

# Streaming Data Transmission by Using Spatially Coupled LDPC Codes

Sijie Wang, Zhongwei Si, Xuehong Lin

Key Lab of Universal Wireless Communications, Ministry of Education  
Beijing University of Posts and Telecommunications(BUPT), Beijing, China  
Email: wangsijie@bupt.edu.cn, sizhongwei@bupt.edu.cn, xhlin@bupt.edu.cn

**Abstract**—With the tremendous development of mobile Internet and big data, much attention has been attracted on streaming data transmission. In this paper, we propose a quasi real-time transmission scheme using spatially coupled LDPC codes for forward error correction. Recursive encoder and sliding-window decoder with low-latency and low-complexity are employed for data transmission. When any error is reported, additional information is requested and retransmitted. Capacity-approaching performance can be achieved by constructing proper ensembles according to channel conditions. Theoretical thresholds and simulation results are provided, which illustrate the good performance.

## I. INTRODUCTION

Streaming transmission is applied in many scenarios, such as online transactions, real-time data fed from sensors, satellite system, etc. The proliferation of the mobile Internet and the big data have fueled the development of applications that treat data as a continuous stream instead of a fixed set. Unlike traditional fixed set, streaming data is massive and fast changing, which brings in challenges for data transmission. The transmission schemes with low-delay and low-complexity are worth investigations. In this paper we focus on the channel coding technique for transmitting streaming data.

The low-density parity-check (LDPC) code was proposed by Gallager in [1]. Good performance can be achieved over various channel conditions by optimizing node degree distributions. Its convolutional counterpart, the spatially coupled LDPC code was proposed in [2]. It has been proved [3] [4] that the spatially coupled LDPC code universally achieves the capacity of the general binary memoryless symmetric channel. In addition, the regularity of degree distribution avoids complicated degree optimization.

For the implementation aspects, a recursive encoder with low-latency and low-complexity was introduced in [5]. A modified method for constructing the parity-check matrix of spatially coupled LDPC codes was proposed in [6], which guarantees the recursive encoding of arbitrary rates. A sliding-window decoder was invented in [7] and further discussed in [8] and [9]. The family of spatially coupled LDPC codes features efficient and continuous encoding and decoding, which enables the feasibility for streaming data transmission.

This work was supported in part by the National Natural Science Foundation of China (No.61401037), the Fundamental Research Funds for the Central Universities (No.2014RC0102, No.2014ZD03-01), and EU FP7 IRSES MobileCloudProject Grant (No.612212).

In this paper, we propose the implementation of a quasi real-time transmission scheme with hybrid automatic repeat request (HARQ) protocol. Spatially coupled LDPC codes are used for forward error correction (FEC), and a recursive encoder and a sliding-window decoder are employed in the system. The low-latency and low-complexity of the encoder and the decoder improve the efficiency, and the HARQ protocol is used to increase the robustness of the system. A capacity-approaching performance owing to the property of spatially coupled LDPC codes can be achieved by constructing proper ensembles according to channel conditions. Theoretical thresholds and simulation results are provided, which illustrate the good performance.

The remainder of the paper is organized as follows. An introduction to the quasi real-time transmission system is given in Section II. Section III describes the implementation aspects of this system. Performance analysis and numerical results are provided in Section IV and Section V. Section VI concludes the paper.

## II. THE QUASI REAL-TIME TRANSMISSION SYSTEM

In this section, a quasi real-time transmission system is introduced. Each module is discussed briefly.

### A. System Model

The system is depicted in Fig.1. The source generates information bits continuously, then these data are encoded by an encoder and transmitted. The received data are decoded by a decoder and sent to the destination if they are recovered successfully. HARQ is applied when the decoder cannot recover the information completely, thus a feedback channel for retransmission request is required. If the encoding and the decoding of the information is carried out in a streaming manner with low latency and if the transmission over the feedback channel is very quick, the system can be considered as a quasi real-time transmission. Spatially coupled LDPC codes are suitable channel coding technique for the model. By dividing the whole streaming data into small blocks and then encoding and decoding them recursively, the latency of the system is significantly reduced while the good performance is maintained.

