

Capacity Bounds for Sticky Channels

Michael Mitzenmacher, *Member, IEEE*

Abstract—The capacity of sticky channels, a subclass of insertion channels where each symbol may be duplicated multiple times, is considered. The primary result is to provide nearly tight numerical upper and lower bounds for the independent and identically distributed (i.i.d.) duplication channel.

Index Terms—Capacity bounds, duplication channels, insertion/deletion channels, sticky channels.

I. INTRODUCTION

IN this paper, we consider the capacity of *sticky channels*, a subclass of insertion channels where transmitted symbol is independently duplicated a random number of times at the receiver, according to some fixed distribution on the positive integers. We focus on the binary case, although our approaches generalize. As an example, when typing at a keyboard, if the key is held too long, multiple copies of the pressed symbol can appear even though only one copy of the symbol was intended. We consider both lower and upper bounds on capacity, with our primary result being numerical approaches that give nearly tight numerical upper and lower bounds for the independent and identically distributed (i.i.d.) duplication channel, where each bit is duplicated with probability p .

A key feature of binary sticky channels that we take advantage of is that contiguous blocks of zeroes (resp., ones) at the sender correspond to blocks of zeroes (resp., ones) at the receiver. It is this property that intuitively makes sticky channels easier than more general insertion/deletion channels. As an example, suppose that a channel duplicates every bit sent so that exactly two copies arrive. In this case, the original transmission is trivial to recover!

A primary motivation for studying sticky channels is to gain more insight into insertion and deletion channels, which have proven extremely challenging. (See, for example, [2], [4], [5], [7].) General methods for computing the capacity of such channels remains an open problem, and codes for such channels thus far appear rather weak. Sticky channels represent possibly the simplest channel of this type. One might hope, in the future, that capacity bounds or codes for more general insertion/deletion channels could arise from a reduction involving sticky channels. Further, while it may be overly optimistic to hope that studying sticky channels may provide a path toward better results for general insertion and deletion channels, having nearly tight capacity bounds on a subclass of channels would be useful for determining the quality of more general approaches.

Manuscript received April 11, 2007; revised October 2, 2007. This work was supported in part by the National Science Foundation under Grant CCF-0634923.

The author is with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: michaelm@eecs.harvard.edu). Communicated by I. Dumer, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2007.911291

Sticky channels also appear natural in practice. Duplications and deletions can occur over channels where there are timing discrepancies in the clocks of the sender and receiver. In such channels, it may be possible to speed up the receiver clock so that all induced errors are symbol duplications instead of deletions, at the risk of possibly introducing more errors.

We provide a numerical approach for computing lower bounds for the capacity of sticky channels which appears to give nearly tight results. We explicitly demonstrate our approach by considering two channels: the elementary i.i.d. duplication channel, where each bit is independently duplicated with probability p , and the geometric i.i.d. duplication channel, where each bit is independently replaced with a geometrically distributed number of copies. We also consider upper bounds, which prove more complex, and hence, we consider only the elementary i.i.d. duplication channel. Our results demonstrate that our lower bounds are nearly tight for this channel.

II. A CAPACITY CORRESPONDENCE

We begin by demonstrating a general correspondence between the capacity of a binary sticky channel and the capacity per unit cost of a derived channel with an alphabet corresponding to the positive integers. This derived channel can be used to calculate bounds for the sticky channel.

Consider a message X sent over the sticky channel, yielding a received sequence Y . Each maximal block of contiguous zeros (or ones) in X will give a maximal block of contiguous zeros (or ones) in Y . If we restrict ourselves to codewords that begin with a zero, we may view the sticky channel as having input and output symbols that are integers, corresponding to length of each maximal block. For the case of the sticky channel, henceforth denoted by SC, each bit is duplicated a number of times given by some distribution \mathcal{D} over the integers; we call \mathcal{D} the *duplication distribution*. We may represent this channel by a matrix P , where P_{ij} is the probability that a block of i contiguous bits, all 0 (resp., 1), at the sender yields a block of length $j \geq i$ contiguous bits, also all 0 (resp., 1), at the receiver. When doing numerical calculations, it can be advantageous to assume that P is finite, which corresponds to assuming that blocks in the input have a fixed maximum length and the duplication distribution has a finite support; we return to this point below.

