# Capacity Upper Bounds for the Deletion Channel

Suhas Diggavi[*], Michael Mitzenmacher[†], and Henry D. Pfister[‡]

[*]School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland
Email: `suhas.diggavi@epfl.ch`

[†]Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA
Email: `michaelm@eecs.harvard.edu`

[‡]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX
Email: `hpfister@tamu.edu`

*Abstract*— We present two upper bounds on the capacity of the i.i.d. binary deletion channel, where each bit is independently deleted with a fixed probability $d$. The first can be numerically evaluated for any fixed $d$. The second provides an asymptotic upper bound as $d$ goes to 1. These appear to be the first non-trivial upper bounds for this *probabilistic* deletion channel.

## I. INTRODUCTION

In this paper, we consider upper bounds on the capacity of the i.i.d. binary deletion channel, where each bit is independently deleted with a fixed probability $d$. The i.i.d. binary deletion channel (henceforth also called the BDC) is perhaps the simplest stochastic model of a channel with synchronization errors. For this channel, there is still no single-letter characterization of the achievable rate.

There has been significant recent progress in improving the capacity lower bounds for the BDC [1]–[7], demonstrating that the capacity is much higher than previously known. But other than the trivial $1 - d$ upper bound given by the corresponding erasure channel, no non-trivial upper bound appears available. In order to close the gap between the known upper and lower bounds on the capacity, we would like to have good upper bounds for the BDC as well.

Our upper bounds are all determined by considering a genie-aided decoder with access to side information about the deletion process. For example, in this framework the erasure channel corresponds to a decoder with access to the *entire* deletion process and gives a trivial upper bound. We consider two regimes. We first consider general bounds that hold for any $d$. Focusing on binary deletion channels, we define *runs* as a set of contiguous zeros (or ones). In Section III, we consider the case when the decoder knows when an entire run has been deleted from the input codeword. This allows us to reformulate the capacity of such a channel in terms of a capacity per unit cost of a new channel based on run lengths.

In Section IV, we also develop bounds for the asymptotic behavior as $d$ approaches 1. Recently, Mitzenmacher and Drinea established a general lower bound for the capacity of the BDC of $c_1(1 - d)$ for $c_1 = 0.1185 > 1/9$ [7]. Motivated by this, we looked for an upper bound of the form $c_2(1 - d)$,

with $c_2 < 1$, in the limit as $d \to 1$. Such a bound is obtained with $c_2 = 0.7918$ for this limiting case.

### A. Previous work

The insertion/deletion/substitution channel was introduced by Levenshtein [8] and the information coding theorem was established by Dobrushin in [9]. The line of work initiated by Levenstein led to combinatorial code constructions for insertion/deletion channels; see for example [10] for a recent survey of such results. Most early work focused on the case when there were a *finite* number of synchronization errors.

For i.i.d. deletion errors, no single-letter characterization of the achievable rate has been found, the main difficulty being that such a channel is not memoryless. When attention is restricted to i.i.d. input codebooks, a single-letter form for a lower bound on the achievable rate was established for binary input i.i.d. insertion/deletion/substitution channels by Gallager [11]. This idea was extended to a slightly more general model for insertion errors in [12]. Ullman [13] also studied this problem in a combinatorial context, establishing achievable rates and some upper bounds. His work does not focus on probabilistic models, but on achieving correct decoding without error for a given fraction of synchronization errors, and as such his bounds do not hold for the channels considered here. More recently, improved lower bounds on the capacity of the BDC based on improved analysis techniques were introduced by Diggavi and Grossglauser [1], [2]. These bounds were further improved by Drinea and Mitzenmacher [5], [6], [14]. Kavčić and Motwani also computed simulation based lower bounds based on a suboptimal decoder [3]. Some of these bounds can be extended to additional channel models, including channels where bits can be duplicated as well as deleted. Recent work by Mitzenmacher has focused on the case where bits can *only* be duplicated, with such channels having been dubbed *sticky channels* [15]; the analysis of sticky channels utilizes many of the ideas developed in Section III. Despite the recent improvements in lower bounds for the BDC, there has been no significant improvements in upper bounds of which we are aware.

