP data flow. These can be processed by the server to estimate the expected packet loss rate PLR . In turn, this can be used to optimize the transmission.

We propose using SW-Raptor codes to achieve reliable transmission and ULP. To this end, and for each GOP, layers are encoded independently of each other using a SW-Raptor code with proper N . The coded symbols of a given layer are then grouped into packets of the same size. The objective is to maximize the expected PSNR at the receiver side, i.e. the average quality experienced by users of all classes for a given packet loss rate, while satisfying the application constraints. In particular, our goal is to identify the optimal trade off between the rate devoted to source coding and the redundancy devoted to SW-Raptor encoding of each layer, expressed in terms of the number of coded packets per layer per GOP.

At the decoder side, if the users of the i -th class are not able to correctly receive all the subscribed i layers, they can still resort to lower layers. These latter, being more protected, can be decoded with higher probability, providing graceful quality degradation. It has to be noted that, although the encoding is done independently for each layer, the optimization of encoding parameters is global across all layers. This allows one to perform ULP by assigning the proper degree of protection to each layer based on their relative importance. In fact, as the base layer reception is necessary to all user classes, it is clearly privileged in terms of protection.

B. Problem statement

We assume that the server is constrained to transmit no more than T packets per GOP, and that each packet contains L symbols from a single layer j so as to make the probabilities of decoding layers independent of each other. No prioritization among packets is considered, and they are subject to independent losses with probability PLR , which is assumed to be known.

The first variable of the optimization problem is the rate devoted to each layer by the SVC encoder. This is related to the quantization parameters (QP) assigned to the layers. We express this variable as $\mathbf{S} = \{S_0, S_1, \dots, S_M\}$, where S_j is the number of source data packets generated per GOP for layer j , $j = 0, \dots, M$.

The second variable to be set is the number of extra packets generated by the DF encoders for the reliable transmission of

each layer. To this end, let us define $\mathbf{N} = \{N_0, N_1, \dots, N_M\}$, with N_j , $j = 0, \dots, M$, being the number of coded packets generated for the j -th layer. These values should satisfy the constraint $\sum_{j=0}^M N_j = T$.

The next step is to define a quality metric as a function of variables \mathbf{S} and \mathbf{N} , and the packet loss rate PLR . We adopt PSNR as a measure of the end-to-end system performance. When H.264/SVC is used, there are several possible PSNR values involved, related to the number of properly decoded enhancement layers. Let us define $PSNR_i^j$ as the average PSNR experienced by users of class C_i when receiving layers up to L_j . It is worth noticing that, in case a user class is not able to receive all the subscribed layers, but, for example, it has to resort to a lower screen resolution, the decoded video will be properly upsampled in order to match the required resolution level. Hence, the PSNR will be evaluated between the original signal and this upsampled version. This is the reason why the dependency on the user class index is necessary. Moreover, $PSNR_i^{-1}$ refers to the case that class- i users are not able to receive even the base layer, so the reconstruction is devoted to error concealment only.

The expected PSNR experienced by users of class C_i can be evaluated as:

$$PSNR_i = \sum_{n=1}^i \left(\left((1 - P_n) \prod_{j=0}^{n-1} P_j \right) PSNR_i^{n-1} \right) + \left(\prod_{j=0}^i P_j \right) PSNR_i^i + (1 - P_0) PSNR_i^{-1} \quad (2)$$

where P_j is the probability of correctly decoding the j -th layer. It is worth noticing that both P_j and $PSNR_i^j$ are dependent on \mathbf{S} and \mathbf{N} . Eq. (2) contains three terms; the first one corresponds to the expected PSNR contribution when decoding fewer layers than subscribed. In particular, the term $(1 - P_n) \prod_{j=0}^{n-1} P_j$ represents the probability of losing layer n while correctly receiving all the underlying layers, and the summation term accounts for all the possible events of receiving a resolution layer lower than the requested one. The second term corresponds to the ideal case that the subscribed resolution is correctly received, i.e. exactly i layers are decoded. Finally, the last term models the event that not even the base layer is received, and error concealment has to be performed. This latter case is important because, even though in SVC it is often assumed that the base layer is always received correctly, this is not necessarily the case; therefore, error propagation due to the loss of the base layer should be taken into account.

