Cross-Packet Coding for Delay-Constrained Streaming Applications

Giuseppe Cocco, *Member, IEEE,* and Dario Floreano, *Senior Member, IEEE*

***Abstract*—Delay-constrained streaming from a moving platform is of paramount importance for applications such as the remote control of drones, where a low-delay video stream is required in order to provide visual feedback to the pilot while providing high quality video at playback. In this letter we propose a scheme based on cross-packet coding for delay-constrained streaming over block fading channels with channel state information at the receiver only. The proposed scheme largely enhances upon a memoryless transmission approach in terms of average decoded rate and provides increased protection to packets transmitted earlier within a block, which is useful for successively compressed sources such as IPPPP video streams. The proposed scheme approaches the asymptotic upper bound over a wide range of SNR already for blocks with size of practical relevance and has a comparatively low complexity.**

I. Introduction

Real-time streaming is of paramount importance in the context of the upcoming *Tactile Internet* [1]. In particular, the video stream from remotely controlled robotic systems, such as unmanned aerial vehicles (UAV), is used as feedback to the pilot in beyond visual line of sight (BVLOS) operations, so that keeping a low delay is of paramount importance. At the same time, good video quality at playback is required in many applications such as advertisement or in the film industry. In order to reduce the end-to-end latency an IPPPP video encoder can be used [1]. In such encoder each video frame is compressed exploiting its correlation with the previous frame only [2]. Transmission over the channel can take place before the whole GOP is generated at the source encoder, since knowledge of future frames is not required by the source encoder. Due to the sequential compression, it is desirable that frames transmitted first have higher reliability with respect to those transmitted afterwards, in that all frames within a GOP successive to a lost frame cannot be used. In modern video compression standards, such as H.264, such strict interdependence is relaxed, in that a given P frame can be encoded using as reference one or more past frames, not necessarily the preceding one [3]. However, also in this case an increased protection of frames transmitted earlier in the GOP is a desirable feature due to the memory of the video encoder.

The last years have witness an increasing effort in the research related to streaming applications over unreliable channels. In [4] a source streaming messages each with a different deadline has been studied in terms of diversity-multiplexing tradeoff (DMT). In [5] a delay-constrained streaming transmission is considered, in which all messages have to be decoded by the same deadline. In both works the information relative to a given message is included in all transmissions from the moment the message is made available to the transmitter up to the decoding deadline. In coding theory such approach is known as *cross-packet coding* and has been applied in the context of hybrid automatic repeat request (HARQ) to limit the waste of bandwidth in case a packet is lost in the channel and transmission of additional information is required. In [6] a turbo code-based scheme is proposed. In such scheme the first k-bits message is encoded with rate k/n and transmitted. If the receiver can not decode it, informs the transmitter through a feedback channel. The transmitter turbo-encodes the first packet together with the second using a rate $2k/(k+n)$, punctures the systematic bits relative to the first one and transmits the remaining n bits. A joint decoding of both messages is then attempted by the receiver. In [7] low-density parity check codes (LDPC) are used. In [8] a turbo code is considered in an HARQ context with outdated channel state information at the transmitter (CSIT). In [9] the problem is studied from an information theoretical perspective considering HARQ with incremental redundancy. All such works focus on an HARQ scenario and assume feedback at each time slot. Furthermore, among these works, the ones using actual channel codes consider at most 3 messages per block. Recently [10] streaming codes have been proposed for transmission over erasure channels. Such codes are based on the sliding window model [11] and are optimal in such context.

In this letter we propose a novel scheme based on cross-packet coding for delay-constrained streaming applications over block fading channels. The proposed solution is based on LDPC codes. Unlike in [7], no puncturing is used, an arbitrary number of packets is considered and no feedback is assumed. Furthermore, in the proposed scheme the size of the generator matrices grows more slowly with the block size with respect to [7] and half the number of *super-iterations* is sufficient to fully harvest the benefits of the joint encoding (JE), which results in a scheme with comparatively low complexity. We show that the performance in terms of frame error rate (FER) and average decoded rate (ADR) improves with the number