Consider the derived channel obtained by using block lengths as symbols. We call this derived channel the sticky block channel, or SBC. As an example, suppose that we have the following input to the SC

0011000101101111001011.

When viewed as an input to the SBC, the corresponding input would be written as

2 2 3 1 1 2 1 4 2 1 1 2.

The mapping between SC and SBC outputs is similarly trivial.

Notice that the SBC is memoryless. It is natural to assign the symbol (integer) i a cost i in the SBC, since it corresponds to i bits in the SC. We claim that it is immediate that the capacity of the SC, under the duplication distribution \mathcal{D} , is equal to the capacity per unit cost of the SBC given by the matrix P .

Although the correspondence is very natural, in order to point out a few details, we formalize a proof, following the notation of [8]. Let an (n, M, ν, ϵ) code be one with block length, codewords, a bound ν on the total cost of a codeword, and average error probability at most ϵ . As in [8, Definition 2], a number $R > 0$ is an ϵ -achievable rate per unit cost if for every $\gamma > 0$, there exists $\nu_0 > 0$ such that for $\nu \geq \nu_0$, an (n, M, ν, ϵ) code can be found with $\log M > \nu(R - \gamma)$. Further, R is achievable per unit cost if it is ϵ -achievable per unit cost for all $0 < \epsilon < 1$, and the capacity per unit cost is the least upper bound over the achievable R .

Theorem 2.1: The capacity per unit cost of the SBC equals the capacity of the SC.

Proof: First, we argue that any rate per unit cost achievable by the SBC is achievable (up to $o(1)$ differences) by the SC channel. If the SBC has capacity per unit cost C , then for every $\epsilon, \gamma > 0$, for sufficiently large n there exists an (n, M, ν, ϵ) code with $\log M > \nu(C - \gamma)$. Moreover, as shown in [8], we may assume that ν is linear in n . Such a code immediately maps to a code for the SC, except that the cost per codeword in the SBC, which corresponds to the length in the SC, can vary. If we accepted variable-length codewords, we could conclude the argument here; we instead aim for fixed-length codewords.

We restrict ourselves to the subset M' of codewords with error probability at most 2ϵ ; by standard arguments, M' must contain at least half the codewords. We can further restrict ourselves to the subset M'' of codewords from M' of a specific cost ν' with $n \leq \nu' \leq \nu$ for which $(\log M'')/\nu'$ is largest. We then have that $\log M''/\nu' > C - 2\gamma$ (for n sufficiently large and γ sufficiently small). This gives us a code on the SC channel with rate at least $C - 2\gamma$ and error probability 2ϵ , from which one direction of the claim follows.

Similarly, if the SC has capacity C , we can use codes for the SC to obtain codes for the SBC, with the rate of the SC mapping to the rate per unit cost of the SBC. A similar problem arises here in that the length n of a block in bits for the SC does not correspond to a fixed number of symbols in the SBC. However, the number of symbols in the SBC is at most n ; we can consider all n possible codeword lengths in the SBC, and choose the length ℓ that gives the largest subset M' of codewords of that length. As $M' > M/n$, this only affects the capacity per unit cost of the SBC by $o(1)$ terms, so we find that a capacity C for the SC gives a capacity per unit cost C for the SBC. \square

We note that we can also apply the analyses of this paper, including Theorem 2.1, to channels corresponding to finite matrices P that do not arise from a duplication distribution, but instead simply satisfy the requirement that $\sum_{j \geq 1} P_{ij} = 1$. For example, suppose block lengths at the sender are at most L bits, and a sent block of length k is received as a block of length $k + 1$ with probability k/c for a constant $c > L$ and as a block of length k otherwise. That is, a block may obtain an extra bit with probability proportional to the length of the block. Such

a channel can also be analyzed using the SC/SBC correspondence. In this regard, our approach can also be used to handle deletions, as long as we can guarantee that it is never the case that *all* the bits of a block are deleted, so that the block structure is preserved.