### B. Spatially Coupled LDPC Codes

A rate  $R = 1 - d_v/d_c$  semi-infinite spatially coupled LDPC code ensemble  $(d_v, d_c)$  is introduced in the following, where

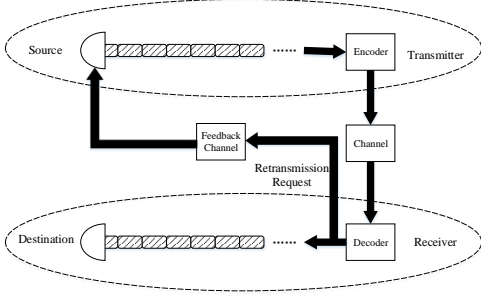


Fig. 1. Streaming data transmission system.

$d_v$  denotes the variable node degree, and  $d_c$  represents the check node degree. Let  $a = \text{gcd}(d_v, d_c)$  denote the greatest common divisor of  $d_v$  and  $d_c$ , and the memory length is  $m_s = a - 1$ . Then there exist positive integers  $d'_v$  and  $d'_c$  such that  $d_v = ad'_v$ ,  $d_c = ad'_c$  and  $\text{gcd}(d'_v, d'_c) = 1$ . We use  $t$  to denote the time instant of the code, then we obtain the parity-check matrix of the spatially coupled LDPC code as

$$\mathbf{H}_{[0, \infty]} = \begin{bmatrix} \mathbf{H}_0(0) & & & & & & & & & & \\ \mathbf{H}_1(0) & \mathbf{H}_0(1) & & & & & & & & & \\ \vdots & \mathbf{H}_1(1) & \ddots & & & & & & & & \\ \mathbf{H}_{m_s}(0) & \vdots & \ddots & \mathbf{H}_0(t-1) & & & & & & & \\ & \mathbf{H}_{m_s}(1) & \ddots & \mathbf{H}_1(t-1) & \mathbf{H}_0(t) & & & & & & \\ & & \ddots & \vdots & \mathbf{H}_1(t) & \ddots & & & & & \\ & & & \mathbf{H}_{m_s}(t-1) & \vdots & \ddots & & & & & \\ & & & & \mathbf{H}_{m_s}(t) & \ddots & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & \ddots & & & & \\ & & & & & & & \ddots & & & \\ & & & & & & & & \ddots & & \\ & & & & & & & & & \ddots & \end{bmatrix}, \quad (1)$$

where  $\mathbf{H}_i(t)$  is a  $d'_v \times d'_c$  submatrix, and each element of it is an  $M \times M$  permutation matrix.

Considering the encoding and decoding of streaming data, a recursive encoder and a sliding-window decoder based on spatially coupled LDPC codes are available. Low-delay and low-complexity of both modules guarantee the implementation of quasi real-time transmission system.

### C. The Recursive Encoder

To introduce the recursive encoder, we start with some definitions for a  $(d_v, d_c)$  ensemble of spatially coupled LDPC codes. First, an information sequence is denoted as

$$\mathbf{u}_{[0, \infty]} = [\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_t, \dots], \quad (2)$$

where  $\mathbf{u}_t = [u_t^0, u_t^1, \dots, u_t^{M(d'_c - d'_v) - 1}]$ ,  $u_t^k \in GF(2)$ ,  $t \geq 0$ . The code sequence is described as

$$\mathbf{v}_{[0, \infty]} = [\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_t, \dots], \quad (3)$$

where  $\mathbf{v}_t = [v_t^0, v_t^1, \dots, v_t^{M d'_c - 1}]$ ,  $v_t^k \in GF(2)$ ,  $t \geq 0$ . The recursive encoder is systematic and therefore  $\mathbf{v}_t$  can be represented as  $\mathbf{v}_t = [\mathbf{v}_t^{(0)}, \mathbf{v}_t^{(1)}]$ , where  $\mathbf{v}_t^{(0)}$  is information bits and  $\mathbf{v}_t^{(0)} = \mathbf{u}_t$ , and  $\mathbf{v}_t^{(1)}$  denotes the parity bits. The code sequence  $\mathbf{v}_{[0, \infty]}$  satisfies the equation