At this point, we can define the expected end-to-end PSNR over a GOP as

$$\overline{PSNR} = \sum_{i=0}^M g_i \cdot PSNR_i \quad (3)$$

where g_i is the percentage of clients belonging to the i -th user class. The goal of our optimization procedure is then to maximize this end-to-end video quality metric by finding the rate allocation of the video layers (i.e. \mathbf{S}) and the amount of coded packets generated per each layer (\mathbf{N}) that maximize

(3) subject to the application constraints. In this paper, such constraint are taken as 1) the total bandwidth allocated to coded video packets must not exceed the available bandwidth, and 2) the base layer PSNR must exceed a minimum threshold to provide adequate quality. The general optimization problem can be cast as follows:

For each GOP

Find $PSNR_{opt} = \text{Max}_{\mathbf{S}, \mathbf{N}} \overline{PSNR}$

subject to the constraints on total rate and base layer quality.

The proposed optimization strategy clearly leads to ULP. In fact, the most important layers (e.g. the base layer) should be protected more as, if their decoding process fails, this impacts on the decoding of all subsequent layers. As more important layers yield a larger contribution to \overline{PSNR} , the optimization algorithm will allocate a larger number of packets to them.

It is worth noticing that in [28] an algorithm is proposed for joint optimization of the coding modes of SVC layers. In fact, in the original joint scalable video model (JSVM) software no joint optimization is performed, but instead the coding modes of each layer are locally optimized, starting from the lowest one and using user-specified QP values. The algorithm in [28] uses a weighted function to achieve several trade-off among the coding options. For example, it allows one to select a coding mode for the base layer that is not the best choice for the base layer itself, but achieves a significant gain in the enhancement layer coding. However, this approach is different from that addressed in this paper. Here, in fact, we regard the video encoder as a black box that can model any scalable video encoder, so we mainly focus on the trade-off between source and DF rate allocation. Moreover, as discussed, we want to privilege the base layer reception. Hence, our situation amounts to setting the weighting factor in [28] to 0, i.e. to the classical JSVM scheme, which yields highly efficient base layer coding.

C. Optimization Algorithm

Let us focus on the objective function in (3) and (2), and work out an expression for the several terms involved, emphasizing the dependency on the variables \mathbf{S} and \mathbf{N} .

Let us assume that exactly r packets out of T are received; the probability of this event is

$$P(r) = \binom{T}{r} PLR^{T-r} (1 - PLR)^r \quad (4)$$

Given the set of received packets, the decoder attempts to recover the original information for each layer. The probability P_j of decoding layer j , $j = 0, \dots, M$ depends on the actual number of received packets related to layer j , and on the erasure correction capabilities of the DF code employed for that layer. So, for instance, layer j will be discarded if it will not be possible for a non-systematic code to decode at least $(1 + \epsilon)S_j$ packets. Let N_j and n_j be respectively the number of transmitted and received packets for layer j , and $\mathbf{N} = \{N_j\}$, $j = 0, \dots, M$ and $\mathbf{n} = \{n_j\}$, $j = 0, \dots, M$, with $r = \sum_{j=0}^M n_j$. We can write that

$$P(j|r, \mathbf{n}, \mathbf{S}, \mathbf{N}) = P(n_j \geq (1 + \epsilon)S_j | r, \mathbf{S}, \mathbf{N})$$

and at this point the probability of decoding layer j can be evaluated as:

$$P_j = P(j|r, \mathbf{n}, \mathbf{S}, \mathbf{N})P(\mathbf{n}|r, \mathbf{N})P(r)$$

In this paper we have used the experimental curves reported in Sect. II-D to derive $P(n_j \geq (1 + \epsilon)S_j | r, \mathbf{S}, \mathbf{N}) = 1 - \text{FSR}((n_j - S_j)/S_j)$, where $\text{FSR}(\epsilon)$ is estimated by simulation.

The probability of receiving packet distribution \mathbf{n} from the transmitted set \mathbf{N} , for each layer L_0, L_1, \dots, L_M , can be written as

$$P(\mathbf{n}|r, \mathbf{N}) = \frac{\prod_{j=0}^M \binom{N_j}{n_j}}{\binom{T}{r}} \quad (5)$$

Once the values of P_j are defined, the values of $PSNR_i^j$ can be evaluated as a function of \mathbf{S} , i.e. of the rate allocation performed by the SVC encoder.

Putting all the elements together, we can finally work out an expression for the objective function \overline{PSNR} versus \mathbf{S} and \mathbf{N} . In principle, we should evaluate this function trying all possible combinations of SVC rate allocations \mathbf{S} (related to the QP parameter for each layer) and DF code allocation, subject to the application constraints, and select the configuration that maximizes the objective function. In Sect. IV we present results obtained with a full search approach, which is a viable solution as we have used a single resolution enhancement layer. In presence of a larger set of quality, resolution and temporal layers one should develop a more sophisticated optimization procedure.

