

# Capacity Bounds and Concatenated Codes over Segmented Deletion Channels

Feng Wang, Tolga M. Duman, *Fellow, IEEE*, and Defne Aktas, *Member, IEEE*

**Abstract**—We develop an information theoretic characterization and a practical coding approach for segmented deletion channels. Compared to channels with independent and identically distributed (i.i.d.) deletions, where each bit is independently deleted with an equal probability, the segmentation assumption imposes certain constraints, i.e., in a block of bits of a certain length, only a limited number of deletions are allowed to occur. This channel model has recently been proposed and motivated by the fact that for practical systems, when a deletion error occurs, it is more likely that the next one will not appear very soon. We first argue that such channels are information stable, hence their channel capacity exists. Then, we introduce several upper and lower bounds with two different methods in an attempt to understand the channel capacity behavior. The first scheme utilizes certain information provided to the transmitter and/or receiver while the second one explores the asymptotic behavior of the bounds when the average bit deletion rate is small. In the second part of the paper, we consider a practical channel coding approach over a segmented deletion channel. Specifically, we utilize outer LDPC codes concatenated with inner marker codes, and develop suitable channel detection algorithms for this scenario. Different maximum-a-posteriori (MAP) based channel synchronization algorithms operating at the bit and symbol levels are introduced, and specific LDPC code designs are explored. Simulation results clearly indicate the advantages of the proposed approach. In particular, for the entire range of deletion probabilities less than unity, our scheme offers a significantly larger transmission rate compared to the other existing solutions in the literature.

**Index Terms**—Segmented deletion channel, synchronization errors, information stability, capacity bounds, MAP detection, LDPC codes, marker codes.

## I. INTRODUCTION

**S**YNCHRONIZATION errors represent important types of channel impairments impacting design of communication systems. Such errors may be caused by a mismatch between

Paper approved by S.-Y. Chung, the Editor for LDPC Coding and Information Theory of the IEEE Communications Society. Manuscript received December 8, 2011; revised July 10 and September 28, 2012.

This work is funded by the National Science Foundation under the contract NSF-TF 0830611.

Part of the work was presented at the Forty-Ninth Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 2011.

F. Wang is with the School of Electrical, Computer and Energy Engineering (ECEE), Arizona State University, Tempe, AZ 85287-5706, USA (e-mail: feng.wang83@asu.edu).

T. M. Duman is with the Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara, 06800, Turkey (e-mail: duman@ee.bilkent.edu.tr). He is on leave from the School of ECEE of Arizona State University.

D. Aktas is with the Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara, 06800, Turkey (e-mail: dak-tas@ee.bilkent.edu.tr).

Digital Object Identifier 10.1109/TCOMM.2012.010213.110836

the transmitter and receiver clocks, or imperfect timing-alignment, e.g., in the read/write process of a bit-patterned media recording system [1]. As a result of these errors, transmitted symbols may be deleted and random symbols may be inserted into the received data stream. The positions of the deletions or insertions are unknown, and the resulting models are referred to as insertion/deletion channels. Since the positions of the insertions/deletions are unknown, study of such channels from an information theoretic or a practical coding point of view prove to be remarkably difficult [2].

Various models for insertion/deletion channels have been proposed in the literature [3]–[5]. Most models assume that synchronization errors occur independently of each other. In this paper, we consider a different type of synchronization errors, i.e., binary insertion/deletion channels with the additional segmentation assumption, which was first introduced in [6]. According to the segmentation assumption, several consecutively transmitted bits are considered as one block or segment and the number of insertions/deletions within each segment is limited to a certain number. Motivation of studying this channel model is that the segmentation assumption appears naturally in many practical systems. For instance, consider the bit-patterned media recording systems for which the read-head cannot perfectly match to the pre-patterned magnetic islands [1]; when a particular island is skipped or read multiple times (due to cycle slips), the next deletion/insertion event will appear only after some time. See also [7] for intermittent errors in high density magnetic recording channels. Another example of this channel model is a wireless communication system with a varying sampling rate, caused by an inadequate yet low-cost timing recovery block [8]. Only during the interval of changing the sampling rate, insertion/deletion errors may occur, which corresponds to several segments of bits with synchronization errors.

Most of the previous results on the capacity bounds for channels with synchronization errors focus on the (bit-wise) i.i.d. channel model [9]–[13], and cannot be directly used for the case of segmented deletion channels. For example, in a recent paper [14], the authors specify a memoryless synchronization error channel by a stochastic transition probability matrix, and obtain analytical lower bounds on the capacity for channels with deletions or duplications only, some of which are expected to be tight for small deletion or duplication probabilities. There is also little work on practical channel coding schemes over the segmented deletion channel, as most of the existing code designs for i.i.d. insertion/deletion channels cannot be directly applied, e.g., in [15]–[18]. The only existing coding approach for this channel is given in [6], where the

proposed codes can correct all the insertions/deletions with no errors when only a single deletion/insertion error per segment is allowed. The key idea is to encode the data sequence so that each segment is a codeword from a 1-deletion/insertion correcting code. Other constraints are also enforced on the codewords which allow for a simple left-to-right, segment by segment decoding algorithm. As an example, a codebook containing 12 codewords is found for a segment size of  $b = 8$ , resulting in an overall code rate of  $R = 0.448$ . Higher code rates can be achieved for larger  $b$ . Although some extensions have also been studied offering higher code rates, these coding algorithms require some check bits and check sums to be known at the receiver side leading to the need of a perfect side-information channel [6].

In this paper, we aim at the development of both the information theoretic and practical coding results for the segmented deletion channel. In particular, we consider the elementary segmented deletion channel, i.e., no more than one deletion per segment is allowed. We first show that the segmented deletion channel falls within the framework of memoryless synchronization channels (with non-binary inputs), and by a proper application of Dobrushin's results [19], we prove that the channel is information stable. Then, we explore several upper and lower bounds on their capacity by providing the transmitter and the receiver with genie-aided information, e.g., about which segment has a deletion error. We demonstrate that the derived upper and lower bounds behave similarly for some range of deletion probabilities, while a wide-range of deletion probabilities exist where further improvement of the results is clearly possible. As an alternative approach, we also show that when the average bit deletion rate is small, asymptotic behavior of the capacity can be derived by following a similar methodology developed in [20] for the case of an i.i.d. deletion channel. As a result, good approximations of the channel capacity for small deletion probabilities or large segment lengths are obtained.

In addition to the capacity characterization of the channel, we also consider a practical concatenated coding approach, for which as in [21], concatenation of an outer LDPC code for error-correction with an inner marker code, which provides re-synchronization capabilities, is explored. Despite the similar encoding procedure (with the i.i.d. insertion/deletion channels), there are significant differences from the previous work [21]. In particular, the soft-output synchronization algorithm in [21] is no longer optimal. Therefore, we introduce bit-level and symbol-level MAP detection algorithms which incorporate the segmentation assumption for improved results. Our approach is motivated by the fact that if we allow for the use of powerful codes with strong error-correcting capabilities, a much higher code rate (compared to the ones reported in [6]) may be achieved with a low probability of error (when we drop the zero-error requirement).

The rest of the paper is organized as follows. We start with a detailed discussion of the channel model in the next section. In Section III, we prove that the Shannon capacity exists, and describe several capacity upper and lower bounds for the segmented deletion channel. Asymptotic behavior of the capacity is also explored for two cases: as the deletion probability  $P_d$  approaching zero with a finite segment length

$b$ , and as the segment length approaches infinity for an arbitrary  $P_d$ . In Section IV, we turn our attention to a practical concatenated coding scheme along with suitable MAP detection algorithms for synchronization which incorporate the segmentation assumption. In Section V, results of LDPC code design for segmented deletion channel are provided, along with Monte-Carlo simulation results for several randomly chosen and specifically designed codes are compared. Finally, concluding remarks are provided in Section VI.

## II. SEGMENTED DELETION CHANNEL MODEL

In this section, we introduce the channel model under consideration more precisely. The channel is a binary input and binary output channel, and the transmitted bit sequence is implicitly partitioned into  $N$  consecutive disjoint blocks  $\{\mathbf{X}_n\}_{n=1}^N$ , and each with length  $b$  bits. There is no explicit segment partitioning step at the transmitter side, however, the receiver is aware of this restriction. During the transmission, a total number of at most  $d_0$  insertions and deletions are allowed to happen for each  $\mathbf{X}_n$ , resulting in a received vector of varying lengths. If we utilize the insertion model in [3], the length of the received vector corresponding to  $\mathbf{X}_n$  takes values in the set  $\{b - d_0, \dots, b, \dots, b + d_0\}$ , and the positions of insertion/deletion errors are uniformly chosen within the segment. In addition to the synchronization errors, substitution errors can also be incorporated [3], [4], i.e., every non-deleted bit maybe incorrectly received with probability  $P_s$ . There is no special marker between the bits of different segments, hence the receiver does not know the segment boundaries.

Throughout the paper, we focus on a particular case, namely, the elementary segmented deletion channel, i.e., the segment  $\mathbf{X}_n$  is received intact with probability  $1 - P_d$ , while one bit within  $\mathbf{X}_n$  is deleted with probability  $P_d$ . If a deletion occurs in  $\mathbf{X}_n$ , each bit within this segment is equally likely to be deleted. Also, the deletions for each segment are independent. A simple example is given as follows. Assuming that the binary sequence 00101101 is transmitted over a segmented deletion channel with  $b = 4$ , it is possible that the third and fifth bits are deleted, leading to a received sequence of 000101. However, receiving 001001 is impossible as in this case two bits from the second segment would need to be deleted, which is not allowed.

## III. CAPACITY BOUNDS FOR SEGMENTED DELETION CHANNELS

### A. Existence of the Shannon Capacity

We first show that the results of Dobrushin in [19] can be applied directly to the segmented deletion channel model and as a result the Shannon capacity exists. The key observation is that Dobrushin's result is more general than the usual set-up that it is applied to, that is, information stability [22] holds for a memoryless channel with synchronization errors, indicating that the asymptotic behavior of the mutual information density between the input and output sequences over the sequence length converges to its mean. Therefore, the Shannon capacity exists, even when the channel input and output alphabets are not identical (e.g. binary) and the information and the transmission capacities are equal. The segmented deletion

TABLE I  
EXAMPLE OF TRANSITION PROBABILITY  $P(\mathbf{Y}'|X')$  FOR  $b = 2$ .