Giuseppe Cocco was with the LTS4 Signal Processing Laboratory and with the Laboratory of Intelligent Systems, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He is now with the Information Theory and Coding group, Universitat Pompeu Fabra (UPF), Barcelona, Spain, Email: giuseppe.cocco@upf.edu

Giuseppe Cocco was partly founded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Individual Fellowship grant agreement No 751062

Dario Floreano is with the Laboratory of Intelligent Systems, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, Email: dario.floreano@epfl.ch

[1] In the present letter we use IPPPP to indicate the fact that video frames are sequentially encoded, but make no assumptions on the group of pictures (GOP) size.

of packets that are jointly encoded. We also show that the reliability of the different messages depends on their position within the encoded macro-block, with earlier messages being more protected. This is a desirable feature for systems in which sequentially compressed data, such as videos, are transmitted. Unlike in [10] we consider a block fading channel and propose an approach that can be implemented using different LDPC constructions, rather than focusing on a specific code design. Up to our knowledge, this is the first time that such aspects are systematically studied in the context of cross-packet coding for streaming over block fading channel and that a coding and decoding scheme show the practicality of a joint encoding approach in such setup.

II. System Model

Let us consider a wireless point-to-point communication system. Time is slotted, with slot duration equal to the channel coherence time. A block fading channel between the transmitter and the receiver is considered and only channel state information at the receiver (CSIR) is assumed. The transmitter is made available a new k-bits message $\mathbf{u}_t$ at the beginning of time slot t, $t = 1, \ldots, M$. In order to decrease the end-to-end latency the transmitter does not wait for the whole block of M messages to be available, but rather encodes and transmits them as they are generated. The received signal in time slot t is

$$\mathbf{y}_t = h_t \mathbf{x}_t + \mathbf{w}_t, \tag{1}$$

where h_t is the fading coefficient which changes in an i.i.d. fashion at each time slot, $\mathbf{x}_t$ is a $1 \times n$ vector representing the transmitted signal while $\mathbf{w}_t$ is a vector of i.i.d. random variable which accounts for the noise, each component of which is distributed according to $\mathcal{N}\left(0, \sigma^2\right)$. Once the whole macro-block of M codewords [2] is received, the receiver tries to decode it jointly. Although in the present letter we restrict ourselves to the case in which the decoding starts after the whole macro-block is received, the same approach introduced in the following can be used to attempt decoding every time a new codeword is available. This can help to meet the low-delay requirements for remote control of UAVs. Combining the two decoding approaches may result in a system which can meet the stringent delay constraints of UAV steering while providing a high quality video at playback [3].

With reference to the video streaming setup mentioned in the Introduction, a macro-block of M codewords may contain a GOP. Note that M refers to the number of channel codewords that is required (for a given message size k) to transport a whole GOP. However, the number of video frames in the GOP can be different from (most likely smaller than) M. In fact, a frame can be divided into multiple packets of k bits before transmission in order to meet the constraints of the physical layer, which is a common practice in video streaming over a wireless medium. This might occur more frequently for I frames, which usually are larger than P frames. Note that transmitting a frame using several channel codewords still preserves the increased error protection of frames transmitted earlier within a GOP.

The setup we consider is different from the one studied in [10], where a sliding window approach is used so that each message is to be decoded by a different deadline and an erasure channel is considered. Unlike other previous works on cross-packet coding, no feedback is assumed from the receiver. Without loss of generality, a binary phase-shift keying (BPSK) signalling is assumed for ease of exposition. Higher order modulations can be used requiring only modifications to the modulator/demodulator. We assume that $Mk \leq n$ (if a modulation of order $|\mathcal{M}|$ is used, the condition becomes $Mk/\log_2|\mathcal{M}| \leq n$).

The details of how messages are encoded, transmitted and decoded are given in the next section.

III. Proposed Scheme

The overall scheme is depicted in Fig. 1.