IV. PERFORMANCE EVALUATION

We have simulated the transmission of H.264/AVC video over a UMTS link, assuming that 256 kbps are available at the application layer. All the generated packets have the same size. As the performance is expected to vary with packet size, we have considered several values of this parameter, ranging from 64 to 1024 bytes. Each packet contains coded symbols belonging to a single layer. All packets are subject to the same average loss rate, and the packet loss process is modeled via a binomial distribution.

Each GOP of each layer is separately encoded using a SW-Raptor code with LDPC pre-code as in Sect. III, using binary symbols, window size $K = 10000$ symbols, $N_o = 1$ (FW) and $N_o = 2$ (SW-Raptor). We did not perform simulations for larger overlaps because, as noted in Sect. II-D, the performance of SW-Raptor codes does not significantly improve by increasing the overlap.

We present results for three video sequences, namely *Coastguard*, *Foreman* and *News*, with CIF resolution, frame rate 30 fps and GOP size of 16 frames. We consider two different client classes. Class C_0 can access video in QCIF resolution, whereas class C_1 can handle CIF resolution. The H.264/SVC encoder implements two levels of spatial scalability, with a QCIF base layer L_0 and a CIF enhancement layer L_1 . Clients of the C_0 class will only subscribe the H.264/SVC base layer,

whereas C_1 clients will subscribe both L_0 and L_1 to achieve the full screen resolution video.

A simple error concealment strategy is adopted. When the base layer L_0 of one GOP has not been received, we copy the last decoded frame of the previous GOP and use it to replace the missing base layer, until a new GOP is received.

In order to reduce the optimization parameters space, a set of constraints on the SVC layers allocation have been imposed. In particular, we imposed $PSNR_0^0 \geq 30$ dB, i.e. we fixed a minimum acceptable quality for the base layer at QCIF resolution, and $PSNR_1^1 > PSNR_1^0$, i.e. decoding of layer L_1 must improve the image quality with respect to the up-sampling from QCIF to CIF. The optimization amounts to seeking the QPs of the base and enhancement layer (QP_0, QP_1) that maximize \overline{PSNR} in (3), given the previous constraints and the allowed transmission rate.

A. Results for video optimization

In this section we present simulation results and, in particular, we analyze the impact on the expected end-to-end PSNR of 5 system parameters, i.e. QP_0, QP_1 , the packet payload size L , the average channel conditions PLR and the ratio g_1 of clients belonging to class C_1 .

Tab. IV, V, VI show some optimization results for the three video sequences, assuming $PLR = 0.05$ and $g_1 = 0.4$. Each table includes the number of packets needed to transmit uncoded video (S_0 and S_1), the number of coded packets for each layer (N_0 and N_1), the corresponding overheads per layer $\epsilon_j = (N_j - S_j)/S_j$ and the achieved $PSNR_{opt}$ using SW as a function of QP_0 and QP_1 . The optimal configuration in terms of QP_0, QP_1, N_0 and N_1 is the one yielding the largest $PSNR_{opt}$. The optimal value of $PSNR_{opt}$ is shown in boldface.

For comparison, in all the tables we report also the optimization results when using the classical FW approach with the same parameters as SW. In particular, we report in brackets the packet allocation of the FW scheme only when it is different from the SW one, and the achieved $PSNR_{opt}$. From Tab. IV, V, and VI, we can observe that the optimal packet allocation for SW and FW is almost the same, but the $PSNR_{opt}$ achieved by FW is larger. From the previous Tables one can notice the interplay among the optimization parameters: increasing QP_0 one reduces S_0 and leaves more room to the enhancement layer S_1 . The optimal rate devoted to error correction, i.e. N_0, N_1 , is then driven by the proposed optimization metric and depends on PLR and g_1 . Finally, from Table VI we note that only in the case of the *Coastguard* sequence the rate budget of 256 Kbps is not enough to guarantee a minimum PSNR of 30 dB. Clearly, this is due to the reduced compression efficiency yielded by SVC on that particular video sequence.

Tab. VII shows the dependency of the optimization process on the packet payload size for the *Foreman* sequence. We consider robust compression of the RTP/UDP/IP header as in [26], [29], [30]. We can notice that $PSNR_{opt}$ has a maximum, indicating that there is an optimal packet size for the transmission. This result can be explained as follows.

TABLE IV

VIDEO OPTIMIZATION AS A FUNCTION OF QP FOR SW-RAPTOR CODES; $T = 66$, $PLR = 0.05$, $L = 256$ BYTES, $g_1 = 0.40$, *News* SEQUENCE.