## II. Preliminaries

Sets are denoted by calligraphic letters: $\mathcal{A}, \mathcal{X}, \mathcal{Y}$. Random variables are denoted by capital letters: $X, Y, Z$. The distribution of a random variable is denoted $p_X(x)$. The set of finite strings of positive integers is $\mathcal{A} = \{1, 2, \ldots\}^*$. The set of finite strings of non-negative integers is $\mathcal{A}_0 = \{0, 1, \ldots\}^*$. The set of finite strings of bits is $\mathcal{B} = \{0, 1\}^*$. Complete strings are denoted by underlines $\underline{X}, \underline{x}$. Single elements of a string are denoted by a subscript index $X_i, x_i$. The interval substrings (i.e., $X_i, X_{i+1}, \ldots, X_j$) of a string are given by $X_i^j, x_i^j$. The length of a string is $l(\underline{a})$. The weight of a string $w(\underline{a})$ is the sum of all entries $\sum_{i=1}^{l(\underline{a})} a_i$.

The *i.i.d. binary deletion channel* (BDC) takes $n$ transmitted bits, and outputs a random subsequence of the input where the subsequence is obtained by deleting each bit independently with probability $d$. The difference between the BDC and the erasure channel is that the erasure channel has knowledge of which subsequence was received whereas the BDC does not. Therefore, the set of possible transmissions is $\{0, 1\}^n$ and the set of possible receptions is $\cup_{m=0}^n \{0, 1\}^m$, where $m \in \{0, \ldots, n\}$ denotes the length of the received string.

Run-length notation parses each binary string into runs of zeros and ones and produces a string of positive integers giving the run lengths (e.g., 001000111 is mapped to $2, 1, 3, 3$). If $n$ bits are transmitted, then the set of possible transmissions is $\{\underline{a} \in \mathcal{A} | w(\underline{a}) = n\}$ and the set of possible receptions is $\{\underline{a} \in \mathcal{A} | w(\underline{a}) \leq n\}$. It is worth noting that run-length notation does not distinguish between complementary strings (e.g., 00101 and 11010), but this is asymptotically irrelevant and ignored henceforth.

Let us define the *run-length projection* $RL(\cdot)$ of a binary string $\underline{b} \in \mathcal{B}$ as the positive-integer string $\underline{a} \in \mathcal{A}$ where $a_i$ is the length of the $i$th run in $\underline{b}$. Let $\underline{V}$ be an input string to a BDC and $\underline{W}$ be the output string. Let $\underline{X} = RL(\underline{V})$ and $\underline{\tilde{Y}} = RL(\underline{W})$ be the run-length projections of $\underline{V}, \underline{W}$. The *run-length channel* (RLC) is defined by transition probabilities $p_{\underline{\tilde{Y}}|\underline{X}}(\underline{y}|\underline{x})$. Note that converting to run-length notation does not immediately make the problem easier; if the length of any run in the input string is reduced to zero, the the adjacent runs are combined in the output, and this is reflected in the RLC. For example, the runs 3,2,3 (corresponding to binary input 00011000) could result in an output of 4 (if the middle four bits were deleted, yielding 0000).

The following simple observation (given without proof) allow us to relate the RLC and the BDC.

*Lemma 1:* The mutual information of the run-length channel (RLC) and the binary deletion channel (BDC) are related by

$$I\left(\underline{X}; \underline{\tilde{Y}}\right) - 1 \leq I\left(\underline{V}; \underline{W}\right) \leq I\left(\underline{X}; \underline{\tilde{Y}}\right) + 1. \tag{1}$$

## III. Upper Bounds via Decoder Side Information

Even for i.i.d. deletion processes, we know of no approach for establishing a single-letter characterization of the capacity. Given the recent success in establishing lower bounds for the BDC, we hope to establish corresponding good upper bounds.