$X'$	$\mathbf{Y}' = 00$	$\mathbf{Y}' = 01$	$\mathbf{Y}' = 10$	$\mathbf{Y}' = 11$	$\mathbf{Y}' = 0$	$\mathbf{Y}' = 1$
0 (00)	$1 - P_d$	0	0	0	$P_d$	0
1 (01)	0	$1 - P_d$	0	0	$P_d/2$	$P_d/2$
2 (10)	0	0	$1 - P_d$	0	$P_d/2$	$P_d/2$
3 (11)	0	0	0	$1 - P_d$	0	$P_d$

channel model can equivalently be described by a  $2^b$ -ary input symbol  $X'$ , and binary sequence of output bits  $\mathbf{Y}'$  (of varying lengths, e.g., for the elementary segmented deletion channel, of length  $b$  or  $b - 1$  bits), it is clear that the model in [19] encompasses as a special case the segmented deletion channel model (when the deletions occur independently in different segments). To illustrate this point further, let us give a simple example. Consider the segmented deletion channel with  $b = 2$  and deletion probability of  $P_d$ . The equivalent channel transition matrix  $P(\mathbf{Y}'|X')$  is as given in Table I.

With the above explanation, from [19], we can safely say that the segmented deletion channel is information stable, and hence its Shannon capacity exists. In fact, the capacity per transmitted bit is given by

$$C = \lim_{T \rightarrow \infty} \frac{1}{T} \max_{P(\mathbf{X})} I(\mathbf{X}; \mathbf{Y}),$$

where  $I(\cdot; \cdot)$  is the mutual information between the input sequence  $\mathbf{X}$ , of length  $T$ , and output sequence  $\mathbf{Y}$ .

Although the channel capacity exists, evaluation of the capacity expression is not straightforward. That is, there is no single-letter or finite-letter formulation which may be amenable for practical computation which is also the case for other channel models with synchronization errors. With this motivation, we next introduce two simple upper/lower bounds on the capacity of segmented deletion channels. First of all, an obvious capacity upper bound can be obtained by providing side information to the receiver about the positions of all the deletions. Therefore, the channel becomes a binary erasure channel with memory and an erasure probability  $P_d/b$ . Since the memory does not affect the capacity of an erasure channel [23],  $1 - P_d/b$  becomes a trivial upper bound on the channel capacity. To obtain a lower bound, we assume that a long interleaver has been introduced before transmission, and the corresponding deinterleaver is used at the receiver before decoding. The equivalent channel is then a binary i.i.d. deletion channel. Since this is a specific signaling scheme, any achievable rate over a binary i.i.d. deletion channel with probability  $P_d/b$  would be achievable on the segmented deletion channel providing us with a lower bound on the channel capacity.

### B. Capacity Upper and Lower Bounds with Side Information

In [13], to obtain an upper bound on the capacity for an i.i.d. deletion channel, some suitable genie-aided information on the deletion process is revealed to the receiver so that the channel becomes memoryless. For the segmented deletion channel, we propose a similar method of obtaining upper and lower bounds on the capacity by providing some side information to both the transmitter and the receiver.

1) *Upper Bound - Version 1:* Define the random process  $\mathbf{V} = \{V_n\}_{n=1}^N$ , where  $V_n$  is a binary valued random

variable which determines whether the  $n$ -th segment  $\mathbf{X}_n$  experiences a deletion error or not. With the side information being provided to both the transmitter and receiver, we have

$$C \leq \frac{1}{b} \max_{P(\mathbf{X}_n)} I(\mathbf{X}_n; \mathbf{Y}_n),$$

where  $\mathbf{Y}_n$  is the received sequence corresponding to  $\mathbf{X}_n$  with length either  $b$  or  $b - 1$ .

Obviously,  $1 - P_d$  fraction of the blocks see noiseless channels, hence with the transmitter/receiver side information, we can transmit  $b$  bits with no error. The remaining  $P_d$  fraction of blocks will equivalently see a deletion channel with  $b$  input bits and exactly one deletion at the output. The capacity of such a channel can be computed (for reasonable values of  $b$ )<sup>1</sup> using the Blahut-Arimoto Algorithm (BAA) [24], [25]. Denoting the capacity of the deletion channel with  $b$  input and  $b - 1$  output bits (where the deleted bit position is random and uniform) as  $C_d(b, 1)$ , we can write an upper bound on the capacity of segmented deletion channel as

$$C \leq 1 - P_d + P_d \frac{1}{b} C_d(b, 1). \quad (1)$$

2) *Upper Bound - Version 2:* Following similar line of arguments, we expect the capacity upper bound to be tighter when “less” side information is provided to the transmitter and the receiver. For example, we define the random process  $\mathbf{W} = \{W_{n'}\}_{n'=1}^{N/2}$ , where  $W_{n'}$  is a random variable taking values  $\{0, 1, 2\}$ , which determines the number of deletions in every pair of segments, i.e., in  $\mathbf{X}_{2n'-1}$  and  $\mathbf{X}_{2n'}$ . When  $W_{n'}$  equals 0 or 2, it contains the same information as in the previous case. Ambiguity only rises when  $W_{n'} = 1$ , since in this case, we have no idea which one of the two segments has the deleted bit, and we simply have a channel with  $2b$  bits at the input and one deletion. Therefore, we can write the capacity upper bound as

$$\begin{aligned} C &\leq (1 - P_d)^2 + 2P_d(1 - P_d) \frac{1}{2b} C_d(2b, 1) + P_d^2 \frac{1}{2b} C_d(2b, 2) \\ &= (1 - P_d)^2 + P_d(1 - P_d) \frac{1}{b} C_d(2b, 1) + P_d^2 \frac{1}{b} C_d(b, 1), \quad (2) \end{aligned}$$

where  $C_d(2b, 2)$  denotes the capacity of a channel with  $2b$  bits of input and one deletion in the first  $b$  bits and another one among last  $b$  bits. The second line follows since for the channel with  $K$  segments of input bits and one deletion in each segment, we can deduce the boundaries of every segment in the received bit sequence without any additional information and hence,  $C_d(Kb, K) = K C_d(b, 1)$ . Comparing (1) and (2), it is obvious that with the random process  $\mathbf{W}$ , we are able to expand the capacity upper bound as a quadratic function of  $P_d$  and thus obtain a tighter result, as will be shown later. Even tighter bounds can be achieved when less and less side information is used at the expense of a much heavier computational load on the BAA algorithm. Details of this further generalization is omitted from this paper.

For large values of  $b$  that are not amenable for the BAA, one can resort to the upper bound  $C_d(B, 1) \leq C_d(b, 1) + (n - 1)b$  reported in [13], where  $B = nb$ . The bound is tight for large

<sup>1</sup>The largest value of  $b$  we could handle in our computations was 24.

$B$  as it is also shown that

$$C_d(B, 1) \geq C'_d(b, 1) + (n-1)b - H\left(\frac{1}{n}\right), \quad (3)$$

where  $H(\cdot)$  is the binary entropy function and  $C'_d(b, 1)$  is the achievable rate for a deletion channel with  $b$  independent uniformly distributed (i.u.d.) input bits and exactly one deletion. The gap between the upper and lower bounds of  $C_d(B, 1)$  gets smaller as  $n$  increases, since the entropy term  $H(\frac{1}{n})$  approaches zero. When  $B \neq nb$ , another upper bound can also be used [13]:

$$C_d(B, 1) \leq \frac{B-1}{B} (2 + C_d(B-1, 1)). \quad (4)$$

3) *Lower Bounds:* Capacity lower bounds can be obtained by revealing some side information to the receiver, and then by subtracting a certain term to make sure what is obtained is in fact a lower bound. Specifically, we can write

$$I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}, \mathbf{V}) - I(\mathbf{X}; \mathbf{V} | \mathbf{Y}) \geq I(\mathbf{X}; \mathbf{Y}, \mathbf{V}) - H(\mathbf{V}).$$

To compute  $I(\mathbf{X}; \mathbf{Y}, \mathbf{V})$ , we cannot optimize the input distribution for every segment, since the side information is only provided to the receiver. Instead, we consider i.u.d. inputs. Hence, the following capacity lower bound is obtained,

$$C \geq 1 - P_d + P_d \frac{1}{b} C'_d(b, 1) - \frac{1}{b} H(P_d). \quad (5)$$

Comparing the capacity upper bound in (1) and lower bound in (5), we see that the difference is  $P_d \frac{1}{b} (C_d(b, 1) - C'_d(b, 1)) + \frac{1}{b} H(P_d)$ . When  $P_d$  approaches zero or one, the term  $\frac{1}{b} H(P_d)$  tends to zero. In fact, when  $P_d$  equals zero or one, the segmented deletion channel becomes a memoryless channel without any synchronization problems, and the capacity is exactly as given in (1). Furthermore, for large  $b$  values, we would expect  $\frac{1}{b} (C_d(b, 1) - C'_d(b, 1))$  to be small. The reason is that since the i.u.d. input sequences are optimal for the calculation of  $C_d(b, 0)$ , when the overall deletion rate per transmitted bit  $\frac{1}{b}$  goes to zero, i.u.d. inputs will be close to optimal, and therefore, the gap between the upper and lower bound on the capacity becomes very small. This observation is quantified in [20], which proves that for an i.i.d. deletion channel with a small deletion probability, i.u.d. inputs achieve the first order term of the channel capacity when we express it as a series expansion in terms of the deletion probability.

Our final argument is that this approach can be easily extended to the case when more than one synchronization errors are allowed in each segment, however, this specific model is not considered here.

### C. Asymptotic Behavior of the Segmented Deletion Channel Capacity

We now focus on the case where  $P_d/b$  is small and characterize the capacity for a segmented deletion channel using a similar approach employed in [20] for the case of i.i.d. deletion channels. In particular, for a finite segment length  $b$  with  $P_d$  approaching zero, and for a fixed  $P_d$  with large segment length  $b$ , we show that the capacity can be characterized asymptotically, and therefore, an approximation

to the exact channel capacity can be obtained for small  $P_d/b$  values.

It is proved in [20] that when computing the channel capacity or capacity bounds for an i.i.d. deletion channel, one can restrict the input sequence to be a stationary ergodic process  $\mathbf{X} = \{X_i\}$  with  $X_i \in \{0, 1\}$ . We make an observation that the same argument also holds for the segmented deletion channel as all the steps in the proof remain valid for our case following a similar approach.

Let  $\mathbf{X}^n$  be the input sequence of length  $n$  and  $\mathbf{Y} = \mathbf{Y}(\mathbf{X}^n)$  represent the corresponding output sequence. Define  $S_L$  to be the set of stationary ergodic processes such that no run (consecutive bits of the same value) has length larger than  $L$  and  $\mathbf{X}^*$  to be the i.i.d. Bernoulli(1/2) process, i.e.,  $X_i^*$  equals 0 or 1 with probability 1/2 each. We define  $I(\mathbf{X}^n) = \lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{X}^n; \mathbf{Y}(\mathbf{X}^n))$ ,  $H(\mathbf{X}) = \lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{X}^n)$ . Theorems 1-3 below present our main results. We note that the proofs of these theorems and the related lemmas are extensions of the corresponding ones in [20], which considers i.i.d. deletion channels.