QP_0	QP_1	S_0	N_0	ϵ_0	S_1	N_1	ϵ_1	$PNSR_{opt}$ (FW)	$PNSR_{opt}$ (SW)
25	42 (44)	49	58 (60)	0.184 (0.224)	7 (5)	8 (6)	0.143 (0.200)	36.06	36.37
26	40 (41)	45	54 (55)	0.200 (0.222)	10 (8)	12 (11)	0.200 (0.375)	36.15	36.38
27	38 (39)	40	48 (50)	0.200 (0.250)	16 (12)	18 (16)	0.125 (0.333)	35.72	36.18
28	38	36	45	0.250	16	21	0.312	35.88	35.93
29	37	33	41	0.242	20	25	0.250	35.60	35.64
30	36	29	36	0.241	25	30	0.200	35.06	35.24
31	36	27	34	0.259	26	32	0.231	34.76	34.82

TABLE V

VIDEO OPTIMIZATION AS A FUNCTION OF QP FOR SW-RAPTOR CODES; $T = 66$, $PLR = 0.05$, $L = 256$ BYTES, $g_1 = 0.40$, *Foreman* SEQUENCE.

QP_0	QP_1	S_0	N_0	ϵ_0	S_1	N_1	ϵ_1	$PNSR_{opt}$ (FW)	$PNSR_{opt}$ (SW)
29	48 (49)	53	62	0.170	4	4	0.000	31.84	32.76
30	42 (43)	46	56 (57)	0.217 (0.239)	8 (7)	10 (9)	0.250 (0.286)	32.88	33.04
31	40 (41)	41	51 (52)	0.244 (0.268)	13 (10)	15 (14)	0.154 (0.400)	32.66	32.81
32	39	36	45	0.250	17	21	0.235	32.51	32.55
33	38 (39)	31	38 (42)	0.226 (0.355)	24 (18)	28 (24)	0.167 (0.333)	31.88	32.13

TABLE VI

VIDEO OPTIMIZATION AS A FUNCTION OF QP FOR SW-RAPTOR CODES; $T = 66$, $PLR = 0.05$, $L = 256$ BYTES, $g_1 = 0.40$, *Coastguard* SEQUENCE.

QP_0	QP_1	S_0	N_0	ϵ_0	S_1	N_1	ϵ_1	$PNSR_{opt}$ (FW)	$PNSR_{opt}$ (SW)
32	42	42	51	0.214	13	15	0.154	30.13	30.25
33	41	34	43	0.265	19	23	0.211	29.87	29.90
34	41	29	39	0.345	20	27	0.350	29.52	29.52
35	40	24	31	0.292	29	35	0.207	29.20	29.24

TABLE VII

VIDEO OPTIMIZATION AS A FUNCTION OF L FOR SW-RAPTOR CODES; $T = 66$, $PLR = 0.05$, BYTES, $g_1 = 0.40$, *Foreman* SEQUENCE.

L	QP_0	QP_1	S_0	N_0	ϵ_0	S_1	N_1	ϵ_1	$PNSR_{opt}$ (FW)	$PNSR_{opt}$ (SW)
1024	30 (31)	49 (43)	12 (11)	15 (14)	0.250 (0.273)	1 (2)	1 (2)	0.000	32.10	32.48
512	30	44 (45)	23	29 (30)	0.261 (0.304)	3	4 (3)	0.333 (0.000)	32.63	32.83
256	30	42 (43)	46	56 (57)	0.217 (0.239)	8 (7)	10 (9)	0.250 (0.286)	32.75	32.91
128	30	41 (42)	92	110 (112)	0.196 (0.217)	20 (16)	23 (21)	0.150 (0.312)	32.75	32.90
64	30	41	184	220	0.196	39	46	0.179	32.68	32.70

On one hand, when the packet is large, the overhead of the packet header is negligible; however, because of the small number of packets, the allocation algorithm has little room for optimization, and the resulting video quality is not high. By reducing the payload size it is possible to fine-tune the system parameters by better distributing protection between the two layers. After a certain payload size, the packet header overhead tends to dominate the performance, using a lot of the available bandwidth. The optimal packet length L , in this case, is 256 bytes.

In Tab. VIII the optimized parameters are reported as a function of PLR for the *News* sequence. As can be expected, when the network conditions get worse, more protection is needed. Moreover, in order to yield smooth video quality, more encoded packets will be devoted to the protection of the base layer, at the expenses of the enhancement layer. As a consequence, the optimal QP_1 tends to increase, so decreasing the best achievable quality. However, the PSNR impairment for PLR ranging from 0.01 to 0.3 is only about 3 dB, showing that the proposed system is effective at providing graceful quality degradation under harsh transmission conditions.

Finally, in Tab. IX the optimized parameters are reported as a function of the percentage g_1 of clients belonging to class C_1 .