$$\mathbf{v}_{[0, \infty]} \mathbf{H}_{[0, \infty]}^T = \mathbf{0}. \quad (4)$$

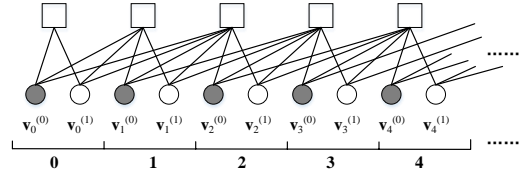


Fig. 2. Procedure of recursive encoding. Gray circles denote information bits, and white circles represent parity bits.

By using a  $(3, 6)$  ensemble as example, we illustrate the procedure of recursive encoding in Fig.2. At the first time instant, one check node connects to only two variable nodes. Assuming variable node  $v_0^{(0)}$  associates with only information bits, then the parity bits associated with  $v_0^{(1)}$  can be determined accordingly. At the second time instant,  $v_1^{(1)}$  represents information bits, and the value of  $v_1^{(1)}$  can be computed using  $v_0^{(0)}$ ,  $v_0^{(1)}$  and  $v_1^{(0)}$ . Similarly, at an arbitrary time instant, we can obtain the value of  $v_t^{(1)}$  according to  $v_{t-2}^{(0)}$ ,  $v_{t-2}^{(1)}$ ,  $v_{t-1}^{(0)}$ ,  $v_{t-1}^{(1)}$  and  $v_t^{(0)}$ . Therefore, the streaming data are encoded sequentially.

Cooperated with the sliding-window decoder described below, an efficient transmission system can be realized.

### D. The Sliding-Window Decoder

Exploiting the convolutional structure of spatially coupled LDPC codes that two bits contained in the same parity-check equation must be no more than  $m_s$  instants away from each other, a sliding-window decoder has been proposed. Here, we again take the  $(3, 6)$  ensemble as example to introduce the implementation of the sliding-window decoder.

Some definitions are given as follows. The window size is defined as  $W$ , which represents that  $W M d'_v$  rows and  $W M d'_c$  columns of  $\mathbf{H}$  are included in the window. All the edges contained in the window are denoted as *window configuration*, that is, the *window configuration* of a size  $W$  sliding-window decoder consist rows from  $t M d'_v$  to  $(t + W) M d'_v - 1$  and columns from  $t M d'_c$  to  $(t + W) M d'_c - 1$  within  $\mathbf{H}$  at instant  $t$ . Moreover, when the window is at time instant  $t$ , the variable nodes at instant  $t$  are called *targeted symbols*, namely the first  $M d'_c$  symbols in the window.

As illustrated in Fig.3, at any time instant  $t$ , the decoder performs belief propagation algorithm over the edges within the window with the aim of decoding all of the *targeted symbols* in the window. The window slides down  $M d'_v$  rows and right  $M d'_c$  columns in  $\mathbf{H}$  after all the *targeted symbols* have been recovered or a maximum number of belief propagation iterations have been performed. Then the decoder performs belief propagation algorithm in the new time instant. The streaming data are decoded in such a sequential manner.

## III. IMPLEMENTATION OF THE QUASI REAL-TIME TRANSMISSION

Streaming data transmission can be realized by using spatially coupled LDPC codes as we mentioned in Section II. In the following, we first propose a modified sliding-window

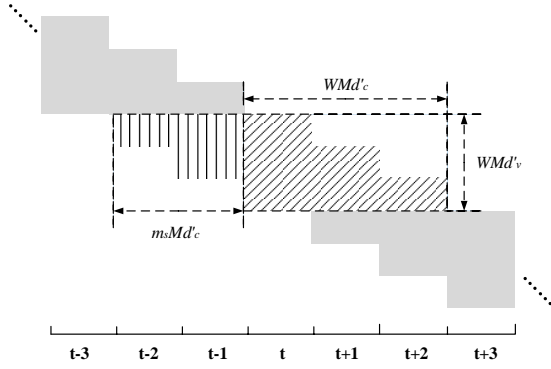


Fig. 3. Illustration of sliding-window decoding with window size  $W = 3$  for a  $(3, 6)$  ensemble of spatially coupled LDPC codes. The *window configuration* is the slash area, and the first  $2M$  symbols are the *targeted symbols*. The symbols within the vertical area is *a priori* information to help the decoding.

decoder. Then the implementation over the binary erasure channel is given.