Genie-aided methods are a useful technique in information theory to establish upper bounds for capacity. In this case, we combine a genie for the RLC with a natural approach for upper-bounding the capacity per unit cost of the RLC in order to obtain our bound.

Specifically, we allow side information that converts the RLC into a memoryless channel, which is much better suited to analysis. Since adding side information only improves the channel capacity, this allows upper bounds on the capacity of BDC via a memoryless channel.

The side information we introduce is the most natural: if a run is entirely deleted, a 0 symbol is introduced at the output of the RLC. Considering our previous example, if the binary input 00011000 had the middle four bits deleted, the runs presented at the enhanced RLC with side information would be $2, 0, 2$. In other words, the side information gives for each run in the input the corresponding length of the run in the output, even if that length is 0. Therefore, we define $\underline{Y}$ to be a *new* modified output of the channel where the sequence of run lengths in $\underline{W}$ has 0 inserted (and the run lengths of $\underline{W}$ split appropriately) if a run was deleted. This is the channel we consider for the upper bound.

Since each run in the input is affected independently, this new channel can also be seen as a memoryless binomial channel where the length of the output run, $y$, is related to $x \geq 1$ denoting the length of the input run via

$$p_{Y|X}(y|x) = \begin{cases} \binom{x}{y} d^{x-y}(1-d)^y & \text{if } 0 \leq y \leq x \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

Now, we can relate the capacity of this memoryless binomial channel to the genie-aided BDC through a capacity-per-unit-cost relationship. Intuitively this is clear, since transmitting input $x$ on the RLC costs $x$ bits on the BDC, and so we expect that the capacity of the BDC is upper bounded by the capacity per unit cost of the binomial channel. This intuition is formalized below.

*Lemma 2:* If we denote the mutual information of the memoryless binomial channel given in (2) as $I(X; Y)$, then,

$$C_{BDC} \leq \max_{p_X(x)} \frac{I(X; Y)}{\mathbb{E}[c(X)]}, \tag{3}$$

where $c(x) = x$.

*Proof:* By Lemma 1, we know that for an input of $n$ bits to the BDC, $\lim_{n \to \infty} \frac{1}{n} I(\underline{V}; \underline{W}) = \lim_{n \to \infty} \frac{1}{n} I(RL(\underline{V}); RL(\underline{W}))$. As the genie-aided receiver knows which runs are deleted, we have $I(RL(\underline{V}); RL(\underline{W})) \leq I(\underline{X}; \underline{Y})$, where $\underline{X}$ is the set of input runs and $\underline{Y}$ is the set of genie-aided output runs. We now need only to observe that

$$\max_{p_{\underline{X}}(\underline{x}): w(\underline{X}) = n} \frac{1}{n} I(\underline{X}; \underline{Y}) \leq \max_{p_{\underline{X}}(\underline{x}): \mathbb{E}[w(\underline{X})] = n} \frac{1}{n} I(\underline{X}; \underline{Y})$$

$$\leq \max_{p_{\underline{X}}(\underline{x})} \frac{I(\underline{X}; \underline{Y})}{\mathbb{E}[w(\underline{X})]}$$

$$= \max_{p_X(x)} \frac{I(X; Y)}{\mathbb{E}[X]}, \tag{4}$$

where the last step follows from the fact the capacity per unit cost of a discrete memoryless channel (DMC) is maximized by choosing an i.i.d. input distribution: $p_{\underline{X}}(\underline{x}) = \prod_{i=1}^{m} p_X(x_i)$ [16]. This completes the proof. ∎

The standard upper bound on the capacity of a DMC (see [17] problem 4.17) was generalized to the capacity per unit cost by Abdel-Ghaffar [16].