B. Performance evaluation at the client side

In realistic network applications, even if the average packet loss rate is known, the actual loss rate between the server and a given client can be different. In this section we validate the impact of a mismatch in the packet loss rate on the proposed optimization procedure. In particular, we perform the optimization using a given value of PLR , but we assume that a receiver experiences an actual packet loss rate $PLR_l \neq PLR$.

For each video sequence, we consider one of the optimal configurations worked out in the previous section, and analyze the performance at the receiver side as a function of PLR_l . We show the results for classes C_0 and C_1 , using both SW and FW-Raptor codes. We assume $PLR = 0.05$, $L = 256$ bytes and $g_1 = 0.40$.

In Fig. 6 and 7 we show the impact of PLR_l on the optimized system for the *Coastguard* sequence. In particular, Fig. 6 shows the performance related to clients belonging to class C_1 , while Fig. 7 depicts the performance for class C_0 . Lines with square markers in Fig. 6(a) represent the probability of correctly decoding both layers. Lines with triangle markers refer to the probability of decoding the base layer only; the resulting quality will be lower because the video will be upsampled to meet the requirements of the C_1 class. Finally, lines with circle markers show the probability not to even

TABLE VIII
VIDEO OPTIMIZATION AS A FUNCTION OF PLR FOR SW-RAPTOR CODES; $T = 66$, $L = 256$ BYTES, $g_1 = 0.40$, *News* SEQUENCE.

PLR	QP_0	QP_1	S_0	N_0	ϵ_0	S_1	N_1	ϵ_1	$PNSR_{opt}$ (FW)	$PNSR_{opt}$ (SW)
0.01	25	40 (41)	49	55 (56)	0.122 (0.143)	10 (8)	11 (10)	0.100 (0.250)	36.68	36.93
0.05	26	40 (41)	45	54 (55)	0.200 (0.222)	10 (8)	12 (11)	0.200 (0.375)	36.15	36.38
0.10	26 (27)	42 (41)	45 (40)	58 (54)	0.289 (0.350)	7 (9)	8 (12)	0.143 (0.333)	35.62	35.90
0.15	27	42 (43)	40	56 (57)	0.400 (0.425)	7 (6)	10 (9)	0.429 (0.500)	35.22	35.43
0.20	28	42 (43)	36	55 (57)	0.528 (0.583)	7 (6)	11 (9)	0.571 (0.500)	34.58	34.91
0.25	28 (29)	44 (43)	36 (33)	59 (56)	0.639 (0.697)	5 (6)	7 (10)	0.400 (0.667)	33.98	34.41
0.30	28 (29)	49 (50)	36 (33)	63	0.750 (0.909)	3	3	0.000	33.63	34.14

TABLE IX
VIDEO OPTIMIZATION AS A FUNCTION OF g_1 FOR SW-RAPTOR CODES; $T = 66$, $PLR = 0.05$, $L = 256$ BYTES, *News* SEQUENCE.

g_1	QP_0	QP_1	S_0	N_0	ϵ_0	S_1	N_1	ϵ_1	$PNSR_{opt}$ (FW)	$PNSR_{opt}$ (SW)
0.10	25	44	49	60	0.224	5	6	0.200	39.32	39.38
0.30	25	43 (44)	49	59 (60)	0.204 (0.224)	6 (5)	7 (6)	0.167 (0.200)	37.20	37.36
0.50	26 (27)	40 (39)	45 (40)	53 (50)	0.178 (0.250)	10 (12)	13 (16)	0.300 (0.333)	35.32	35.55
0.70	29	37	33	41	0.242	20	25	0.250	34.26	34.31
0.90	30 (32)	36	29 (24)	36 (32)	0.241 (0.333)	25 (27)	30 (34)	0.200 (0.259)	33.37	33.65

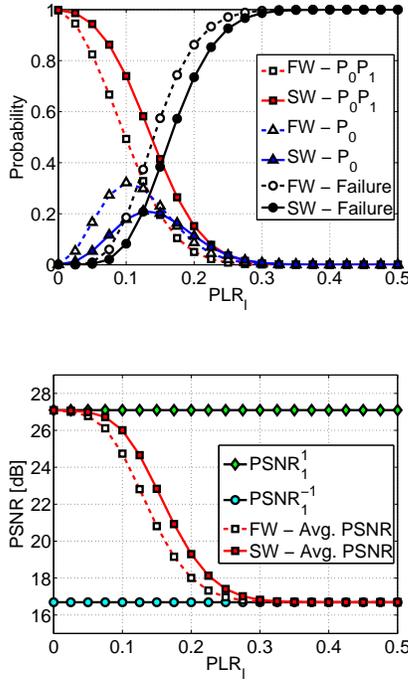


Fig. 6. Performance analysis at the client side for the *Coastguard* sequence. 6(a): probability that the video be decoded at a given quality by clients belonging to C_1 . 6(b): expected PSNR as a function of PLR_1 .

decode the base layer, i.e. the system failure probability. The expected PSNR is plotted as a function of PLR_1 in Fig. 6(b). In both figures, solid lines refer to SW-Raptor coding, whereas dotted lines refer to FW. In the two cases, and for this video sequence, the optimization process yields the same coded packet allocation. This means that the comparison between the two systems is actually a comparison between the FW and SW strategies.