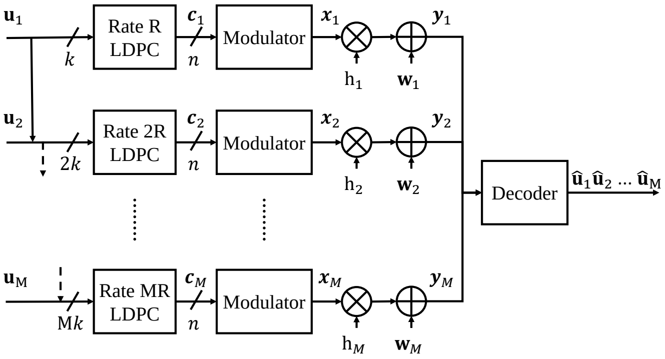

Figure 1. System overview. The input of a given LDPC encoder is the set of all available messages.

A. Encoder

At time $t = 1$ the transmitter is made available message $\mathbf{u}_1$. This is encoded using an LDPC code with rate $R_1 = R = k/n$, thus generating the codeword $\mathbf{c}_1 = \mathbf{u}_1 \mathbf{G}_1$, $\mathbf{G}_1$ being the $k \times n$ generator matrix. The codeword $\mathbf{c}_1$ is mapped to the BPSK sequence $\mathbf{x}_1$ using the mapping $0 \rightarrow -1, 1 \rightarrow +1$ and transmitted over the channel within a time slot. At the beginning of time slot $t = 2$, the transmitter is provided a new message $\mathbf{u}_2$. Rather than encoding it separately as in the case of the first message, the new message is concatenated with the previous one, thus forming the new message $\mathbf{u}_2' = [\mathbf{u}_1 \mathbf{u}_2]$. Message $\mathbf{u}_2'$ is then encoded using a $(2k) \times n$ generator matrix $\mathbf{G}_2$. The new codeword $\mathbf{c}_2$ is mapped into the sequence $\mathbf{x}_2$ and sent over the channel. In general, at the beginning of time slot t, the transmitter has available t messages $\mathbf{u}_1, \ldots, \mathbf{u}_t$. These are concatenated into the message $\mathbf{u}_t' = [\mathbf{u}_1, \ldots, \mathbf{u}_t]$ and encoded with the $(tk) \times n$ generator matrix $\mathbf{G}_t$, generating codeword $\mathbf{c}_t$ with rate $R_t = tk/n = tR$. Note that the rate in each channel block increases linearly with the channel block index, while the overall rate remains equal to R. Note also

[2] Note that the number of messages that are transmitted in a macro-block and the number of codewords in a macro-block are both equal to M.

[3] By "playback" we mean that a decoding delay equal to the duration of a GOP is allowed.

that the proposed approach uses smaller generator matrices with respect to [7]. In [6] and [7] the generated codeword in the second slot is longer than the one in the first, due to the concatenation of the two available packets and the fact that the same amount of redundancy is transmitted in both slots. The length of the signal vector sent through the channel is kept constant in both slots by puncturing (in the second one) the systematic bits corresponding to $\mathbf{u}_1$. In [7] the size of the generator matrix in the first slot is $k \times n$, while in the second slot it is $(2k) \times (n+k)$. Although the case of a generic number of transmissions is not considered in [7], extending such approach to the considered setup would lead to a matrix of size $(tk) \times (n + (t-1)k)$ in slot t, which is larger than in the proposed scheme.

B. Decoder

The decoding takes place in two steps: a forward super-iteration and a backward super-iteration. Let us consider the first. At time block t the receiver observes the signal $\mathbf{y}_t$, described by Eqn. 1. The decoder first applies maximum ration combining (MRC) to each of the M messages. This can be done since, unlike in [6] [7], the systematic bits are not punctured. Let us denote with $\mathbf{y}_t[i:j]$ the subset of elements of vector $\mathbf{y}_t$ in positions $i, i+i, \ldots, j-1, j$. Let us also assume that the systematic bits are located at the beginning of the packet, while the parity bits are to be found at the end, i.e.: $\mathbf{c}_t = [\mathbf{u}_1 \ldots \mathbf{u}_t \mathbf{s}_t]$, $\mathbf{s}_t$ being a $1 \times (n - tk)$ vector containing the parity of packet t. In order to decode message $\mathbf{u}_1$ the first k symbols of each of the M packets are combined as follows:

$$\mathbf{z}_1[1:k] = \sum_{m=1}^{M} h_m \mathbf{y}_m[1:k], \tag{2}$$

i.e., the first k elements of vector $\mathbf{z}_1$ are the combination of all replicas of $\mathbf{u}_1$ (h_m should be replaced by its conjugate h_m^* if a complex modulation is considered) [4]. The last $n-k$ elements of $\mathbf{z}_1$ are set to the non-systematic part of $\mathbf{y}_1$, that is:

$$\mathbf{z}_1[k+1:n] = \mathbf{y}_1[k+1:n]. \tag{3}$$

The decoder calculates the log-likelihood ratio vector $\mathbf{L}_1$ for the first packet starting from $\mathbf{z}_1$ and attempts decoding using belief propagation for a given number of iterations. The *a-posteriori* probabilities thus obtained for message $\mathbf{u}_1$ are fed to the second decoder and used as priors for the first k symbols. Symbols from $k+1$ to $2k$ are obtained using again MRC, this time for message $\mathbf{u}_2$. This process is iterated until block number M. In general, at block t the L-values of the first $k(t-1)$ symbols are calculated using the a-posteriori probabilities coming from the systematic part of block $t-1$, while the symbols from $k(t-1)+1$ to kt are calculated using MRC:

$$\mathbf{z}_t[k(t-1)+1:kt] = \sum_{m=t}^{M} h_m \mathbf{y}_m[k(t-1)+1:kt]. \tag{4}$$

The last $n-kt$ symbols are set to the non-systematic part of packet t. Once the posteriors for packet M are calculated, the macro forward-iteration ends and the macro backward-iteration starts. The LDPC decoders are run again starting from message $M-1$ and proceeding backward until message 1. The prior probabilities for the systematic part of block t are set to the corresponding posteriors from block $t+1$. Once the backward iteration ends (i.e., LDPC iterations for block 1 are completed), the final probabilities for each bit are calculated. This is done by averaging out the a-posteriori probabilities of each bit across all blocks in which it is present[5]. Finally, a hard decision with threshold 0.5 is performed. Note that the hard decision on message $\mathbf{u}_M$ can be taken right at the end of the forward super-iteration, since the backward iteration does not modify the a-posteriori probabilities for such message. The forward super-iteration is depicted in Fig. 2, while the backward super-iteration is shown in Fig. 3.

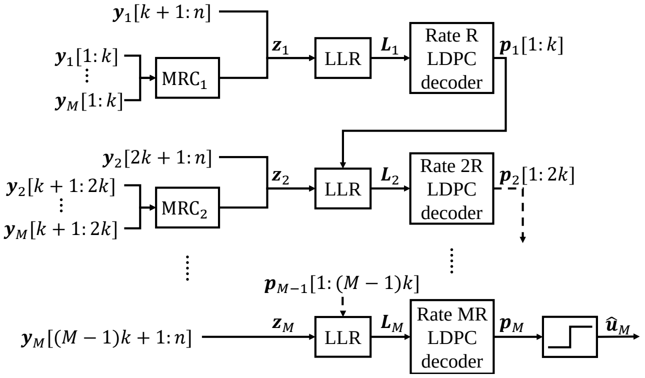

Figure 2. Decoder: forward super-iteration.