**Theorem 1.** Consider a segmented deletion channel with a fixed segment length  $b$  and deletion probability  $P_d$  approaching zero. We have  $\forall \epsilon > 0$ ,

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{X}^{*,n}; \mathbf{Y}(\mathbf{X}^{*,n})) \\ = 1 - \frac{P_d}{b} (1 + \log_2 b - A) - \frac{H(P_d)}{b} - O(P_d^{2-\epsilon}), \end{aligned} \quad (6)$$

where  $A = \sum_{l=1}^{\infty} 2^{-l-1} l \log_2 l \approx 1.28853$ , and  $O(\cdot)$  is the standard big  $O$  notation. Clearly, this is an achievability result and serves as a lower bound on the capacity of the segmented deletion channel as  $P_d \rightarrow 0$ .

**Theorem 2.** For a segmented deletion channel with a fixed segment length  $b$  and deletion probability  $P_d$  approaching zero, there exists  $P_{d,0} > 0$  such that  $\forall P_d < P_{d,0}$  and  $\epsilon > 0$ , for any input process we have

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{X}^n; \mathbf{Y}(\mathbf{X}^n)) \\ \leq 1 - \frac{P_d}{b} (1 + \log_2 b - A) - \frac{H(P_d)}{b} + O(P_d^{3/2-\epsilon}). \end{aligned} \quad (7)$$

Clearly, the right-hand side serves as an upper bound on the capacity for the segmented deletion channel with a finite  $b$  and  $P_d \rightarrow 0$ .

Before proving the given theorems, we present two lemmas whose proofs are given in Appendix A.

**Lemma 1.** For a segmented deletion channel with i.i.d. Bernoulli(1/2) process as the input, we have  $\forall \epsilon > 0$ ,

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^{*,n}) | \mathbf{X}^{*,n}) \\ = \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - A \frac{P_d}{b} + O(P_d^{2-\epsilon}). \end{aligned} \quad (8)$$

**Lemma 2.** For any  $\epsilon > 0$ , there exists  $K < \infty$  and  $P_{d,0} > 0$  such that  $\forall P_d < P_{d,0}$  the following statement holds for the segmented deletion channel. For any positive integer  $L^*$ , if

$\mathbf{X} \in S_{L^*}$ , and  $H(\mathbf{X}) \geq 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$  with  $\gamma > 0$ , then

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^n) | \mathbf{X}^n) \geq \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - A \frac{P_d}{b} - K P_d^{2-\epsilon} (1 + P_d^{1/2} L^*). \quad (9)$$

*Proof of Theorem 1:* Without loss of generality, assume that  $n$  is a multiple of  $b$ . We have  $I(\mathbf{X}^n; \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y} | \mathbf{X}^n)$ . With the i.i.d. Bernoulli(1/2) input process  $\mathbf{X}^{*,n}$ , for the output process  $\mathbf{Y}(\mathbf{X}^{*,n})$ , we obtain

$$\begin{aligned} H(\mathbf{Y}(\mathbf{X}^{*,n})) &= - \sum_{\mathbf{y}} P(\mathbf{y}) \log_2(P(\mathbf{y})) \\ &= - \sum_{m=0}^{n/b} \binom{n/b}{m} (1 - P_d)^{n/b-m} P_d^m \\ &\quad \cdot \log_2 \left( \frac{1}{2^{n-m}} \binom{n/b}{m} (1 - P_d)^{n/b-m} P_d^m \right) \\ &= n \left( 1 - \frac{P_d}{b} \right) + H_T, \end{aligned} \quad (10)$$

where  $\mathbf{y}$  and  $m$  represent the realization of process  $\mathbf{Y}$  and the corresponding total number of deletions in  $\mathbf{y}$ , respectively. The term

$$\begin{aligned} H_T &= \sum_{m=0}^{n/b} \binom{n/b}{m} (1 - P_d)^{n/b-m} P_d^m \\ &\quad \cdot \log_2 \left( \binom{n/b}{m} (1 - P_d)^{n/b-m} P_d^m \right), \\ &= \frac{1}{2} \log_2(2\pi e \frac{n}{b} P_d (1 - P_d)) + o(1) \\ &= O(\log_2 n), \end{aligned}$$

(using Corollary 1 of [26]). The proof then follows by combining the results of  $H(\mathbf{Y}(\mathbf{X}^{*,n}))$  and  $H(\mathbf{Y}(\mathbf{X}^{*,n}) | \mathbf{X}^{*,n})$  given in Lemma 1. ■

*Proof of Theorem 2:* It is clear that for any input  $\mathbf{X}^n$ , the number of deletions is Binomial( $n/b, P_d$ ) distributed, leading to

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^n)) \leq 1 - \frac{P_d}{b}, \quad (11)$$

where the equality is achieved when input sequence is i.i.d. Bernoulli(1/2) distributed. In light of Theorem 1, for i.u.d inputs  $I(\mathbf{X}^{*,n}) > 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$ ,  $\gamma > 0$ , therefore, we only need to consider stationary ergodic processes with  $H(\mathbf{X}) \geq I(\mathbf{X}^{*,n}) > 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$  when computing the upper bounds on the capacity. Combining (11) and Lemma 2, we obtain an upper bound on  $I(\mathbf{X}_{L^*})$  for  $\mathbf{X}_{L^*} \in S_{L^*}$ , which is constructed from  $\mathbf{X}$  by flipping the  $(L^* + 1)$ -th bit in each run with a length longer than  $L^*$ , until no run length exceeds  $L^*$ .

Next, we show that we do not lose much with this restriction for large enough  $L^*$  values. Let  $\mathbf{F}$  be the vector (of the same length as  $\mathbf{Y}(\mathbf{X}^n)$ ) taking values of 1 wherever the corresponding bit in  $\mathbf{Y}(\mathbf{X}_{L^*}^n)$  is flipped and 0 otherwise. From [20] (Eqn. (27) and Eqn. (28)), we have  $|H(\mathbf{Y}(\mathbf{X}^n)) - H(\mathbf{Y}(\mathbf{X}_{L^*}^n))| \leq H(\mathbf{F})$ ,  $|H(\mathbf{Y}(\mathbf{X}^n) | \mathbf{X}^n) - H(\mathbf{Y}(\mathbf{X}_{L^*}^n) | \mathbf{X}_{L^*}^n)| \leq H(\mathbf{F})$  and  $\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{F}) \leq \left(\frac{P_d}{b}\right)^{1/2-\epsilon'} \log_2 L^*/2L^* \forall \epsilon' > 0$ , if

$L^* > \log_2(b/P_d)$ . Therefore, there exists  $\mathbf{X}_{L^*} \in S_{L^*}$  such that

$$|I(\mathbf{X}) - I(\mathbf{X}_{L^*})| \leq \left(\frac{P_d}{b}\right)^{1/2-\epsilon'} \log_2 L^*/L^*, \quad (12)$$

Combining (11), (12), Lemma 2 and taking  $L^* = \lfloor \frac{1}{P_d} \rfloor$ , we get the claim. ■

Theorems 1 and 2 give the asymptotic capacity for an elementary segmented deletion channel with a fixed segment length  $b$  for small  $P_d$  values. For a fixed  $P_d > 0$  and a large segment length  $b$ , we have a different characterization.

**Theorem 3.** For a fixed  $P_d$ , for any  $\epsilon > 0$ , there exists  $b_0 > 0$ , such that  $\forall b > b_0$ , the following statement holds for the segmented deletion channel,

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{X}^{*,n}; \mathbf{Y}(\mathbf{X}^{*,n})) \\ \geq 1 - \frac{P_d}{b} (1 + \log_2 b - A) - \frac{H(P_d)}{b} - O(b^{-2+\epsilon}), \end{aligned} \quad (13)$$

where  $\mathbf{X}^*$  is the Bernoulli(1/2) process, and

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{X}^n; \mathbf{Y}(\mathbf{X}^n)) \\ \leq 1 - \frac{P_d}{b} (1 + \log_2 b - A) - \frac{H(P_d)}{b} + O(b^{-3/2+\epsilon}), \end{aligned} \quad (14)$$

where  $\mathbf{X}$  is any input process.

Before the proof of the theorem, a lemma (whose proof is in Appendix C) is given.

**Lemma 3.** For any stationary ergodic process  $\mathbf{X} \in S_b$  with  $H(\mathbf{X}) \geq 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$ ,  $\gamma > 0$ , and any  $\epsilon > 0$ , there exists  $\kappa < \infty$  and  $b_0 > 0$ , such that  $\forall b > b_0$

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^n) | \mathbf{X}^n) \\ \geq \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - A \frac{P_d}{b} - \kappa b^{-2+\epsilon} (1 + b^{1/2}). \end{aligned} \quad (15)$$

Specifically, consider an i.i.d. Bernoulli(1/2) process  $\mathbf{X}^*$ . By flipping the  $(b + 1)$ -th bit in each run with a length longer than  $b$ , until no run length exceeds  $b$ , we obtain a modified process  $\mathbf{X}_b^* \in S_b$ . We can show that

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}_b^{*,n}) | \mathbf{X}_b^{*,n}) \\ = \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - A \frac{P_d}{b} + O(b^{-2+\epsilon}). \end{aligned} \quad (16)$$

*Proof of Theorem 3:* From (10) and (16), we have

$$I(\mathbf{X}_b^*) = 1 - \frac{P_d}{b} (1 + \log_2 b - A) - \frac{H(P_d)}{b} - O(b^{-2+\epsilon}). \quad (17)$$

As in [20], define  $\alpha = P(L_0 > b)/b$ , which is the upper bound of the density of bits in  $\mathbf{X}^*$  to be flipped to ensure no run length exceeds  $b$ . For an i.i.d. Bernoulli(1/2) process, we have  $\alpha = \frac{1}{b} \sum_{l=b+1}^{\infty} l/2^{l+1} = (1 + \frac{2}{b})2^{-b-1}$ . Therefore,  $\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{F}) \leq H(\alpha) = O(b^{-\zeta})$  with  $\zeta > 2$ , where  $\mathbf{F}$  has the same definition as the one in the proof of Theorem 2. Following the same steps leading to (12), we can write,  $|I(\mathbf{X}^*) - I(\mathbf{X}_b^*)| = O(b^{-\zeta})$ . Combining this result with (17),

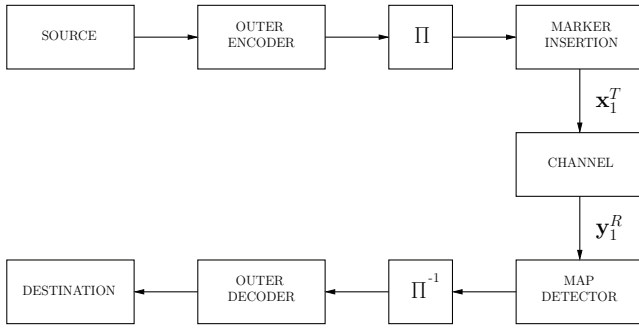


Fig. 1. Block diagram of the considered concatenated scheme.

the lower bound on the capacity given in (13) is proved.