We can notice that, when PLR_1 is less than PLR , the expected PSNR is high but limited to the target quality level that has been selected considering the average network conditions.

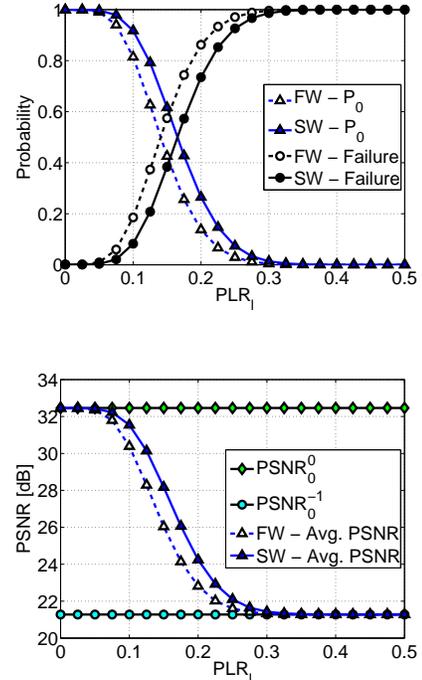


Fig. 7. Performance analysis at the client side for the *Coastguard* sequence. 7(a): probability of video decoding for C_0 clients. 7(b): expected PSNR as a function of PLR_1 .

On the other hand, when the link conditions are worse than PLR , the SVC approach provides graceful degradation (Fig. 6(b)). Moreover, the system using SW-Raptor outperforms the FW-Raptor code in terms of decoding probability. For example, when $PLR_1 = 2PLR = 0.1$, SW-Raptor yields a probability of successfully decoding the enhancement layer of about 0.74. In the same conditions, FW only achieves 0.49. In general, Fig. 6(b) reveals that SW-Raptor yields superior expected PSNR than FW. Since this figure represents the average quality of all clients belonging to C_1 , the difference between SW and FW Raptor codes is small. However, it must

be pointed that the video resolution for a single client can be either high (both layers decoded) or low (only the base layer decoded).

In Fig. 7 the system performance is reported for clients belonging to C_0 . Although part of the bit-rate is devoted to encoding a layer that is not useful to such clients, also this class of users achieves good visual quality and robustness. Indeed, even for $PLR_l = 0.10$, the probability of correct decoding exceeds 91%. Also in this case, the SW-Raptor scheme outperforms FW (81%).

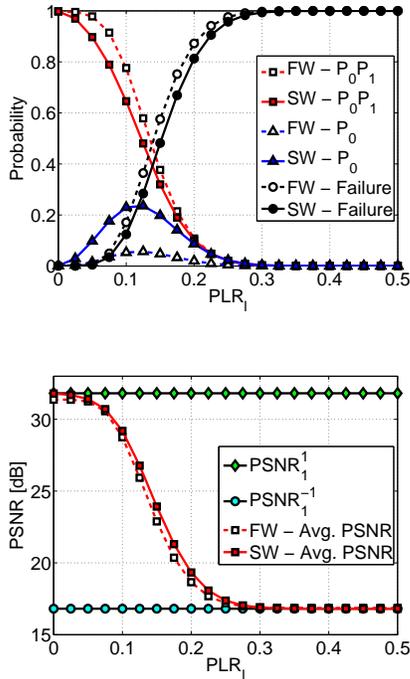


Fig. 8. Performance analysis at the client side for the *News* sequence. 8(a): probability that the video be decoded at a given quality by clients belonging to C_1 . 8(b): expected PSNR as a function of PLR_l .

In Fig. 8 and Fig. 9 the same analysis on the impact of a mismatched PLR_l is shown in the case of the *News* sequence. In this case the optimization reported in Tab. IV leads to different SVC layers and packet allocations for SW and FW codes, respectively. In particular, using SW the optimal allocation yields the same QP_0 but a lower value for QP_1 , i.e. more source packets are allocated to L_1 . This extra rate, coupled with the improved error correction performance given by SW codes, explains the gain in terms of the average decoded quality showed in Fig. 8(b) (the achievable PSNR is 31.81 dB using SW and 31.39 dB using FW).