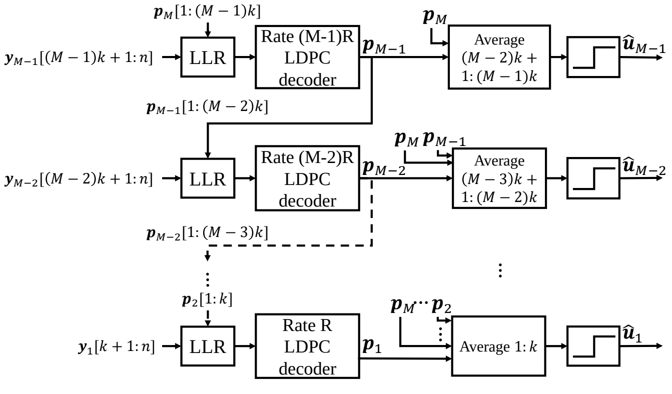
$\boldsymbol{p}_M[1:(M-1)k]$
$\boldsymbol{y}_{M-1}[(M-1)k+1:n]$
LLR
Rate (M-1)R LDPC decoder
$\boldsymbol{p}_{M-1}$
$\boldsymbol{p}_M$
Average $(M-2)k+1:(M-1)k$
$\hat{\boldsymbol{u}}_{M-1}$
$\boldsymbol{p}_{M-1}[1:(M-2)k]$
$\boldsymbol{y}_{M-2}[(M-2)k+1:n]$
LLR
Rate (M-2)R LDPC decoder
$\boldsymbol{p}_{M-2}$
$\boldsymbol{p}_M\,\boldsymbol{p}_{M-1}$
Average $(M-3)k+1:(M-2)k$
$\hat{\boldsymbol{u}}_{M-2}$
$\boldsymbol{p}_{M-2}[1:(M-3)k]$
$\boldsymbol{p}_2[1:k]$
$\boldsymbol{y}_1[k+1:n]$
LLR
Rate R LDPC decoder
$\boldsymbol{p}_1$
$\boldsymbol{p}_M \cdot\cdot\, \boldsymbol{p}_2$
Average $1:k$
$\hat{\boldsymbol{u}}_1$

Figure 3. Decoder: backward super-iteration.

IV. Benchmark

The ADR is defined as $\overline{R} \triangleq R(1 - PLR)$, PLR being the packet loss rate. In [5] it is shown that the JE approach is asymptotically optimal in the sense that:

$$\lim_{M\to\infty} \overline{R} = \begin{cases} R, \text{if } R < \overline{C} \\ 0, \text{if } R > \overline{C}, \end{cases} \tag{5}$$

[4] Note that y_m in Eqn. 2 represents the received signal in slot number m. We recall that subscript defines the slot number in which a given codeword is received and, equivalently, the position of the codeword within the macro-block.

[5] This is to avoid taking the final decision based one single block, as it might suffer from deep fading.

where $\overline{C}$ is the ergodic capacity. According to 5, at most an ADR equal to R can be approached in the limit of infinite M if this is lower than the ergodic capacity. As mentioned in Section II, we consider a BPSK modulation for the sake of clarity and without loss of generality. The corresponding ergodic capacity can be calculated starting from the expression for the capacity of an antipodal modulation [12], and averaging it out with respect to the distribution of the channel coefficients, thus obtaining the following:

$$\overline{C} = E_h\left\{1-\frac{1}{2}\sum_{n=0}^{1}E_w\left\{\log_2\left[\sum_{i=0}^{1}\exp\left(-\frac{|a^{(n)}h+w-a^{(i)}h|^2-|w|^2}{2\sigma^2}\right)\right]\right\}\right\}, \tag{6}$$

where $a^{(0)}=+1$, $a^{(1)}=-1$, $w \simeq \mathcal{N}(0,\sigma^2)$ and $E_x\{.\}$ is the expectation with respect to x.

As a side remark we point out that, as observed in [5], the JE approach is asymptotically optimal as long as $R < \overline{C}$, while better approaches can be found otherwise. In fact, as shown in Eqn. (5), if the rate exceeds the ergodic capacity the ADR goes to zero in the limit of infinite M. This may lead to poor performance if the value of $\overline{C}$ is not known with sufficient accuracy or if a rate $R > \overline{C}$ is imposed at application level. As the results in [5] suggest, in such cases an approach based on MT, signal superposition or time-sharing might perform better than JE in terms of ADR. A comparison of the performance achieved by a practical implementation of such approaches cannot be tackled here for a matter of space and goes beyond the scope of the present letter.