*Theorem 1:* [Abdel-Ghaffar] Consider a memoryless channel defined by $p_{Y|X}(y|x)$ with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$. Let $q_Y(y)$ be any distribution on $\mathcal{Y}$ which is absolutely continuous with respect to $p_{Y|X}(y|x)$; Then, for a cost function $c(x)$, the capacity per unit cost satisfies

$$C \le \sup_{x \in \mathcal{X}} \frac{I(x)}{c(x)}, \tag{5}$$

where $I(x) \overset{def}{=} D(p_{Y|X}(\cdot|x) \| q_Y(\cdot))$ is the standard information divergence. ∎

Choosing a valid output run-length distribution allows us to upper bound this capacity per unit cost and thereby the capacity of the BDC, but we still must deal with maximizing over the countably infinite input alphabet $\mathcal{X}$. Since we cannot simply truncate the support of $\mathcal{X}$ and still obtain a valid upper bound, we need bound this infinite optimization problem by a finite one.

We do this in two steps. First, we compute a lower bound on the capacity per unit cost for a binomial channel by truncating the input alphabet. Let $A$ be the lower bound and $p_U(u)$ be the output distribution associated with the optimized input distribution. Next, we spread the finite tail of $p_U(u)$ out to infinity using a geometric distribution to get

$$q_Y(y) = \begin{cases} p_U(y) & \text{if } y < m, \\ c2^{-Ay/(1-d)} & \text{if } y \ge m, \end{cases} \tag{6}$$

where $c = 2^{Am/(1-d)}(1 - 2^{-A/(1-d)})\left(1 - \sum_{y<m} p_U(y)\right)$. Combining any output distribution of this form with Theorem 1 gives the main result of this section:

*Theorem 2:* Then the capacity of BDC satisfies

$$C_{BDC} \le \max\left((1+\delta)A, \max_{1 \le x < M} \frac{1}{x} I(x)\right), \tag{7}$$

where $\delta$ is a positive constant, $A$ comes from (6), and $M$ is a positive integer defined in the proof.

*Proof:* The proof follows from separating the supremum from (5) into two parts. The first uses the the true value of $I(x)$ and ranges over $1 \le x < M$. The second uses a closed form upper bound on $I(x)$ using the geometric tail and handles $x \ge M$.

Let $T(x) = \sum_{y=0}^{m-1} p_{Y|X}(y|x)$ the left tail of $p_{Y|X}$ and define $B = \max_{0 \le y < m} \log \frac{1}{q_Y(y)}$. If $c \ge 1$, let $M = \min\{x \in \mathbb{N} | B\,T(x) \le \delta A\}$. If $c < 1$, then choose $\alpha > 0$ and let $m_0 = \min\{x \in \mathbb{N} | T(x) \le \alpha \log \frac{1}{c}\}$, $m_1 = \frac{1+\alpha}{\delta A} \log \frac{1}{c}$, and $M =$

| $d$ | LB | UB |
|------|--------|---------|
| 0.05 | 0.7283 | 0.816 |
| 0.10 | 0.5620 | 0.704 |
| 0.15 | 0.4392 | 0.6188 |
| 0.20 | 0.3467 | 0.5507 |
| 0.25 | 0.2759 | 0.4943 |
| 0.30 | 0.2224 | 0.4466 |
| 0.35 | 0.1810 | 0.4063 |
| 0.40 | 0.1484 | 0.3711 |
| 0.45 | 0.1229 | 0.33987 |
| 0.50 | 0.1019 | 0.31082 |
| 0.55 | 0.08432 | 0.28382 |
| 0.60 | 0.06956 | 0.25815 |
| 0.65 | 0.05686 | 0.2331 |
| 0.70 | 0.04532 | 0.2083 |
| 0.75 | 0.03598 | 0.183 |
| 0.80 | 0.02727 | 0.157 |
| 0.85 | 0.01938 | 0.1298 |
| 0.90 | 0.01238 | 0.0999* |
| 0.95 | 0.00574 | 0.064* |

TABLE I

THE LOWER BOUND FROM [6] AND THE UPPER BOUND DERIVED FROM THEOREM 2. ENTRIES DENOTED * ARE WORSE THAN THE $1 - d$ BOUND.