In Fig. 9 we can appreciate the performance in terms of decoding probability and PSNR for the users in the class C_0 . In this case the source/channel coding allocation is the same for both SW and FW codes and the gain of the SW approach is due to the better error correction capability.

Finally, the effectiveness of our approach is confirmed in Fig. 10, where we compared the performance of using FW or SW, given the same packet allocation on the *News* sequence. In particular, the optimal allocation for FW has been used also

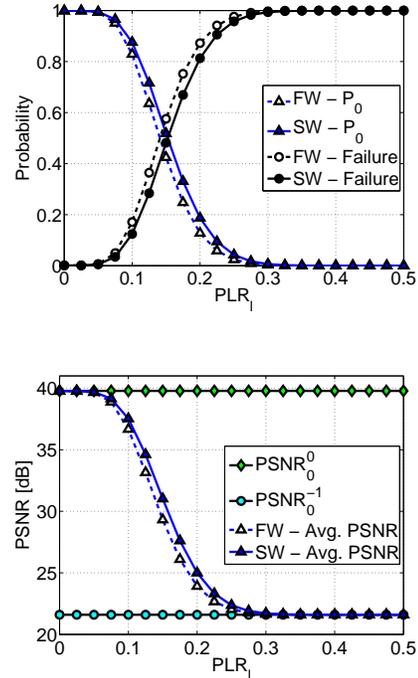


Fig. 9. Performance analysis at the client side for the *News* sequence. 9(a): probability of video decoding for C_0 clients. 9(b): expected PSNR as a function of PLR_l .

in the SW case to show the improvement yielded by the coding strategy.

We can conclude that SW-Raptor codes always provide a performance improvement with respect to the classical, fixed window approach. Moreover, the proposed algorithm for optimal encoding and coded packet rate allocation achieves satisfactory video protection even when the link conditions unexpectedly worsen.

V. CONCLUSION

In this paper we proposed a new class of DF codes, called SW-Raptor codes, which allow one to virtually enlarge the source block length by overlapping different data windows. We provided simulation results related to the standalone performance of SW-Raptor codes, showing that they outperform state-of-the-art DF codes, yielding very small overhead.

Then, we proposed SW-Raptor codes for unequal loss protection of scalable video, in the context of MBMS digital broadcasting over UMTS mobile networks. We defined an optimization procedure to select the rates to be allocated to the various layers and the number of coded packets per layer, so as to optimize an expected end-to-end quality metric. As the number of encoded packets is different for each layer, the scheme provides ULP and graceful quality degradation. The experimental results show that the proposed encoding scheme achieves very low decoding failure probability, also when the actual packet loss rate is different from the nominal value. Moreover, SW-Raptor codes outperform the equivalent scheme employing classical Raptor codes.

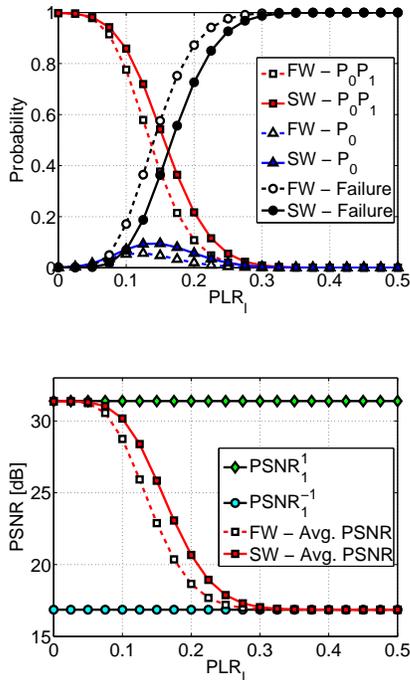


Fig. 10. Performance analysis at the client side for the *News* sequence. Fig. 10(a): probability that the video be decoded at a given quality by clients belonging to C_1 . 10(b): expected PSNR as a function of PLR_1 . In this analysis, the system using SW adopts the packet allocation corresponding to the greatest $PSNR_{opt}(FW)$.

Future research study will focus on the development of sliding-window systematic raptor codes, extending those adopted in the MBMS standard, and on the identification of practical and efficient optimization strategies in the presence of several enhancement layers.