#### A. The Modified Sliding-Window Decoder

Derived from the conventional sliding-window decoder, we propose in this paper a modified windowed decoder for the quasi real-time transmission system. It can be found that not only the *targeted symbols* but the symbols next to them are recovered quickly. Hence the modified sliding-window decoder does not slide until the iteration converges instead of recovering only the *targeted symbols* as the conventional decoder. As a result, it occurs with a probability that several frames can be recovered within one window. If the symbols in the first  $s$  blocks are all recovered successfully, the window slides down  $Msd'_v$  rows and right  $Msd'_c$  columns in  $\mathbf{H}$ . The window in the modified decoder can slide faster without any performance loss when transmitting over the BEC.

Because the complexity of belief-propagation decoding scales linearly in block length, the complexity of the sliding-window decoding is proportional to the window size. Accordingly, the decoding latency is also decided by the window size.

#### B. HARQ

When the channel condition is too poor for the decoder to recover all the source information, automatic repeat request is required to help the decoding. The decoder sends a retransmission request to the transmitter via the feedback channel, then additional bits will be transmitted to help recover the source information. In this paper, we only consider chase combining that the decoder uses both of the original information and retransmission information to perform decoding.

#### C. Transmission Procedure

The implementation of the quasi real-time transmission is introduced in the following. Let  $\mathbf{v}_t$  denote the  $t$ th frame transmitted. For each successfully decoded window, the decoder sends a *acknowledgement(ACK)* containing sign  $s$  to

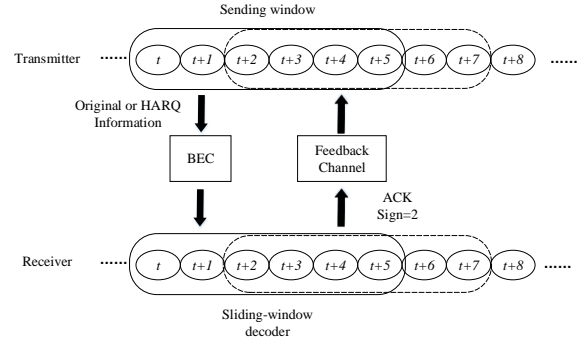


Fig. 4. Illustration of transmission with window size  $W = 6$  for quasi real-time system. When decoding at instant  $t$  has been finished, an ACK with sign that the first two frames has been recovered successfully is transmitted. Then two new frames are transmitted over the BEC.

the transmitter, where  $s$  represents the first  $s$  frames in the window has been decoded successfully.

- Step 1) **Transmission control:** Streaming data are generated by the source. They are encoded by the recursive encoder and transmitted over the BEC. When the frame  $\mathbf{v}_t$  to  $\mathbf{v}_{t+W-1}$  is being transmitted, the transmitter waits for the ACK. When an ACK containing sign  $s$  ( $s \neq 0$ ) is received, next  $s$  frames will be transmitted. An example is illustrated in Fig.4.
- Step 2) **Error detection:** When  $W$  frames to be decoded have arrived at the receiver, the sliding-window decoder is triggered and belief propagation is performed within one window. If the *targeted symbols* in the window cannot be recovered after a maximum number of belief propagation iterations, the decoder sends a retransmission request to the transmitter, i.e., a ACK with sign 0.
- Step 3) **HARQ:** The transmitter retransmits the *targeted symbols* when receiving the retransmission request, and then the decoder updates the channel observations of the *targeted symbols* according to the retransmitted information. The sliding-window decoder repeats the decoding process, i.e., Step 2.
- Step 4) **Re-encoding:** If a maximum number of retransmissions occurs but the *targeted symbols* are not recovered still, the decoder discards all the information in the window. The encoder restarts the encoding from the information bits corresponding to the *targeted symbols*. Then return to Step 1.