V. Numerical Results

We evaluate the performance of the proposed approach in terms of ADR, *message*-FER and *block*-FER. The message-FER is the FER for a given message within a block while the block-FER is the average FER across the block. All generator matrices have been randomly generated according to [13]. We set $k = 20$ and $n = 4500$. The number of LDPC iterations was set to 50 and Rayleigh fading was considered. In Fig. 4 the ADR is plotted against the signal-to-noise ratio (SNR) $1/\sigma^2$ for blocks of size 1 (memoryless transmission (MT)), 2, 10 and 200 [6]. The upper bound $\min\{R,\overline{C}\}$ is also plotted and the point at which $R=\overline{C}$ is highlighted on such curve. From the figure it can be seen that the ADR increases with the block size, getting closer to the theoretical upper bound as the number M of messages per block grows larger. Although the result presented in [5] is an asymptotic one, the proposed scheme shows a significant improvement with respect to the MT case already for $M = 200$. In Fig. 5 the message-FER curves for a block with $M = 10$ messages is shown. In order to have a clean picture only the curves for messages in the set $\{1,4,7,10\}$ are shown. The figure shows that messages transmitted earlier in the block are more protected with respect to messages transmitted later (a similar behaviour is observed also in the rest of messages that are not shown in the figure). This is a desirable feature for transmission of sequentially compressed sources with delay constraints, such as real-time IPPPP video streams. In fact, due to the sequential compression, messages transmitted later within a block rely on previous ones in order to be decoded. Losing any of the previous messages would result in the impossibility to reconstruct the data from that point onwards.

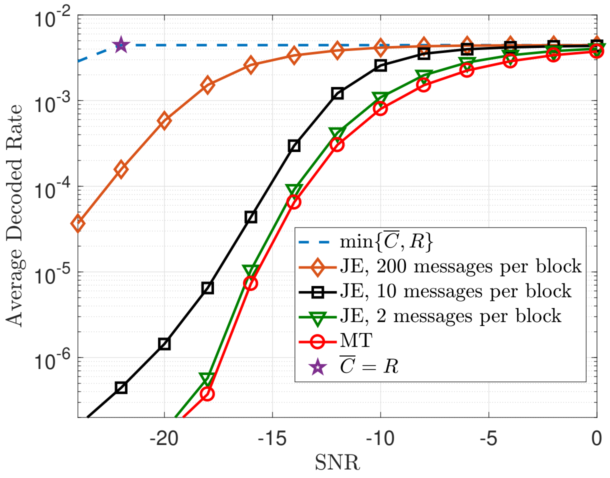

Figure 4. Average decoded rate for blocks with 1 (MT), 2, 10 and 200 messages. The upper bound $\min\{\overline{C},R\}$ is also shown.

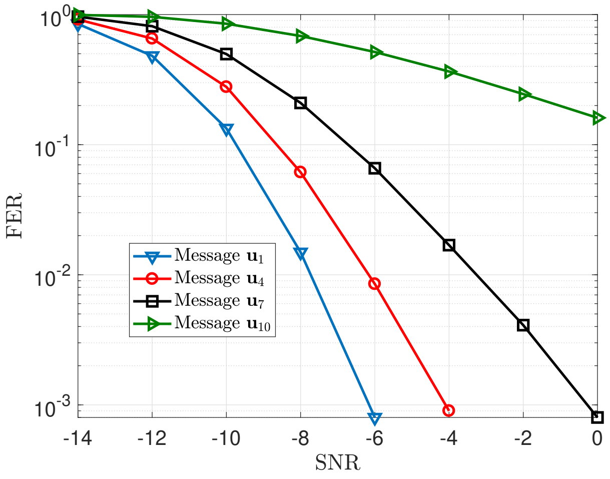

Figure 5. FER for messages $\mathbf{u}_1$, $\mathbf{u}_4$, $\mathbf{u}_7$ and $\mathbf{u}_{10}$ within a block of size 10. The FER increases with the message number, thus giving more protection to messages transmitted earlier. This is a desirable feature for sequentially encoded sources, such as, but not limited to, video streams with an IPPPP structure.

Besides providing increased protection against errors in earlier blocks, the proposed scheme also allows to decrease the block-FER with respect to MT. This can be seen in Fig. 6, where the block-FER is plotted against the SNR. As shown in the figure, the block-FER decreases for a given SNR as the number of messages per block increases.

[6] Note that, as mentioned in Section II, the number of frames in a GOP transported by a macro-block of size M can be significantly smaller than M, since a frame can be divided into several message blocks to meet the constraints of the physical layer.