$\max\{m_0, m_1\}$. The closed form bound is given by

$$
\begin{aligned}
I(x) &= \sum_{y=0}^{x} p_{Y|X}(y|x) \log \frac{p_{Y|X}(y|x)}{q_Y(y)} \\
&= \sum_{y=0}^{x} p_{Y|X}(y|x) \log \frac{1}{q_Y(y)} - H(Y|X=x) \\
&\overset{(a)}{\le} \sum_{y=0}^{x} p_{Y|X}(y|x) \log \frac{1}{q_Y(y)} \\
&= \sum_{y=0}^{m-1} p_{Y|X}(y|x) \log \frac{1}{q_Y(y)} + \sum_{y=m}^{x} p_{Y|X}(y|x) \log \frac{1}{q_Y(y)} \\
&\le B \sum_{y=0}^{m-1} p_{Y|X}(y|x) + \sum_{y=m}^{x} p_{Y|X}(y|x)\left(\log \frac{1}{c} + \frac{Ay}{1-d}\right) \\
&\overset{(b)}{\le} B\,T(x) + \log \frac{1}{c} \sum_{y=m}^{x} p_{Y|X}(y|x) + Ax, \tag{8}
\end{aligned}
$$

where $(a)$ follows from the bound[1] $H(Y|X=x) \ge 0$ and $(b)$ follows from $\mathbb{E}[Y|X=x] = (1-d)x$.

If $c < 1$ and $x \ge m_0$, then the first term of (8) is upper bounded by $\alpha \log \frac{1}{c}$ and this gives $I(x) \le Ax + (1+\alpha) \log \frac{1}{c}$. If $x > m_1$, then it follows from the definition of $m_1$ that $\frac{I(x)}{x} \le (1+\delta)A$.

If $c \ge 1$ and $x \ge M$, then the first term of (8) is upper bounded by $\delta Ax$ and the second term is negative. This allows us to write $\frac{I(x)}{x} \le (1+\delta)A$. ∎

Theorem 2 provides a simple upper bound on the capacity of the BDC which can be optimized by computer. Table III presents the numerical results derived from Theorem 2 and compares them with the numerical lower bounds given in [6]. Notice that this upper bound does not yet improve over the trivial $1 - d$ upper bound for $d \ge 0.9$.

---

[1] We also have an improved bound on $I(x)$ which is omitted due to length. Its main advantage is that $m_1$ remains bounded as $\delta \to 0$.

## IV. Asymptotic Upper Bounds

We now introduce an approach for obtaining upper bounds on the capacity of the BDC in the limit as the deletion probability $d$ approaches 1; this approach is motivated by the lower bound approach of [7]. In this case, the genie-aided decoder has available a special marker symbol that is sent after every block of $b = \lceil \alpha/(1-d) \rceil$ input bits; this marker cannot be deleted. Adding this special marker symbol can only increase the capacity of the channel (since it could simply be ignored). With the addition of these markers, however, the channel can be though of as a memoryless channel, with each block of $b$ input bits viewed as a symbol; that is, the input alphabet consists of all possible sequences of $b$ bits and the output sequence consists of all possible sequences of up to $b$ bits. We bound the mutual information for this ostensibly simpler channel.

As we are primarily interested in the asymptotic behavior as $d$ goes to 1, we introduce the following simplifications. Each block can be described as an initial bit and a finite vector $(q_1, \ldots, q_j)$ with each $q_i > 0$ and $\sum_{i=1}^{j} q_i = 1$ as follows: the block is divided into $j$ contiguous subblocks, where each $i$th subblock consists of $bq_i$ copies of the same bit; the subblocks alternate between 0's and 1's; and the initial bit determines the value for the first subblock. For this to make sense, the $q_i$ values should also be chosen so that the subblock lengths are integers. However, by taking large enough block lengths we can allow $q_i$ to take on values arbitrarily close any real value, and henceforth we ignore any issues related to rounding to obtain integer-length subblocks. Similarly, by taking $b$ sufficiently large we can make the distribution of the number of bits that pass through a block arbitrarily close to a discrete Poisson distribution with mean $\alpha$. We use this distribution for the number of output bits in what follows. It is clear that the effects of these simplifications vanish in the limiting case as $d$ approaches 1.