Streaming data are transmitted using the procedure we illustrated above. Moreover, if no frame is successfully recovered after a certain amount of re-encoding, a lower rate ensemble should be chosen to fit for the channel condition. Assuming the transmission over the feedback channel is quick, the streaming data transmission is considered as quasi real-time because the recursive encoder and the sliding-window decoder feature both low-latency and low-complexity.

#### IV. PERFORMANCE ANALYSIS

In this section, the latency of the system is discussed. Analysis based on density evolution is provided.

##### A. Latency

When ignoring the time for the transmission over the channel, the latency of the system only relates to the encoding and decoding. Moreover, the encoder can be considered as no delay since the information bits are encoded frame by frame. Therefore the latency of the decoder determines the latency of the system. We have the latency of the decoder

$$\Lambda_{WD} = T_W + T_{dec}(W), \quad (5)$$

where  $T_W$  is the time required to receive all the symbols in the window, and  $T_{dec}(W)$  is the time to decode one window when window size is  $W$ . As illustrated above,  $T_W$  is the time to receive  $s$  frames except the first decoding instant which needs to receive all the symbols for the first window.

Assuming a length- $L$  data is transmitted, the latency is proportional to  $L$  when using block LDPC codes. When the data are divided into  $N$  blocks and then encoded using spatially coupled LDPC codes, the latency reduces to be proportional to  $LW/N$ .

##### B. Density Evolution

From the procedure we described above, no error propagation occurs because every frame is either recovered completely or discarded. Ignoring the influence of sliding-window decoding at former time instant, the density evolution at any time instant retains the same. Let  $x_i$  define the remaining erasure probability at instant  $i$  and  $y_j$  denote the message passing from the check node  $j$  to variable node. For convenience, let  $i, j \in \{W\}$  represent the  $i$ th variable node or the  $j$ th check node within the window. First,  $x_i$  is initialized as follows,

$$x_i = \begin{cases} \epsilon, & i \in \{W\} \\ 0, & otherwise \end{cases}. \quad (6)$$

Then the values of  $x_i$  and  $y_j$  are updated as

$$x_i = \epsilon \left( \frac{1}{m_s + 1} \sum_{j=i}^{i+m_s} y_j \right)^{d_v - 1}, i \in \{W\} \quad (7)$$

and

$$y_j = \begin{cases} 1 - \left( 1 - \frac{1}{m_s + 1} \sum_{i=j-m_s}^j x_i \right)^{d_c - 1}, & j \in \{W\} \\ 1, & otherwise \end{cases}. \quad (8)$$

The density evolution of the (3,6) ensemble for  $W = 10$  and  $\epsilon = 0.45$  is shown in Fig.5. We find that the *targeted symbols* within the window converge quickly, e.g., the erased probability converges to 0 after approximately 20 iterations. This verifies the threshold saturation of terminated spatially coupled LDPC codes. The *targeted symbols* are recovered earlier than the rest because their check degree is lower than others.

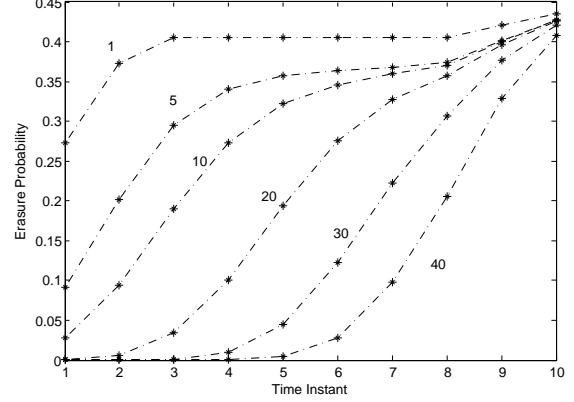


Fig. 5. Density evolution of the (3,6) ensemble, the number next to each dotted line is the iterative number.

In addition, we find that, after 40 iterations, the erased probability of the symbols at time instant 2,3,4 also converges to 0. That is, not only the *targeted symbols* but the symbols next to them also converge after a few more iterations. This phenomenon enables the implementation of the modified sliding-window decoder which has a higher sliding speed than the conventional one. Multiple frames may be transmitted when one *ACK* is received using this decoder.