REFERENCES

- [1] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, September 2007.
- [2] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on Selected Areas in Commun.*, vol. 18, no. 6, pp. 819–828, Jun 2000.
- [3] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," in *Proc. of 33rd Asilomar Conf. on signals, systems and computers*, Oct 1999, vol. 1, pp. 342–346.
- [4] Xing-Jun Zhang, Xiao-Hong Peng, R. Haywood, T. Porter, "Robust video transmission over lossy network by exploiting H.264/AVC data partitioning," in *Proc. BROADNETS 2008 - 5th International Conference on Broadband Communications, Networks and Systems*, 2008, pp. 307 – 314.
- [5] E. Baccaglioni, T. Tillo, and G. Olmo, "Slice sorting for unequal loss protection of video streams," *IEEE Signal Proc. Letters*, vol. 15, pp. 581 – 584, 2008.
- [6] S. Yingbo, W. Chengke, and D. Jianchao, "A Novel Unequal Loss Protection Approach for Scalable Video Streaming over Wireless Networks," *IEEE Trans. on Consumer Electronics*, vol. 53, no. 2, pp. 363 – 368, May 2007.
- [7] Yunqiang Liu and Songyu Yu, "Adaptive unequal loss protection for scalable video streaming over IP network," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 4, pp. 1277 – 1282, May 2007.
- [8] Hojin Ha and Changhoon Yim, "Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 736 – 744, May 2008.
- [9] M. Luby, T. Stockhammer, and M. Watson, "Application layer FEC in IPTV services," *IEEE Communications Magazine*, vol. 46, no. 5, pp. 736 – 744, May 2008.
- [10] M. Luby, "LT Codes," in *Symposium on Foundations of Computer Science*. IEEE, November 2002, pp. 271–280.
- [11] A. Shokrollahi, "Raptor Codes," *Transaction on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.
- [12] DVB-H, "Url: <http://www.dvb-h.org> (January 2010)," .
- [13] 3GPP, "Url: <http://www.3gpp.org> (January 2010)," .
- [14] 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, no. 8, October 2002.
- [15] J. W. Byers, M. Luby and M. Mitzenmacher, "Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads," in *INFOCOM*, New York, NY, March 1999, IEEE, pp. 275–283.
- [16] S. Ahmad, R. Hamzaoui, M. Al-Akaidi, "Robust Live Unicast Video Streaming with Rateless Codes," in *Proc. of Packet Video Conference*, 2007.
- [17] H. Jenkac and T. Stockhammer, "Asynchronous Media Streaming over Wireless Broadcast Channels," in *Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME)*, July 2005, pp. 1318–1321.
- [18] A. Serdar Tan, A. Aksay, C. Bilen, G. Bozdagi Akar, E. Arikan, "Rate-Distortion Optimized Layered Stereoscopic Video Streaming with Raptor Codes," in *Proc. of Packet Video Conference*, 2007.
- [19] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Fagoonce, Z. Xiong, "Scalable Data Multicast Using Expanding Fountain Codes," in *Proc. 45th Annual Allerton Conference on Communications, Control and Computing*, Monticello, USA, September 2007.
- [20] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, Z. Xiong, K. Vermeirsch, Y. Dhondt, "Expanding Window Fountain codes for scalable video multicast," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME 2008)*, June 2008, pp. 77 – 80.
- [21] A. Bouabdallah, J. Lacan, "Dependency-Aware Unequal Erasure Protection Code," in *Proc. of PV, 15th Int. Packet Video Workshop*, April 2006.
- [22] A. Bouabdallah, J. Lacan, M. Diaz, "Adaptive Dependency-Aware Unequal Erasure Protection Code (available at dmi.ensica.fr/img/pdf/bouabdallahlacandiaz_soelden.pdf)," .
- [23] C. Hellge and T. Schierl and T. Wiegand, "Receiver driven layered multicast with layer-aware forward error correction," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, October 2008, pp. 2304 – 2307.
- [24] M. Bogino, P. Cataldi, M. Grangetto, E. Magli, G. Olmo, "Sliding-Window Digital Fountain Codes for Streaming of Multimedia Applications," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, New Orleans, USA, May 2007.
- [25] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, second edition, December 2003.
- [26] ETSI 3GPP, "3GPP TR 126 937 v7.0.0 Release 7," Tech. Rep., June 2007.
- [27] M. Luby, T. Gasiba, T. Stockhammer, M. Watson, "Reliable Multimedia Download Delivery in Cellular Broadcast Networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, March 2007.
- [28] H. Schwartz, T. Wiegand, "Further results for an RD-optimized multi-loop SVC encoder," in *Document JVT-W071*, April 2007.
- [29] K. Svanbro, "RFC 3409: Lower Layer Guidelines for Robust RTP/UDP/IP Header Compression," Tech. Rep., December 2002.
- [30] P. Seeling, M. Reisslein, F. Fitzek, S. Hendrata, "Video Quality Evaluation for Wireless Transmission with Robust Header Compression," in *IEEE ICIS-PCM*, Singapore, December 2003.