Now, we consider the channel $\tilde{V} \to \tilde{W}$, where $\tilde{V}$ is a $b$-bit input symbol (between successive markers) and $\tilde{W}$ is the corresponding output. Since $l(\tilde{W})$ denotes the received sequence length, the mutual information $I(\tilde{V}; \tilde{W})$ can be written as

$$I(\tilde{V}; \tilde{W}) = I(\tilde{V}; \tilde{W}, l(\tilde{W}))$$
$$= \underbrace{I(\tilde{V}; l(\tilde{W}))}_{=0} + I(\tilde{V}; \tilde{W}|l(\tilde{W}))$$
$$= \sum_{l=0}^{b} p_{l(\tilde{W})}(l) \, I(\tilde{V}; \tilde{W}|l(\tilde{W}) = l)$$
$$\leq \sum_{l=0}^{k} p_{l(\tilde{W})}(l) I(\tilde{V}; \tilde{W}|l(\tilde{W}) = l) + \sum_{l=k+1}^{b} p_{l(\tilde{W})}(l) \, l.$$

Recall that we take the limit as $d \to 1$, so that the length distribution of the received sequence is exactly Poisson. This implies that

$$I(\tilde{V}; \tilde{W}) \leq \sum_{l=0}^{k} p_{l(\tilde{W})}(l) \, I(\tilde{V}; \tilde{W}|l(\tilde{W}) = l) + \sum_{l=k+1}^{\infty} \frac{l \, e^{-\alpha} \alpha^l}{l!}. \quad (9)$$

We note that a numerical upper bound $u$ of the capacity of this channel immediately yields the upper bound $\frac{u}{b} = \frac{1-d}{\alpha} u$ on the asymptote of the capacity of the BDC as $d \to 1$. Notice that, for our upper bound, we now are left with bounding a summation of terms up to only received length $k$; we make use of this fact in our resulting numerical optimization. On the other hand, small values of $k$ are perfectly suitable; when $\alpha = 1$ and $k = 4$, the (rather weak) upper bound $I(\tilde{V}; \tilde{W}|l(\tilde{W}) = l) \leq l$ for $l > k$ results in a loss of less than $0.02(1-d)$.

The key to computing this bound is the following two lemmas. The first is quite simple and its proof is omitted.

*Lemma 3:* Consider any DMC defined by $p_{Y|X}(y|x) = W(y|x)$. Suppose there are inputs $x_0, \ldots, x_m$ and non-negative constants $a_1, \ldots, a_m$ such that $\sum_{i=1}^{m} a_i = 1$ which also satisfy

$$W(y|x_0) = \sum_{i=1}^{m} a_i W(y|x_i),$$

then there is a capacity achieving input distribution with zero probability assigned to $x_0$. ∎

The following lemma shows that we can achieve capacity with a reduced input space consisting of symbols which have at most $k$ alternating subblocks. Intuitively, if the sender only needs to consider the cases where at most $k$ bits are obtained at the receiver, then it makes sense that each symbol should consist of at most $k$ alternating subblocks. The advantage of this constraint is that we can significantly reduce the search space over distributions for the sender.

*Lemma 4:* For any input $\underline{u}$ with more than $k$ alternating subblocks, we can show that there exist $k$ sequences $\underline{v}_1, \ldots, \underline{v}_k$ (each with at most $k$ alternating blocks of zeros and ones) such that the modified deletion channel satisfies

$$p_{\tilde{W}|\tilde{V}}(\tilde{w}|\underline{u}) = \sum_{i=1}^{k} s_i \, p_{\tilde{W}|\tilde{V}}(\tilde{w}|\underline{v}_i)$$

for all $\tilde{w}$ such that $l(\tilde{w}) \leq k$. Of course, we require that $s_i \geq 0$ and $\sum_{i=1}^{k} s_i = 1$.