The purpose of density evolution is to calculate the belief propagation threshold (threshold in short) which is an erasure probability for the BEC. An ensemble can be recovered by belief propagation algorithm when the channel erasure probability is no large than the threshold. Threshold is a main parameter to measure the performance of spatially coupled LDPC codes.

##### C. Throughput

A key measurement of an HARQ system is the throughput efficiency, which is defined as the average number of information bits successfully delivered to the destination through one channel use. In the system we described above, using  $N_d$  to denote the number of discarded frames and  $N_a$  to represent the number of all transmitted frames (including retransmissions), the throughput equals

$$Throughput = \frac{N_a - N_d}{N_a} \times Rate. \quad (9)$$

The highest throughput achieves when all the frames are received successfully without retransmissions. Therefore the highest throughput is  $(1 - \epsilon)$  when the channel erasure probability of a BEC equals  $\epsilon$ .

#### V. NUMERICAL RESULTS

In this section, we provide the numerical results. The throughput of the system is provided according to both the density evolution and the finite-length simulations.

The theoretical throughput of the system is related to the decoding threshold of spatially coupled LDPC codes obtained via density evolution. When the threshold of a rate  $r_1$  ensemble

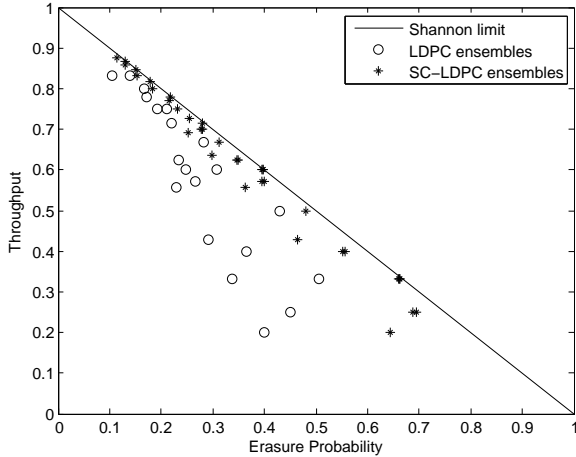


Fig. 6. Theoretical throughput for quasi real-time transmission system.

$\mathcal{C}_1$  is  $\epsilon_{C_1}$ , as mentioned earlier, the theoretical throughput is  $r_1$  when  $\epsilon \leq \epsilon_{C_1}$  if ensemble  $\mathcal{C}_1$  is used. There also exists a rate  $r_2$  ensemble  $\mathcal{C}_2$  whose threshold is  $\epsilon_{C_2}$ , and  $\epsilon_{C_2}$  is always smaller than  $\epsilon_{C_1}$  when  $r_2 > r_1$ . So when using this two ensembles, we choose  $\mathcal{C}_1$  to achieve throughput  $r_1$  when  $\epsilon = \epsilon_{C_1}$  and  $\mathcal{C}_2$  to achieve throughput  $r_2$  when  $\epsilon = \epsilon_{C_2}$ . In this way, a relationship between the threshold and the theoretical throughput is built.

When fixing  $d_v$  and varying the value of  $d_c$ , we construct a series of ensembles with various design rates. Diverse ensembles are generated for  $d_v = \{3, 4, 6, 8, 9\}$ , and the theoretical throughput is illustrated in Fig.6. For the purpose of comparison, we also include the theoretical throughput achieved by using regular LDPC block codes with the same setup of node degrees. It can be observed that spatially coupled LDPC codes achieve in general a higher throughput than the regular LDPC block codes. Note that, in addition to the better performance, the spatially coupled LDPC codes also allows quasi real-time processing.

In the following we provide the numerical results of the achieved system throughput using finite-length spatially coupled LDPC codes. We use  $\mathcal{C}_i(d_v, d_c)$  to denote an ensemble of spatially coupled LDPC codes and construct 5 ensembles including:  $\mathcal{C}_1(3, 12)$ ,  $\mathcal{C}_2(3, 9)$ ,  $\mathcal{C}_3(4, 10)$ ,  $\mathcal{C}_4(3, 6)$  and  $\mathcal{C}_5(4, 6)$ . The parameter  $M$  is 512, the window size  $W$  is set to be 10, and the maximum retransmission number is 3. Simulations for erasure probability  $\epsilon \in (0, 1)$  are performed.