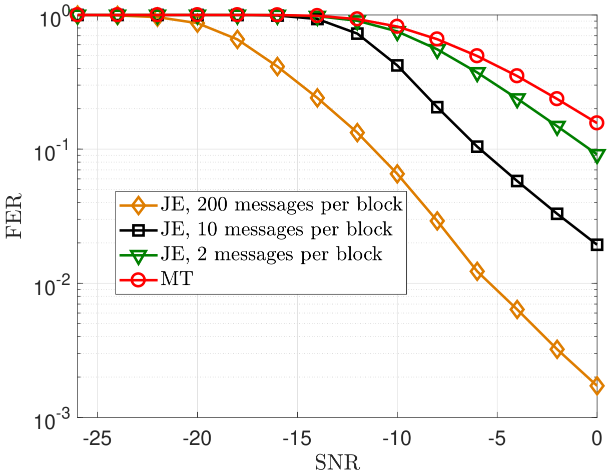

Figure 6. FER for blocks with 1 (MT), 2, 10 and 200 messages.

VI. Conclusion

Motivated by the problem of delay-constrained streaming from UAVs requiring high quality videos in playback, we proposed a new scheme based on cross-packet coding for transmission over a block fading channel with channel state information at the receiver only. The proposed scheme largely enhances upon a memoryless transmission approach in terms of average decoded rate and provides higher reliability to messages transmitted earlier within a block, which is desirable for successively compressed sources such as, but not limited to, IPPPP videos. The proposed scheme approaches the asymptotic upper bound over a wide range of SNR already for blocks with size of practical relevance. Up to our knowledge this is the first time that cross-packet coding is applied in such context.

Acknowledgements

G. Cocco would like to thank Gianluigi Liva for the helpful discussions.

References

[1] S. Haddadin, L. Johannsmeier, and F. Díaz Ledezma, "Tactile robots as a central embodiment of the tactile internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 471–487, Feb. 2019.

[2] *Recommendation ITU-T H.263; Infrastructure of audiovisual services – Coding of moving video*, https://www.itu.int/rec/T-REC-H.263/en.

[3] *Recommendation ITU-T H.264; Infrastructure of audiovisual services – Coding of moving video*, http://www.itu.int/rec/T-REC-H.264.

[4] A. Khisti and S. Draper, "The streaming-DMT of fading channels," *IEEE Trans. on Info. Theory*, vol. 60, no. 11, pp. 7058–7072, 2014.

[5] G. Cocco, D. Gündüz, and C. Ibars, "Streaming transmission over block fading channels with delay constraint," *IEEE Trans. on Wireless Commun.*, vol. 12, no. 9, pp. 4315–4327, Sep. 2013.

[6] C. Hausl and A. Chindapol, "Hybrid ARQ with cross-packet channel coding," *IEEE Commun. Letters*, vol. 11, no. 5, pp. 434–436, May 2007.

[7] J. Chui and A. Chindapol, "Design of cross-packet channel coding with low-density parity-check codes," in *IEEE Information Theory Workshop on Information Theory for Wireless Networks*, Solstrand, Norway, July 2007.

[8] M. Jabi, A. Benyouss, M. Le Treust, E. Pierre-Doray, and L. Szczecinski, "Adaptive cross-packet HARQ," *IEEE Trans. on Commun.*, vol. 65, no. 5, pp. 2022–2035, May 2017.

[9] K. Trillingsgaard and P. Popovski, "Generalized HARQ protocols with delayed channel state information and average latency constraints," *IEEE Trans. on Info. Theory*, vol. 64, no. 2, pp. 1262–1280, Feb. 2018.

[10] S. L. Fong, A. Khisti, B. Li, W. Tan, X. Zhu, and J. Apostolopoulos, "Optimal streaming codes for channels with burst and arbitrary erasures," *IEEE Trans. on Info. Theory*, vol. 65, July 2019.

[11] A. Badr, P. Patil, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Layered constructions for low-delay streaming codes," *IEEE Trans. on Info. Theory*, vol. 63, no. 1, pp. 111–141, 2017.

[12] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. on Info. Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.

[13] R. Neal, "Software for low density parity check codes," https://www.cs.utoronto.ca/~radford/ldpc.software.html, 2012.