To obtain the upper bound, again, in light of the achievability result, we only consider stationary and ergodic processes with  $H(\mathbf{X}) \geq 1 - (\frac{P_d}{b})^{1-\gamma}$ ,  $\gamma > 0$ . Under this condition, (12) still holds. Taking  $L^* = b$ , we conclude that  $|I(\mathbf{X}) - I(\mathbf{X}_b)| = O(b^{-1.5+\epsilon})$ . Combining this result with (11) and (15) (which provides the upper bound on  $I(\mathbf{X}_b)$  for  $\mathbf{X}_b \in S_b$ ), we get the claim. ■

From the above theorems, we conclude that the channel capacity for segmented deletion channel as  $\frac{P_d}{b} \rightarrow 0$  is dominated by the expression

$$C_{\text{est}} = 1 - \frac{P_d}{b} (1 + \log_2 b - A) - \frac{H(P_d)}{b}, \quad (18)$$

where  $A \approx 1.28853$ .

#### IV. CONCATENATED CODING OVER SEGMENTED DELETION CHANNELS

We now focus on a practical channel coding scheme suitable for segmented deletion channels, as illustrated in Fig. 1.

Our main motivation of choosing this special coding scheme is that such solutions have been shown in recent literature [21] to perform well for other types of synchronization error channels (e.g., i.i.d. deletion/insertion channels). In fact, the use of markers or watermark along with a powerful error correcting code is the state-of-the-art in coding over such channels [27].

The information bits are first encoded by an outer LDPC code, then the transmitted sequence is formed by periodically inserting marker bits to the interleaved sequence of coded bits. Marker bits and their expected positions are known to both of the transmitter and the receiver. Marker code structure and rate optimization is possible for different segmented deletion channels by using a method introduced in [21] for the i.i.d. insertion/deletion channel case. The method utilizes the Monte Carlo simulations to obtain the achievable rates of specific marker codes over different insertion/deletion channels with varying marker code rates. Optimized marker code structure and rate can then be found by after examining different sets of parameters.

Let  $\mathbf{x}_1^T = \{x_k\}_{k=1}^T$  and  $\mathbf{y}_1^R = \{y_n\}_{n=1}^R$  be the sequences of bits at the channel input and channel output, respectively, where the number  $T$  of transmitted bits is a constant system parameter. We assume  $T = Nb$ , where  $N$  is the total number of segments. Since the channel is an elementary segmented

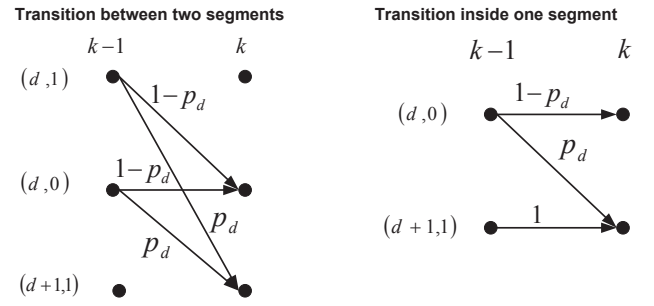


Fig. 2. Trellis section for the bit-level MAP detector.

deletion channel, the number  $R$  of received bits is a random variable taking values in the set  $\{T - N, T - N + 1, \dots, T\}$ , depending on the realization of the deletion process. The transmitter and the receiver have no information on the positions of the deletions.

At the receiver side, MAP detection for synchronization purposes is first executed to generate soft information on the transmitted bits, by exploiting the marker code structure and the particular channel characteristics. Then, after being deinterleaved, this information feeds the outer LDPC decoder. Iterative detection/decoding is performed when the output soft information is feedback to the MAP detector to start a new iteration, according to the turbo principle [28].

In our previous work, MAP detection algorithm was specifically designed for i.i.d. deletion channels [21]. This detector can be directly applied to a segmented deletion channel with a deletion probability for each bit set to  $p_d = P_d/b$ . However, this would be a sub-optimal choice since it ignores the additional information due to the segmentation assumption. For example, if the detector determines that the first bit of a segment is deleted, we can naturally deduce that there will be no error in the next  $b-1$  bits. In the following sections, we describe two other detectors that take the additional segmentation assumption into consideration and provide improved results.

##### A. Improved Bit-Level Synchronization

The MAP detection algorithm is similar to the general forward backward algorithm (FBA) in [28] with some differences. Let us introduce a trellis diagram, as shown in Fig. 2, with the state of trellis at time  $k$  (when  $x_k$  is transmitted) defined to be  $s_k = (d_k, i)$ . The term  $d_k$  denotes the number of deletions up to time  $k$  and  $i$  is an indicator, i.e.,  $i = 0$  when no deletion occurs within the segment up to time  $k$ , and  $i = 1$  otherwise. The transition probability from one state to another state is determined by the bit-wise deletion probability, which is set to be  $p_d = P_d/b$ . When  $x_k$  is not the first bit of a segment, transition for state  $(d, 1)$  to  $(d+1, 1)$  or  $(d+1, 0)$  is prohibited since there has already been one bit deletion in the segment.

Similar to [21], we define the function  $F(x_k, y_n) = \begin{cases} 1 & \text{if } y_n = x_k \\ 0 & \text{if } y_n \neq x_k \end{cases}$ , and also the sets of forward/backward variables in the usual sense,  $\alpha_k(s_k) = P(\mathbf{y}_1^{k-d_k}, s_k)$ ,  $\beta_k(s_k) = P(\mathbf{y}_{k-d_k+1}^R | s_k)$ . These coefficients can be computed by means of the following

forward/backward recursions [29]:

**Case 1:  $x_k$  is the first bit of the segment:**

$$\begin{aligned}\alpha_k(s_k) &= P\left(s_k = (d_k, i), \mathbf{y}_1^{k-d_k}\right) \\ &= ip_d(\alpha_{k-1}(d_k - 1, 1) + \alpha_{k-1}(d_k - 1, 0)) \\ &\quad + (1 - i)(1 - p_d)(\alpha_{k-1}(s_k) + \alpha_{k-1}(d_k, 1)) \\ &\quad \cdot \sum_{x_k} P(x_k)F(x_k, y_{k-d_k}),\end{aligned}\quad (19)$$

$$\begin{aligned}\beta_{k-1}(s_{k-1}) &= P\left(\mathbf{y}_{k-1-d_{k-1}+1}^R | s_{k-1} = (d_{k-1}, i)\right) \\ &= (1 - i)\left(p_d\beta_k(d_{k-1} + 1, 1) + (1 - p_d)\beta_k(s_{k-1})\right) \\ &\quad \cdot \sum_{x_k} P(x_k)F(x_k, y_{k-d_k}) \\ &\quad + i\left((1 - p_d)\beta_k(d_{k-1}, 0) \sum_{x_k} P(x_k)F(x_k, y_{k-d_k})\right. \\ &\quad \left. + p_d\beta_k(d_{k-1} + 1, 1)\right),\end{aligned}\quad (20)$$

**Case 2:  $x_k$  is not the first bit of the segment:**

$$\begin{aligned}\alpha_k(s_k) &= (1 - p_d(1 - i))\alpha_{k-1}(s_k) \sum_{x_k} P(x_k)F(x_k, y_{k-d_k}) \\ &\quad + ip_d\alpha_{k-1}(d_k - 1, 0),\end{aligned}\quad (21)$$

$$\begin{aligned}\beta_{k-1}(s_{k-1}) &= (1 - p_d(1 - i))\beta_k(s_{k-1}) \sum_{x_k} P(x_k)F(x_k, y_{k-d_k}) \\ &\quad + (1 - i)p_d\beta_k(d_{k-1} + 1, 1).\end{aligned}\quad (22)$$

We are interested in the exact ‘‘frame synchronization’’ scenario, leading to

$$\alpha_0(s_k) = \begin{cases} 1 & \text{if } s_k = (0, 0) \\ 0 & \text{otherwise} \end{cases}, \quad (23)$$

$$\beta_T(s_k) = \begin{cases} 1 - P_d & \text{if } s_k = (T - R, 0) \\ P_d & \text{if } s_k = (T - R, 1) \\ 0 & \text{otherwise} \end{cases}. \quad (24)$$

Finally, the target probability can be computed as

**Case 1:**

$$\begin{aligned}P(\mathbf{y}_1^R | x_k) &= (1 - p_d) \sum_{d_k=0}^{\lceil k/b \rceil} \sum_{i=0}^1 \alpha_{k-1}(d_k, i)\beta_k(d_k, 0)F(x_k, y_{k-d_k}) \\ &\quad + p_d \sum_{d_k=0}^{\lceil k/b \rceil} \sum_{i=0}^1 \alpha_{k-1}(d_k - 1, i)\beta_k(d_k, 1),\end{aligned}\quad (25)$$

**Case 2:**

$$\begin{aligned}P(\mathbf{y}_1^R | x_k) &= \sum_{d_k=0}^{\lceil k/b \rceil} \sum_{i=0}^1 (1 - p_d(1 - i))\alpha_{k-1}(d_k, i)\beta_k(d_k, i) \\ &\quad \cdot F(x_k, y_{k-d_k}) + p_d \sum_{d_k=0}^{\lceil k/b \rceil} \alpha_{k-1}(d_k - 1, 0)\beta_k(d_k, 1),\end{aligned}\quad (26)$$

where  $\lceil \cdot \rceil$  indicates the ceiling function.

## B. Symbol-Level Synchronization

As illustrated in [21], the MAP detection algorithm we described in the previous subsection is not optimal. A symbol-

level MAP detector can be applied under this scenario by treating one segment as a symbol.