*Proof:* We show that any symbol with $k+1$ subblocks can be replaced by multiple sequences with at most $k$ subblocks while achieving the same output distribution at the receiver for all received sequences of up to $k$ bits. The lemma then follows by induction.

Consider any symbol with $k + 1$ subblocks. Since at most $k$ bits are obtained at the receiver, some subblock is entirely deleted. Imagine that instead of using this bit sequence, we initially choose a block to be deleted, remove it, and then send the corresponding remaining bits over the channel. If we can choose correct probabilities for removing each block, this gives us a distribution over symbols with at most $k$ subblocks, proving the lemma. Specifically, let us denote a symbol $S$ by an ordered pair consisting of a bit and a vector of $q_i$ values. Consider the symbol $(0, (q_1, \ldots, q_{k+1}))$. We will find probabilities $s_i$ such that the output distribution is the same if we instead send the symbol $(1, (q_2/(1-q_1), \ldots, q_{k+1}/(1-q_1))$ with probability $s_1$, $(0, (q_1/(1-q_{k+1}), \ldots, q_k)$ with

probability $s_{k+1}$, and $(0, (q_1/(1-q_i)), \ldots, (q_{i-1}+q_{i+1})/(1-q_i), \ldots, q_{k+1}/(1-q_i))$ with probability $s_i$.

To see that appropriate $s_i$ exist, imagine the following experiment: rather than bits being deleted by the channel with $j \leq k$ making it through, we could choose $j$ bits of $S$ randomly and send them (in the appropriate order). But instead of choosing just $j$ bits, imagine instead that we successively choose bits at random from the input until we obtain at least one bit from each subblock (giving $k + 1 > j$ bits). In this scenario, some bit and its corresponding block is chosen last; we let $s_i$ be the probability the $i$th subblock was chosen last.

We therefore see that the distribution of $j \leq k$ bits chosen from $S$ is equivalent to the distribution obtained by first deleting the $i$th block with probability $s_i$ and then choosing the $j \leq k$ bits from the remaining blocks. This equivalence completes the lemma. ∎

Combining Lemma 3 and Lemma 4 shows that, for the channel with $l(\tilde{W}) \leq k$, there is a capacity achieving input distribution which puts probability mass only on sequences with at most $k$ alternating blocks. Therefore, we have

$$\max_{p_{\tilde{V}}(\cdot)} I(\tilde{V}; \tilde{W}|l(\tilde{W}) \leq k) \leq \max_{\underline{v} \in \mathcal{V}} D\left(p_{\tilde{W}|\tilde{V}}(\cdot|\underline{v}) \middle\| q_{\tilde{W}}(\cdot)\right), \quad (10)$$

where $\mathcal{V}$ is the set of input symbols with at most $k$ alternating symbols and $q_{\tilde{W}}(\underline{w})$ is an arbitrary distribution over binary strings of length $k$. Hence, combining (9) and (10) gives a computable asymptotic upper bound for the deletion channel.

Now, we consider some of the practical issues involved. Basically, the channel input is represented as a $k$-vector $(q_1, \ldots, q_k)$ and the output is one of the $2^{k+1}-1$ binary strings of length at most $k$. This ignores whether the input string starts with a sequence of 0's or 1's though. To handle this, we use the channel symmetry implied by complementary strings. This symmetry implies that any optimal input distribution must choose $\underline{v}$ with the same probability as the complement of $\underline{v}$. Therefore, we can compute (10) by choosing $q_{\tilde{W}}(\underline{w})$ with this symmetry and then maximizing only over inputs starting with 0's. This works because the symmetry of $q_{\tilde{W}}(\underline{w})$ implies the divergence must be the same for any input and its complement.

The most tedious part of this computation is coding an efficient subroutine that computes the output distribution for a fixed input. Once we have this subroutine, the following steps are straightforward. First, we pick a large number of input vectors (either randomly or in a $k-1$ dimensional lattice) and compute their associated output distributions. Next, we use the Blahut-Arimoto algorithm to optimize the implied DMC. Let $q_{\tilde{W}}(\underline{w})$ be the output distribution associated with the optimal output distribution of this DMC. Finally, we perform the $k-1$ dimensional optimization using (9) and (10).