As Fig.7 shows, with different channel erasure probabilities there exist different ensembles which provide the efficiency close to the Shannon limit. For different range of the channel condition, we may choose a proper coding realization to achieve the capacity-approaching performance. We can have a similar conclusion as from the theoretical throughput, the spatially coupled LDPC code provides a high-efficiency and low-latency solution to the streaming data transmission.

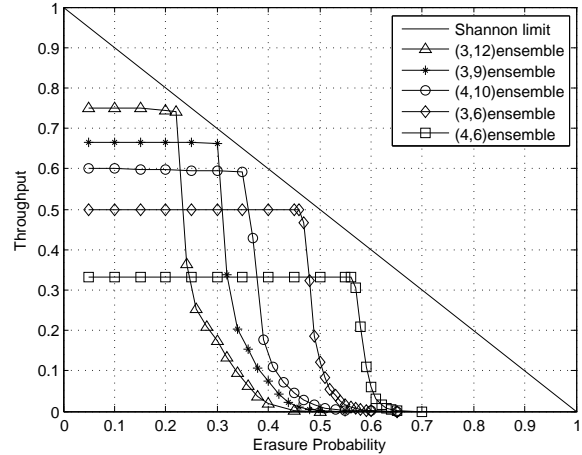


Fig. 7. Simulation results by using  $\mathcal{C}_1$ ,  $\mathcal{C}_2$ ,  $\mathcal{C}_3$ ,  $\mathcal{C}_4$  and  $\mathcal{C}_5$ .

## VI. CONCLUSION

In this paper we have proposed a quasi real-time transmission scheme by using spatially coupled LDPC codes as the forward error correction. Employing recursive encoder and sliding-window decoder, low-latency and low-complexity transmission for streaming data is realized. When any error is reported, a retransmission of additional information is triggered to aid the decoding, which increases the robustness of the system. Capacity-approaching performance can be achieved by constructing proper ensembles according to channel conditions. Both theoretical thresholds and simulation results illustrate the good performance.

## REFERENCES

- [1] R. G. Gallager. *Low Density Parity Check Codes*. MIT Press, 1963.
- [2] A. Jimenez-Felström, and K. S. Zigangirov. "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 2181-2191. Sept. 1999.
- [3] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov. "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inform. Theory*, vol. 56, no. 10, pp. 5274-5289. Oct. 2010.
- [4] S. Kudekar, T. Richardson, and R. Urbanke. "Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 803-834. Feb. 2011.
- [5] A. E. Pusane, A. Jimenez-Felström, A. A. Sridharan, M. M. Lentmaier, K. S. Zigangirov, and D. J. Costello, Jr.. "Implementation aspects of LDPC convolutional codes," *IEEE Trans. Commun*, vol. 56, no. 7, pp. 1060-1069, Jul. 2008.
- [6] J. Ma, Z. Si, Z. He, K. Niu. "Recursive encoding of spatially coupled LDPC codes with arbitrary rates," *26th IEEE Int. Symp. PIMRC*, Hong Kong, China, Aug, 2015.
- [7] M. Papaleo, A. R. Iyengar, P. H. Siegel, J. Wolf, and G. Corazza. "Windowed erasure decoding of LDPC convolutional codes," *IEEE Inform. Theory Workshop*, Cairo, Egypt, pp. 78-82, Jan. 2010.
- [8] G. E. Corazza, A. R. Iyengar, M. Papaleo, P. H. Siegel, A. Vanelli-Coralli, and J. K. Wolf, "Latency constrained protograph-based LDPC convolutional codes," *6th Int. Symp. Turbo Codes Iterative Inf. Process.*, Brest, France, pp. 6-10, Sep. 2010.
- [9] R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza. "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inform. Theory*, vol. 58, no. 4, pp. 2303-2320, Apr. 2012.