Let us define the binary event  $D_{k,n}$ , with  $k \in \{1, 2, \dots, N\}$  and  $n \in \{1, 2, \dots, R\}$ , which denotes whether, of the first  $k$  transmitted segments of bits, exactly  $n$  bits are received or not. Thanks to the assumption of 1-deletion per segment, symbol-level MAP detection becomes feasible for large values of  $b$ , and the forward/backward recursions are given as follows:

$$\begin{aligned}\alpha_k(n) &= P(\mathbf{y}_1^n, D_{k,n}) \\ &= P_d \alpha_{k-1}(n - b + 1) \sum_{j=0}^{b-1} \prod_{\substack{i=0 \\ i \neq j}}^{b-1} \sum_{x_{bk-i}} P(x_{bk-i})F(x_{bk-i}, y_{n-i'}) \\ &\quad + (1 - P_d) \alpha_{k-1}(n - b) \prod_{i=0}^{b-1} \sum_{x_{bk-i}} P(x_{bk-i})F(x_{bk-i}, y_{n-i}),\end{aligned}\quad (27)$$

and

$$\begin{aligned}\beta_k(n) &= P(\mathbf{y}_{n+1}^R | D_{k,n}) \\ &= P_d \beta_{k+1}(n + b - 1) \sum_{j=1}^b \prod_{\substack{i=1 \\ i \neq j}}^b \sum_{x_{bk+i}} P(x_{bk+i})F(x_{bk+i}, y_{n+i'}) \\ &\quad + (1 - P_d) \beta_{k+1}(n + b) \prod_{i=1}^b \sum_{x_{bk+i}} P(x_{bk+i})F(x_{bk+i}, y_{n+i}),\end{aligned}\quad (28)$$

respectively, where  $i' = i$  when  $i < j$  and  $i' = i - 1$  when  $i > j$ . The final soft output information is generated as

$$\begin{aligned}p(\mathbf{y}_1^R | x_{b(k-1)+1}, \dots, x_{bk}) &= P_d \sum_{n=0}^{\min(bk, R)} \sum_{j=0}^{b-1} \alpha_{k-1}(n - b + 1)\beta_k(n) \\ &\quad \cdot \prod_{\substack{i=0 \\ i \neq j}}^{b-1} F(x_{bk-i}, y_{n-i'}) \\ &\quad + (1 - P_d) \sum_{n=0}^{\min(bk, R)} \alpha_{k-1}(n - b)\beta_k(n) \\ &\quad \cdot \prod_{i=0}^{b-1} F(x_{bk-i}, y_{n-i}).\end{aligned}\quad (29)$$

Note that both the bit-level and symbol-level synchronization algorithms can be extended to the case of generalized segmented deletion channels. For instance, consider the case where at most two deletion errors are allowed in each segment. For the bit-level synchronization algorithm, the indicator  $i$  should then take values of 0, 1 and 2 and the trellis in Fig. 2 needs to be modified accordingly. For the symbol-level synchronization algorithm, the necessary change is to consider one more state in the FBA algorithm, e.g., add  $\alpha_{k-1}(n - b + 2)$  in the forward recursion.

## C. Computational Complexity Comparisons

For the sake of computational safety, all the calculations of MAP detection algorithm are implemented in the log domain to avoid numerical instability. Therefore, instead of the multiplication operation, the most time-consuming part becomes log domain addition, denoted as  $\log\_add$ . To compare the

complexity of the two algorithms, in this section, we use the number of  $\log\_add$  operations required as a metric.

Consider the symbol-level MAP detection with  $T$  bit inputs and  $R$  bit outputs, the size of the trellis diagram is  $(R+1) \times (N+1)$ , where  $N = T/b$ . For every time instance we only care about  $T - R + 1$  states instead of all the  $R + 1$  states, since the maximum number of bits allowed to be deleted is  $T - R$ . From (27), computation of each forward quantity needs  $2b + 2 \log\_add$  operations. Therefore, there are altogether  $N(T - R + 1)(2b + 2) \log\_add$  operations for the forward recursion as well as for the backward recursion. For the same reason, to generate output soft information in (29), approximately  $2^b N(T - R + 1)(b + 1) \log\_add$  operations are needed for the symbol-level MAP detection.

For the bit-level MAP detection, the size of the trellis diagram is  $2(T - R + 1) \times (T + 1)$ . Computation of each forward quantity needs 2 or 4  $\log\_add$  operations for (19) and 3 or 2 operations for (21), depending on the value of  $i$ . Hence, on average, total number of  $T(T - R + 1)(5b + 1)/b = N(T - R + 1)(5b + 1) \log\_add$  operations are required for the forward recursion. The same result holds for the backward recursion. Following the same line of arguments, approximately  $2T(T - R + 1)(3b + 1)/b = 2N(T - R + 1)(3b + 1) \log\_add$  operations are needed for the bit-level MAP detector to generate output soft information.

It is clear that the number of deletions,  $T - R \sim TP_d/b$ . Therefore, the recursions require similar computational load for both detectors, i.e., the number of  $\log\_add$  operations equals  $O(T^2/b)$ . Difference lies in the generation of the soft information. As expected, complexity of symbol-level MAP detection grows exponentially with  $b$  while the one for the bit-level MAP detector only depends on the length of codeword, i.e.,  $T$ .

V. NUMERICAL EXAMPLES

In this section, we first provide several numerical results of the approximation and upper/lower bounds on the capacity of the elementary segmented deletion channels. Then we give examples comparing synchronization algorithms along with some results on the outer LDPC code design [21].

A. Examples for Capacity Results

In this subsection, some explicit results on the capacity bounds are provided as a function of  $P_d$  and  $b$ . First of all, using BAA, the largest value of  $b$  we can handle for the calculation of  $C_d(b, 1)$  is 24, resulting in  $C_d(24, 1) = 19.65$ , and therefore from (1),  $C \leq 1 - 4.35 \frac{P_d}{b}, \forall b \geq 24$ . The two versions of upper bounds in Section III-B are compared in Table II and Fig. 3. In Table II, we compute the upper bounds in (1) and (2) for the case of  $2 \leq b \leq 15$ . For the second upper bound (2), since we could not obtain exact values of  $C_d(2b, 1)$  when  $b > 12$ , we resort to (4). Fig. 3 compares the capacity upper bound for  $b = 3, 7$  and  $15$ . As expected, the improvement is more obvious as  $b$  decreases. Another observation is that when  $b > 15$ , it makes no sense to use (2), as the bound on  $C_d(2b, 1)$  becomes very loose.

We present  $C_{est}$  in (18) for different segment lengths in Fig. 4. The result illustrates that for the same value of

TABLE II  
CAPACITY UPPER BOUND COMPARISONS FOR  $b \leq 15$ .

$b$	$C \leq$	
	UB (1)	UB (2)
2	$1 - 0.5P_d$	$1 - 0.915P_d + 0.445P_d^2$
3	$1 - 0.510P_d$	$1 - 0.794P_d + 0.284P_d^2$
4	$1 - 0.458P_d$	$1 - 0.694P_d + 0.236P_d^2$
5	$1 - 0.428P_d$	$1 - 0.617P_d + 0.189P_d^2$
6	$1 - 0.397P_d$	$1 - 0.555P_d + 0.158P_d^2$
7	$1 - 0.370P_d$	$1 - 0.507P_d + 0.137P_d^2$
8	$1 - 0.347P_d$	$1 - 0.466P_d + 0.120P_d^2$
9	$1 - 0.326P_d$	$1 - 0.433P_d + 0.107P_d^2$
10	$1 - 0.308P_d$	$1 - 0.405P_d + 0.097P_d^2$
11	$1 - 0.292P_d$	$1 - 0.380P_d + 0.089P_d^2$
12	$1 - 0.277P_d$	$1 - 0.362P_d + 0.085P_d^2$
13	$1 - 0.264P_d$	$1 - 0.344P_d + 0.050P_d^2$
14	$1 - 0.253P_d$	$1 - 0.275P_d + 0.023P_d^2$
15	$1 - 0.242P_d$	$1 - 0.245P_d + 0.001943P_d^2$

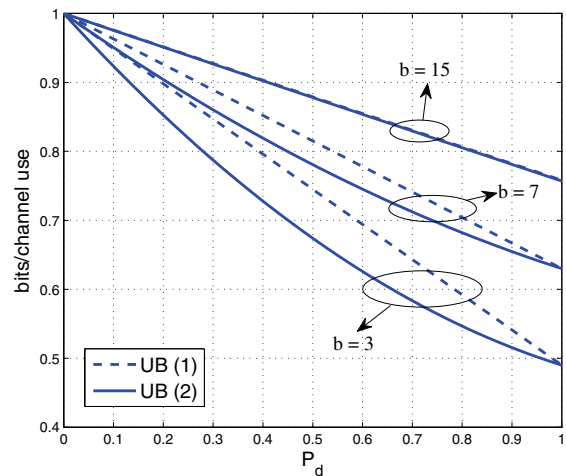


Fig. 3. Capacity upper bound comparisons for  $b = 3, 7, 15$ .

$P_d/b$ , segmented deletion channels with a larger  $b$  offer a higher capacity. Comparison of upper/lower bounds from Section III-B and  $C_{est}$  is provided in Fig. 5 for  $b = 12$ . It is clear that the lower bound remains tight up to around  $P_d = 0.4$  while the upper bound is quite loose. When  $P_d/b = 0.08333$ , i.e.,  $P_d = 1$ , every segment has deletion errors, and the decoupling of different segments is possible without any side information. As discussed before, the upper bound gives the exact value of capacity and  $C_{est}$  exceeds the capacity as given in Table III but it remains close to it. We also observe that both the lower bound and  $C_{est}$  are not monotonically decreasing and there is a “tail” like behavior close to  $P_d = 1$ . It is not a surprising result, as the deletion rate approaches unity, segment-level synchronization becomes less critical and almost every segment has deletion errors. In this case, a higher capacity may be achieved as the synchronization errors become less and less important.



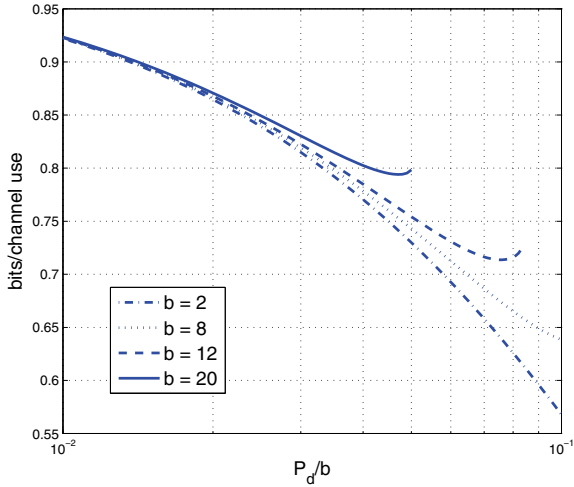


Fig. 4. Estimate of the segmented deletion channel capacity ( $C_{est}$ ) for small  $P_d/b$ .

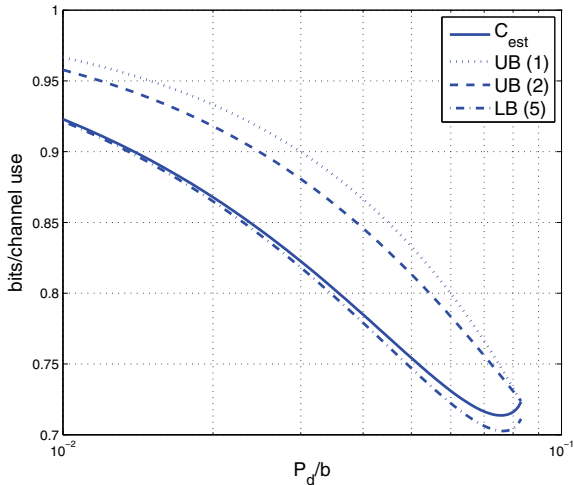


Fig. 5. Comparison of upper and lower bounds on the segmented deletion channel capacity for  $b = 12$ .