To obtain a valid upper bound, we cannot underestimate the maximum on the RHS of (10). Since the function is smooth and the $k$-D domain is bounded, a good global optimizer should suffice. We used the MATLAB function "fmincon" and verified our results by restarting it from a large number of random initial points. Our best result is currently $C_{BDC} \leq$

$0.7918(1 - d)$ as $d \to 1$, and it was computed using $k = 6$ and $\alpha = 2.0$.

## V. DISCUSSION

In this paper we have studied upper bounds for the *i.i.d.* deletion channel. While there have been upper bounds for the combinatorial deletion problem (e.g. [13]), which can be considered an adversarial deletion model, we believe our results give the best non-trivial upper bounds for the *probabilistic* deletion channel. One approach using side information yields good upper bounds for low deletion probabilities; another approach focused on the case where $d \to 1$. We are currently examining other techniques that bridge these two regimes. We believe that such upper bounds will shed considerable light on the coding techniques and the capacity of the i.i.d. deletion channel.

## ACKNOWLEDGEMENT

## REFERENCES

[1] S. N. Diggavi and M. Grossglauser, "On transmission over deletion channels," in *Proc. Annual Allerton Conf. on Commun., Control, and Comp.*, (Monticello, Illinois), Oct. 2001.

[2] S. N. Diggavi and M. Grossglauser, "Information transmission over finite buffer channels," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1226–1237, March 2006.

[3] A. Kavčić and R. Motwani, "Insertion/deletion channels: Reduced-state lower bounds on channel capacities," in *Proc. IEEE Int. Symp. Information Theory*, (Chicago, Illinois), p. 229, June 2004.

[4] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of deletion channels," in *Proc. Annual Allerton Conf. on Commun., Control, and Comp.*, (Monticello, Illinois), Oct. 2004.

[5] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of the deletion channel," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4648–4657, Oct. 2006.

[6] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of i.i.d. deletion and duplication channels," [Online]. Available: http://www.eecs.harvard.edu/~michaelm/postscripts/toitsubmita.ps.

[7] M. Mitzenmacher and E. Drinea, "A simple lower bound for the capacity of the deletion channel.," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4657–4660, Oct. 2006.

[8] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics - Doklady*, vol. 10, pp. 707–710, February 1966.

[9] R. L. Dobrushin, "Shannon's theorems for channels with synchronization errors," *Problems of Inform. Transm.*, vol. 3, no. 4, pp. 11–26, 1967.

[10] N. J. A. Sloane, "On single - deletion - correcting codes." In D. Ray-Chaudhuri Festschrift, 2001. [Online]. Available: http://www.research.att.com/~njas/doc/dijen.ps.

[11] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors." Lincoln Lab. Group Report, Oct. 1961.

[12] K. S. Zigangirov, "Sequential decoding for a binary channel with drop-outs and insertions," *Problems of Inform. Transm.*, vol. 5, no. 2, pp. 17–22, 1969.

[13] J. D. Ullman, "On the capabilities of codes to correct synchronization errors," *IEEE Trans. Inform. Theory*, vol. 13, pp. 95–105, Jan. 1967.

[14] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of deletion channels," in *Proc. IEEE Int. Symp. Information Theory*, (Chicago, Illinois), p. 227, June 2004.

[15] M. Mitzenmacher, "Capacity bounds for sticky channels," [Online]. Available: http://www.eecs.harvard.edu/~michaelm/postscripts/submitisit2007a.pdf.

[16] K. A. S. Abdel-Ghaffar, "Capacity per unit cost of a discrete memoryless channel," *Electronic Letters*, vol. 29, pp. 142–144, Jan. 1993.

[17] R. G. Gallager, *Information theory and reliable communications*. New York: John Wiley and Sons, 1968.