### B. Detection/Decoding Results

We first consider practical coding schemes with the aim of confirming the performance gain over the existing techniques. The only reported practical coding scheme is introduced in [6], where for  $b = 8$ , the code rate is 0.448. This code is able to correct all the possible synchronization errors when at most one deletion error occurs per segment. Although codes with higher rates are also provided which allow for some errors for high deletion rates, we will not consider them here, since they assume that some information generated from the transmitted sequence, e.g., parity check bits, is known at the receiver via a perfect side channel.

In Fig. 6, we compare the bit error (BER) performance of several detectors with a single-pass decoding, i.e., MAP detection for synchronization is executed only once. We adopt a binary randomly picked LDPC code with a rate 0.78, length 4521 and insert the marker “01” every 15 LDPC-coded bits. Obviously, the symbol-level MAP detection with iterative soft

TABLE III  
COMPARISON OF CAPACITY BOUNDS.

$P_d$	$b = 3$			$b = 12$		
	LB (5)	$C_{est}$	UB (1)	LB (5)	$C_{est}$	UB (1)
0.001	0.99557	0.99576	0.99949	0.99876	0.99877	0.99972
0.01	0.96688	0.96874	0.99493	0.99039	0.99052	0.99721
0.05	0.87361	0.88292	0.97466	0.96179	0.96239	0.98608
0.1	0.78182	0.80045	0.99972	0.93223	0.93344	0.97217
0.2	0.63566	0.67292	0.89866	0.88247	0.88489	0.94434
0.3	0.52069	0.57659	0.84799	0.84051	0.84414	0.91652
0.5	0.35743	0.45059	0.74665	0.77326	0.77931	0.86086
0.75	0.26572	0.40546	0.61997	0.71728	0.72636	0.79130
1	0.38153	0.56785	0.49330	0.71319	0.72529	0.72173

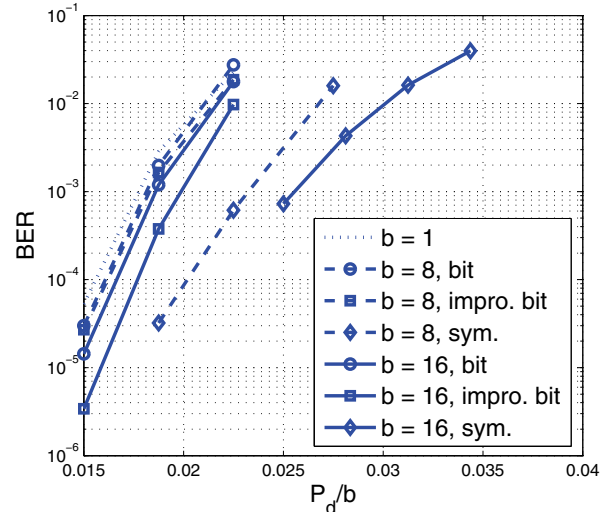


Fig. 6. BER performance with different MAP detectors.

demapping [30] outperforms the other detectors. However for large  $b$ , it becomes infeasible. One solution is to consider only the  $M$  largest soft values among the  $2^b$  outputs as done in greedy multiuser detection algorithm [31]. Another observation is that the bit-level MAP detector for an i.i.d. deletion channel [21] works well at low deletion rates. With the same overall code rate  $R = 0.693$  and under the single-pass decoding assumption, the bit-level MAP detector for an i.i.d. deletion channel provides almost the same performance compared to the one discussed in Section IV-A for  $b = 4$ . This is not a surprising result since the segmentation assumption does not provide additional information to the detector due to the limited number of deletions in this regime. Our final comment is that when the segment length  $b$  is increased for the same average bit deletion probability  $P_d/b$ , the error probability is lower (which is parallel to the findings (e.g., in terms of capacity bound results) in this paper).

### C. LDPC Code Design and Examples

As also discussed in [21], the design of LDPC codes for insertion/deletion channels can rely on utilizing the extrinsic information transfer (EXIT) charts [32] to predict the detection/decoding performance when iterative decoding algorithm is applied. For the MAP detectors described in Section IV, let  $I_V$  and  $I_S$  be the mutual information between the LDPC-coded bits and the corresponding input/output soft values (log-likelihood ratios), respectively. It is shown in [32] that when

TABLE IV  
EXAMPLE LDPC CODE PARAMETERS FOR SEGMENTED DELETION CHANNELS.

	$b$	$r_M$	$r_C$	$d_c$	$d_v$	$a$
$P_d = 0.1$	8	0.9	0.9423	52	{2 3 71}	{0.5667 0.425 0.0083}
$P_d = 0.3$	8	0.8333	0.8636	22	{2 3 42}	{0.5284 0.458 0.0136}
$P_d = 0.5$	8	0.75	0.8	15	{2 3 16}	{0.3936 0.576 0.0304}
$P_d = 0.7$	8	0.7143	0.75	12	{2 3 14}	{0.3354 0.634 0.0306}
$P_d = 0.9$	8	0.7143	0.7	10	{2 3 12}	{0.4301 0.522 0.0479}
$P_d = 0.2$	16	0.9	0.9444	54	{2 3 64}	{0.6385 0.351 0.0105}
$P_d = 0.4$	16	0.8571	0.9062	32	{2 3 31}	{0.5493 0.431 0.0197}
$P_d = 0.6$	16	0.8	0.875	24	{2 3 16}	{0.4206 0.547 0.0324}
$P_d = 0.8$	16	0.7778	0.8421	19	{2 3 14}	{0.395 0.569 0.036}
$P_d = 1$	16	0.75	0.8	15	{2 3 14}	{0.3318 0.638 0.0302}

the detection EXIT chart, which describes the relationship between output  $I_S$  and input  $I_V$ , is non-flat, i.e., each received symbol depends on multiple transmitted symbols, LDPC code design for this case is beneficial. For the segmented deletion channel, since it is not memoryless, instead of using randomly picked LDPC codes as in Fig. 6 or the ones optimized for the AWGN channels (with a flat detection EXIT chart), specially designed LDPC codes can provide a better performance.

Consider a check-regular LDPC code with constant check node degree  $d_c$ . Let  $I$  be the total number of different variable node degrees of the LDPC code denoted by  $d_{v,i}$ ,  $i = 1, \dots, I$  and  $a_i$  be the fraction of variable nodes with degree  $d_{v,i}$ . The goal of code design for a fixed code rate  $r_C$  is to find the set of parameters  $a_i$ ,  $d_{v,i}$  and  $d_c$  which provide the best detection/decoding performance.

Some optimized codes are listed in Table IV with an average variable node degree of [32]  $\bar{d}_v = 3$  and  $r_M$  is the optimized 2-bit marker code rate obtained using a similar approach as in [21]. In Fig. 7, the highest achievable code rates for the concatenated coding scheme are plotted as a function of  $P_d$  for  $b = 8$ . The solid line denotes the achievable rates when LDPC codes from Table IV are used while the dashed line represents the case for codes optimized for the AWGN channels. As the deletion rate increases, the achievable rate drops from 0.84 to 0.446 bits/channel use. Compared to the codes in [6], we can always achieve a higher code rate for  $P_d < 1$  due to the more sophisticated detector/decoder configuration and possibility of arbitrarily low error probabilities (instead of no-errors). Also included in the figure is the capacity lower bound in (5) which shows that even though significantly improved code rates are obtained by the specific designs over previously known codes optimized for AWGN channels and over codes in [6], there is room for further improvement.

To further illustrate the advantage of the designed codes, we pick several codes from Table IV, each of length 10000, and depict their error rate performance in Fig. 8 using the bit-level synchronization algorithm over the segmented deletion channel. Again, performance of LDPC codes optimized for the AWGN channels are given in dashed lines (of the same rate but different variable/check node distributions). Parameters of Code 1, 2 and 3 for the segmented deletion channel are given in the second, third and fourth row of Table IV, and their overall code rates are 0.719, 0.6 and 0.5337, respectively. It is obvious that the specifically designed outer LDPC codes for the segmented deletion channel offer a better performance. We also observe that the concatenated coding scheme can achieve

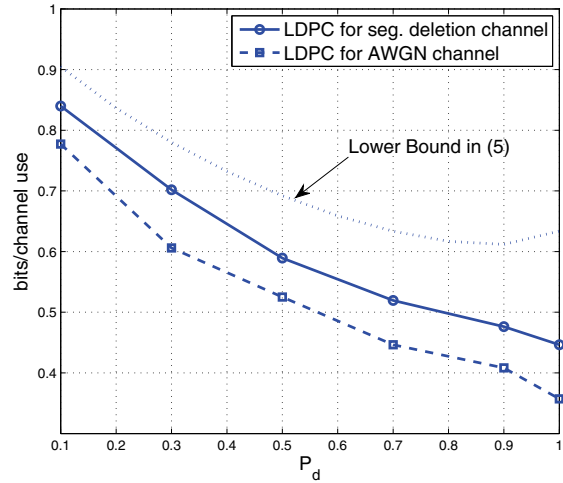


Fig. 7. Achievable rates as a function of  $P_d$  for different choices of outer LDPC codes.

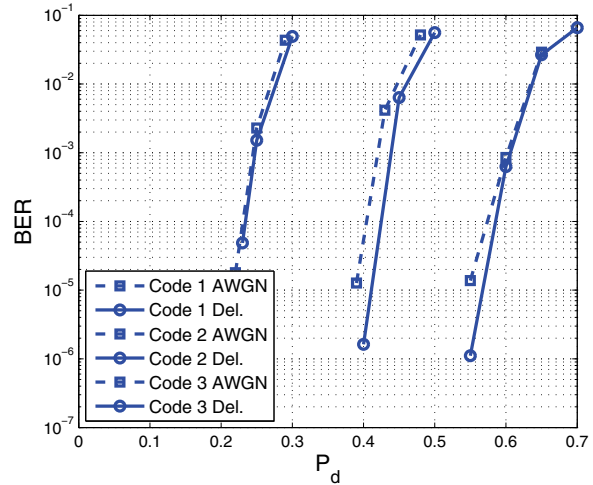


Fig. 8. BER for several LDPC codes over segmented deletion channels.

a higher code rate when  $\frac{P_d}{b}$  gets smaller. We note, however, that the results obtained are not very close to the capacity bounds. For instance, if we consider an error rate of  $10^{-3}$  as reliable communications, from Fig. 8, the corresponding  $P_d$  for these three codes are 0.24, 0.44 and 0.6, while the corresponding capacity lower bounds are 0.8127, 0.7152 and 0.6589, respectively. A difference of 0.1 bits/channel use exist between the capacity lower bounds and the actual achieved code rates with the practical channel coding approach, which as also previously stated, indicates that there is certainly room for significant improvement with more sophisticated practical coding solutions.

## VI. SUMMARY AND CONCLUSIONS

In this paper, we have considered channels with synchronization errors modeled by a bit deletion process with an additional segmentation assumption. We started with the argument that such channels are information stable, and their channel capacity exists. Then, we introduced several capacity upper and lower bounds in an attempt to understand the channel

capacity behavior. The results indicate that when the deletion probability is near zero or near unity (for each segment), the upper and lower bounds behave similarly and the obtained results are very close to the actual channel capacity. However, there is a wide-range of deletion probabilities where they are far apart, hence there is clearly more room for improvement (in terms of obtaining tighter capacity bounds). In addition to the information theoretic analysis of the channel, we have also considered a practical channel coding approach. Specifically, we used outer LDPC codes concatenated with inner marker codes, and developed suitable channel detection algorithms for this case. Different MAP based channel synchronization algorithms operating at the bit level and at the symbol level were introduced. Furthermore, we have compared complexity of the two algorithms and designed specific LDPC codes for segmented deletion channels which provide better decoding performance than the ones optimized for AWGN channels. Simulation results clearly show the advantages of the proposed approach. In particular, for the entire range of deletion probabilities less than unity, the proposed approach offers a significantly larger transmission rate than the only other alternative solution of the zero-error codes designed in [6].

#### APPENDIX A

##### PROOF OF LEMMA 1

*Proof of Lemma 1:* Define  $\mathbf{D}^n$  to be an  $n$ -bit vector that contains a 1 if and only if the corresponding bit in  $\mathbf{X}^n$  is deleted. We have  $H(\mathbf{D}^n) = \frac{n}{b}(P_d \log_2 b + H(P_d))$ . With this definition, the random process  $\mathbf{D}$  is non-stationary even though  $\mathbf{X}$  is stationary and ergodic. In order to make it stationary, we let the “first” segment of the channel start at a random position which is uniformly chosen from  $\{1, 2, \dots, b\}$ , which does not affect the capacity. It is easy to deduce that  $H(\mathbf{Y}|\mathbf{X}^n) = H(\mathbf{D}^n) - H(\mathbf{D}^n|\mathbf{X}^n, \mathbf{Y})$ . The exact evaluation of the term  $H(\mathbf{D}^n|\mathbf{X}^n, \mathbf{Y})$  is troublesome; however, under the condition that  $P_d/b$  is small, it can be bounded.

The following arguments follow similar steps as in [20], which considers the case of i.i.d. deletions. Let  $\hat{\mathbf{D}}^n$  be the vector obtained by flipping “1”s in  $\mathbf{D}^n$  for two cases. First, when a particular run experiences deletion errors, which is referred to as the error run, and the number of deletions exceeds one, we flip all 1s in  $\mathbf{D}^n$  which are associated with that error run. Secondly, when different error runs span the same segment, we flip all 1s in  $\mathbf{D}^n$  which are associated with these error runs. One example is given as follows. Suppose we transmit a sequence 001 000 001 110 over a segmented deletion channel with  $b = 3$ , and receive 01000110. Obviously, one bit gets deleted from each segment resulting in a total number of 24 possible realizations of  $\mathbf{D}$  (one of the two 0’s gets deleted from the first segment, one of the three 0’s gets deleted from the second segment, one of the two 0’s gets deleted from the third segment, and one of the two 1’s gets deleted from the last segment). Since the third bit run (five consecutive 0’s) have two deletion errors and the fourth bit run with only one error but share the same segment with another error run, we assume an auxiliary channel that generates 01000001110 and the corresponding  $\hat{\mathbf{D}}$  can only be either 100 000 000 000 or 010 000 000 000 with equal probability. By doing so, we guarantee that every deletion

error from this auxiliary channel belongs to a bit run with a single deletion and every bit from that run can be deleted with an equal probability.

The process  $\hat{\mathbf{D}} = f(\mathbf{D}, \mathbf{X})$  is also stationary with  $P(\hat{D}_i = 1)$  being upper bounded by  $P_d/b$ . A lower bound on  $P(\hat{D}_i = 1)$  can be obtained as follows. Let  $l_0$  be the length of a bit run which contains  $X_i$  and spans  $(j - m_1)$ -th to  $(j + m_2)$ -th segments. When  $\hat{D}_i = 1$ , the  $(j - m_1)$ -th to  $(j + m_2)$ -th segments will not experience deletion error except for the  $j$ -th segment, to which  $X_i$  belongs. Also, any bit from a run which starts from the  $(j + m_2)$ -th segment or ends in the  $(j - m_1)$ -th segment will not be deleted. Let  $l_1$  and  $l_2$  be the lengths of the run which ends in the  $(j - m_1)$ -th segment and the one which starts from the  $(j + m_2)$ -th segment, respectively. There is only one deletion error in these segments and it has to be in the  $j$ -th segment. Therefore, considering the worst case scenario and denoting the joint probability mass function of pairs of run lengths with  $P_L(\cdot, \cdot)$ , we have,

$$\begin{aligned} P(\hat{D}_i = 1) &\geq \sum_{l_1, l_2=1}^{\infty} \frac{P_d}{b} (1 - P_d)^{l_0 + l_1 + l_2 + 4(b-1)} P_L(l_1, l_2) \\ &\geq \frac{P_d}{b} - (l_0 + E[l_1] + E[l_2] + 4(b-1))P_d^2. \end{aligned} \quad (30)$$

For any input process with a finite average run length, we can write  $P(\hat{D}_i = 1) \in (P_d/b - K^*l_0P_d^2, P_d/b)$ , where  $K^* < \infty$  is a nonnegative integer.

With the above introduction of  $\hat{\mathbf{D}}$  and letting  $\hat{\mathbf{Y}}$  to be the outcome of  $\mathbf{X}^n$  corresponding to the deletion pattern  $\hat{\mathbf{D}}^n$ , it is clear that runs with length  $l = 1$  do not contribute to  $H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}})$ . Furthermore, no run with more than one deletion can contribute to  $H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}})$  as they all have been reversed. Therefore, only runs with length  $l \geq 2$  and one deletion lead to a contribution of  $\log_2 l$  to  $H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}})$  since the deleted bit is uniformly chosen, which is guaranteed by the definition of  $\hat{\mathbf{D}}$  and the channel model. Finally, we conclude that  $H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}}) = \sum_{r \in \mathbb{R}} \log_2(l_r)$ , where  $\mathbb{R}$  is the set of runs on which deletions occur and  $l_r$  is the corresponding run length. Therefore, from [20], let  $L_0$  be the bit perspective run length of the input sequence (for an arbitrary transmitted bit, the length of the run it belongs to), for any stationary ergodic process such that  $E[L_0 \log_2 L_0] < \infty$ , we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}}) = \frac{P_d}{b} E[\log_2 L_0] - \delta, \quad (31)$$

where  $0 \leq \delta \leq K^*P_d^2 E[L_0 \log_2 L_0]$ .

Define  $\mathbf{Z} = \mathbf{D} \oplus \hat{\mathbf{D}}$ , which represents the difference between  $\mathbf{D}$  and  $\hat{\mathbf{D}}$ . The process  $\mathbf{Z}$  is stationary with  $z = P(\mathbf{Z}_i = 1) \leq K^*E[L_0]P_d^2$ . Note that  $(\mathbf{X}^n, \hat{\mathbf{Y}}, \hat{\mathbf{D}}^n)$  is a function of  $(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n, \mathbf{Z}^n)$ , we have  $|H(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n) - H(\mathbf{X}^n, \hat{\mathbf{Y}}, \hat{\mathbf{D}}^n)| = |H(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n) - H(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n, \mathbf{Z}^n)| = H(\mathbf{Z}^n|\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n) \leq H(\mathbf{Z}^n)$ . Same argument also holds for  $|H(\mathbf{X}^n, \mathbf{Y}) - H(\mathbf{X}^n, \hat{\mathbf{Y}})|$ . Therefore from [20],  $|H(\mathbf{D}^n|\mathbf{X}^n, \mathbf{Y}) - H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}})| \leq 2H(\mathbf{Z}^n) \leq 2nH(z)$ . Hence, the following equation follows,

$$H(\mathbf{Y}|\mathbf{X}^n) = H(\mathbf{D}^n) - H(\hat{\mathbf{D}}^n|\mathbf{X}^n, \hat{\mathbf{Y}}) + n\delta', \quad (32)$$

where  $-2H(z) \leq \delta' \leq \delta + 2H(z)$ . Combining (31) and (32), we obtain

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}|\mathbf{X}^n) = \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - \frac{P_d}{b} E[\log_2 L_0] + \delta'. \quad (33)$$

For the input process  $\mathbf{X}^*$ , it is easy to verify that  $E[L_0 \log_2 L_0] < \infty$ . In this case,  $z = O(P_d^2)$ , and therefore,  $\delta' = O(P_d^{2-\epsilon})$  for any  $\epsilon > 0$ . Hence, from (33), the lemma is proved. ■

#### APPENDIX B PROOF OF LEMMA 2

*Proof of Lemma 2:* Lemma 2 provides a lower bound on  $\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}|\mathbf{X}^n)$ . Based on the result given in (33), the only work is to quantify the lower bounds on  $\delta'$  and  $E[\log_2 L_0]$  for any stationary ergodic process.

First of all, (32) states that  $\delta' \geq -2H(z)$ . From the proof of Lemma 1, we have  $z = P(\mathbf{Z}_i = 1) \leq K^* E[L_0] P_d^2$ . According to [20] (Lemma IV.3), for any stationary ergodic process satisfying the condition  $H(\mathbf{X}) > 1 - (\frac{P_d}{b})^{1-\gamma}$  ( $\gamma > 0$ ), the mean of the bit perspective run length  $E[L_0] \leq K'(1 + (\frac{P_d}{b})^{1/2-\epsilon'} L^*)$ ,  $K' < \infty$  for any integer  $L^*$ . Combining the upper bound on  $z$  and  $E[L_0]$ , we conclude that  $H(z) \leq K'' P_d^{2-\epsilon}(1 + P_d^{1/2} L^*) \forall P_d < P_{d,0}$  and consequently  $\delta' \geq -K' P_d^{2-\epsilon}(1 + P_d^{1/2} L^*)$  [20], where  $K' < \infty$  is a positive integer. Also from [20] (Lemma IV.3), we have  $|A - E[\log_2 L_0]| = O(P_d^{1/2-\epsilon} \log_2 L^*)$ . Combining these results with (33), the lemma is proved. ■

#### APPENDIX C PROOF OF LEMMA 3

*Proof of Lemma 3:* In this case, define  $\hat{\mathbf{D}}^n$  to be generated by flipping the ones in  $\mathbf{D}^n$  when the corresponding error run spans two segments, which is different from the one defined in the proof of Lemma 1. In order to obtain a stationary process  $\hat{\mathbf{D}}$ , we still let the first segment of the input process start at a random position which is uniformly chosen from  $\{1, 2, \dots, b\}$ .

For any stationary and ergodic process  $\mathbf{X}$ , the starting point of a bit run is uniformly distributed within the segment<sup>2</sup>. Also, since the positions of the segment boundaries are random with a uniform distribution, the probability that the error run with length  $l_0$  spans two segments is  $\frac{l_0-1}{b}$ , if we restrict the input process  $\mathbf{X} \in S_b$ , i.e.,  $l_0 \leq b$ . Therefore, it is clear that  $P(\hat{\mathbf{D}}_i = 1) = \frac{P_d}{b}(1 - \frac{l_0-1}{b})$ . Also, with the same definition of  $\mathbf{Z}$  as in the proof of Lemma 1, we have  $z = P(\mathbf{Z}_i = 1) \leq b^{-2} E[L_0]$ . Following the same steps of the proof in Lemma 1, we have, for any stationary ergodic process  $\mathbf{X} \in S_b$  such that  $E[L_0 \log_2 L_0] < \infty$ ,

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}) = \frac{P_d}{b} E[\log_2 L_0] - \frac{P_d}{b^2} E[(L_0 - 1) \log_2 L_0]. \quad (34)$$

<sup>2</sup>To see this, let us first consider the case of  $b = 2$  and suppose that the bit run starts at the first bit of the segment with probability  $p_1$  and at the last bit of the segment with probability  $p_2$ . Clearly,  $p_1 = p_1 \cdot p_{\text{even}} + p_2 \cdot p_{\text{odd}}$ , where  $p_{\text{even}}$  and  $p_{\text{odd}}$  are the probabilities of the run length being an even or odd number, respectively. Since  $p_{\text{even}} = 1 - p_{\text{odd}}$ , we have  $p_1 = p_2 = 0.5$ . Extension to the general case is straightforward and the detailed proof is omitted.

Substituting (34) into (32), the following result appears under the same condition,

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{Y}|\mathbf{X}^n) = \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - \frac{P_d}{b} E[\log_2 L_0] + \delta', \quad (35)$$

where  $-2H(z) \leq \delta' \leq \delta + 2H(z)$ ,  $\delta = b^{-2} E[L_0 \log_2 L_0]$ .

For the process  $\mathbf{X}_b^* \in S_b$ ,  $z \leq b^{-2} E[L_0] = O(b^{-2})$ , and therefore,  $H(z) = O(b^{-2+\epsilon})$ . Since  $-2H(z) \leq \delta' \leq b^{-2} E[L_0 \log_2 L_0] + 2H(z)$  and it is easy to verify in this case  $E[L_0 \log_2 L_0] < \infty$ , we conclude that  $\delta' = O(b^{-2+\epsilon})$  for any  $\epsilon > 0$ . Hence, (16) is proved.

To show (15), we follow the same rationale in the proof of Lemma 2. Since  $z \leq b^{-2} E[L_0]$  and for any stationary ergodic process satisfying the condition  $H(\mathbf{X}) > 1 - (\frac{P_d}{b})^{1-\gamma}$  ( $\gamma > 0$ ),  $E[L_0] \leq \kappa'(1 + (\frac{P_d}{b})^{1/2-\epsilon'} b)$  (let  $L^* = b$ ), we get  $H(z) \leq \kappa^* b^{-2+\epsilon}(1 + b^{1/2}) \forall b > b_0$ . Using the conclusion that  $|A - E[\log_2 L_0]| = O(b^{-1/2+\epsilon})$  [20] (Lemma IV.3), the result follows. ■

#### REFERENCES

- [1] R. L. White, R. M. H. New, and R. F. W. Pease, "Patterned media: a viable route to 50 Gbit/in<sup>2</sup> and up for magnetic recording?" *IEEE Trans. Magn.*, vol. 33, no. 1, pp. 990–995, Jan. 1997.
- [2] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, pp. 1–33, June 2009.
- [3] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors," MIT Lincoln Lab., Tech. Rep., Oct. 1961.
- [4] M. C. Davey and D. J. Mackay, "Reliable communication over channels with insertions, deletions and substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.
- [5] M. Mitzenmacher, "Capacity bounds for sticky channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 72–77, Jan. 2008.
- [6] Z. Liu and M. Mitzenmacher, "Codes for deletions and insertion channels with segmented errors," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 224–232, Jan. 2010.
- [7] A. Mazumdar and A. Barg, "Channels with intermittent errors," in *Proc. 2011 IEEE International Symp. Inf. Theory*, pp. 1753–1757.
- [8] L. Dolecek and V. Anantharam, "On communication over channels with varying sampling rate," in *2007 Inf. Theory Appl. Workshops (at UCSD)*.
- [9] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1226–1237, Mar. 2006.
- [10] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of deletion channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4648–4657, Oct. 2006.
- [11] —, "Improved lower bounds for the capacity of i.i.d. deletion and duplication channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2693–2714, Aug. 2007.
- [12] E. Drinea and A. Kirsch, "Directly lower bounding the information capacity for channels with i.i.d. deletions and duplications," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 86–102, Jan. 2010.
- [13] D. Fertonani and T. M. Duman, "Novel bounds on the capacity of the binary deletion channel," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2753–2765, June 2010.
- [14] A. R. Iyengar, P. H. Siegel, and J. K. Wolf, "Modeling and information rates for synchronization error channels," in *Proc. 2011 IEEE International Symp. Inf. Theory*, pp. 380–384.
- [15] T. G. Swart and H. C. Ferreira, "Insertion/deletion correcting coding schemes based on convolution coding," *IEEE Electron. Lett.*, vol. 38, no. 16, pp. 871–873, Aug. 2002.
- [16] N. J. A. Sloane, "On single-deletion-correcting codes," in *Codes Designs: Proc. Conf. Honoring Professor Dijen K. Ray-Chaudhuri Occasion His 65th Birthday*, pp. 273–291.
- [17] L. Cheng and H. Ferreira, "Rate-compatible path-pruned convolutional codes and their applications on channels with insertion, deletion and substitution errors," in *Proc. 2005 IEEE Inf. Theory Workshop*, pp. 20–25.
- [18] L. McAven and R. Safavi-Naini, "Classification of the deletion correcting capabilities of Reed-Solomon codes of dimension 2 over prime fields," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2280–2294, June 2007.

- [19] R. L. Dobrushin, "Shannon's theorems for channels with synchronization errors," *Problems Inf. Transmission*, vol. 3, no. 4, pp. 11–26, 1967.
- [20] Y. Kanoria and A. Montanari, "On the deletion channel with small deletion probability," in *Proc. 2010 IEEE International Symp. Inf. Theory*, pp. 1002–1006.
- [21] F. Wang, D. Fertonani, and T. M. Duman, "Symbol-level synchronization and LDPC code design for insertion/deletion channels," *IEEE Trans. Commun.*, vol. 59, no. 5, pp. 1287–1297, May 2011.
- [22] T. T. Kadota, "On the information stability of stationary ergodic processes," *SIAM J. Appl. Mathematics*, vol. 26, no. 1, pp. 176–182, Jan. 1974.
- [23] S. Verdú and T. Weissman, "The information lost in erasures," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5030–5058, Nov. 2008.
- [24] S. Arimoto, "An algorithm for calculating the capacity of an arbitrary discrete memoryless channel," *IEEE Trans. Inf. Theory*, vol. 18, pp. 14–20, Jan. 1972.
- [25] R. E. Blahut, "Computation of channel capacity and rate distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, pp. 460–473, Jan. 1972.
- [26] P. Jacquet and W. Szpankowski, "Entropy computations via analytic depoissonization," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1072–1081, May 1999.
- [27] H. Mercier, V. Bhargava, and V. Tarokh, "A survey of error-correcting codes for channels with symbol synchronization errors," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 1, pp. 87–96, First Quarter 2010.
- [28] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [29] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Inf. Theory*, vol. 21, pp. 404–411, July 1975.
- [30] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, Nov. 1997.
- [31] A. A. AlRustamani, A. D. Damnjanovic, and B. R. Vojcic, "Turbo greedy multiuser detection," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 8, pp. 1638–1645, Aug. 2001.
- [32] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.



**Feng Wang** received his B.S. degree from Southeast University, Nanjing China, in 2005, the M.S. degree from Michigan Technological University, Houghton, in 2007, and Ph.D. degree from Arizona State University, Tempe, in 2012, respectively, all in electrical engineering. He is currently with LitePoint Corp., Sunnyvale, CA. His research interests lie in the areas of digital communications, particularly coding and detection techniques, with application to digital storage systems and wireless communication systems.



**Tolga M. Duman** (S'95-M'98-SM'03-F'11): Tolga M. Duman is a Professor of Electrical and Electronics Engineering Department at Bilkent University in Turkey, and is on leave from the School of ECEE at Arizona State University. He received the B.S. degree from Bilkent University in Turkey in 1993, M.S. and Ph.D. degrees from Northeastern University, Boston, in 1995 and 1998, respectively, all in electrical engineering. Prior to joining Bilkent University in September 2012, he has been with the Electrical Engineering Department of Arizona State

University first as an Assistant Professor (1998–2004), then as an Associate Professor (2004–2008), and starting August 2008 as a Professor. Dr. Duman's current research interests are in systems, with particular focus on communication and signal processing, including wireless and mobile communications, coding/modulation, coding for wireless communications, data storage systems and underwater acoustic communications.

Dr. Duman is a Fellow of IEEE, a recipient of the National Science Foundation CAREER Award and IEEE Third Millennium medal. His publications include a book on MIMO Communications (by Wiley in 2007), over 50 journal papers and over 100 conference papers. He served as an editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2003–08), IEEE TRANSACTIONS ON COMMUNICATIONS (2007–2012) and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (2002–07). He is currently the coding and communication theory area editor for IEEE TRANSACTIONS ON COMMUNICATIONS (2011–present) and an editor for *Elsevier Physical Communications Journal* (2010–present).



**Defne Aktas** (S'96, M'03) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1996, and the M.S. and the Ph.D. degrees from The Ohio State University, Columbus, in 1998 and 2002, respectively, all in electrical engineering. She is currently an Assistant Professor in the Department of Electrical and Electronics Engineering at Bilkent University, Ankara, Turkey. Before joining Bilkent University, she was a Research Fellow at The University of Melbourne, Melbourne, Australia and a Postdoctoral Researcher at The Ohio State

University. Dr. Aktas is a recipient of the European Commission Marie Curie Fellowship. Her research interests are in physical layer aspects of wireless communication systems, with emphasis on coding and information theoretic analysis of multiple input multiple output (MIMO) systems.