We emphasize that when considering the capacity per unit cost of the SBC, we consider the limit $I(X; Y)/|X|$ as $|X|$ grows; this is the standard definition, corresponding to the cost for the sender. However, in the general setting we have described we could also consider $I(X; Y)/E[|Y|]$, which would correspond to the (expected) cost for the receiver. For sticky channels governed by a duplication distribution these values are equal, but they need not be so in the more general block setting with general finite matrices P .

Also, while we have thus far discussed only binary alphabets, we can extend these ideas to determine a similar correspondence for nonbinary alphabets. Suppose that the alphabet for the sticky channel consists of the integers in the range $[0, m - 1]$. A natural corresponding representation for the sticky block channel has each symbol consist of an ordered pair with the first field being a block length and the second field being number in the range $[1, m - 1]$ representing the difference modulo m between the two corresponding symbols in the sticky channel. For example, for a 4-ary channel, the SC input

$$2\ 2\ 2\ 2\ 1\ 1\ 3\ 0\ 0\ 0\ 1\ 0\ 0\ 2$$

would be represented as

$$(4, 2)(2, 3)(1, 2)(3, 1)(1, 1)(2, 3)(1, 2).$$

Note that for the first symbol of the SBC we can adopt the convention that the second field gives the first symbol from the SC; this does not affect the capacity.

Here, the symbol (i, j) for the SBC should have a cost of i , since it corresponds to i symbols in the SC. The capacity of the SC under the duplication distribution \mathcal{D} again has an immediate correspondence to the capacity per unit cost of the SBC.

III. LOWER BOUNDS

This correspondence gives us a means to find a lower bound for the capacity of the SC; simply find a lower bound on the capacity per unit cost of the corresponding SBC. When the block lengths in the SC have bounded finite length and the duplication distribution has finite support, then the corresponding matrix P is finite. In this case, capacity per unit cost given a matrix P can be computed numerically, using a variation of the Blahut–Arimoto algorithm for calculating the capacity, under the conditions that we are dealing with finite alphabets and positive symbol costs, as is the case here [6]. This approach does not give an actual efficient coding scheme, but yields a distribution of block lengths, from which the capacity per unit cost can be derived. (This approach can also be efficiently applied in the case of m -ary alphabets, taking advantage of the symmetry among all SBC symbols corresponding to the same block length.)

We emphasize that the only methods we are aware of for effectively computing the capacity per unit cost require finite alphabets for the derived SBC. As this approach uses a numerical

algorithm, it does not give a closed-form expression for the capacity; finding such an expression remains an open question.

We can therefore find lower bounds for channels where \mathcal{D} has finite support by enforcing a finite maximum block length. This truncation necessarily causes the derived bound to be less than the actual capacity, but the expectation is that, for most channels, with reasonable maximum block lengths, the calculation will yield quite good bounds. By taking increasing values for the bound on lengths of blocks, we can obtain bounds closer and closer to the true capacity. Our experiments demonstrate that these bounds converge very quickly for the channels we study.

For the elementary i.i.d. duplication channel, where each bit is duplicated with probability p , \mathcal{D} has finite support. For the geometric i.i.d. duplication channel, where each sent bit yields k copies with probability $(1-p)p^{k-1}$, it does not. In this case, we can treat as equivalent all received blocks above some threshold length, effectively truncating the output distribution beyond a certain point. This truncation also still gives a lower bound on the capacity per unit cost, as we are throwing away information at the receiver. Increasing the threshold should improve the accuracy of our bound.

IV. UPPER BOUNDS

While we can gain some insight into how tight our derived lower bounds are by increasing the maximum input block length and considering the corresponding change, such an approach does not lead to a corresponding upper bound.

More formal upper bounds can be obtained using information divergence. A similar approach was suggested for deletion channels in [3]. In this setting, however, we are aiming for very tight bounds, making the approach much more challenging. An advantage we have here is that we have explicit approximations to the optimal input distribution, arising from the distributions we find via the lower bound techniques that can serve as a guide in obtaining upper bounds.

We recall this approach in the setting of discrete memoryless channels. For example, consider a channel $X \rightarrow Y$ with input alphabet \mathcal{X} and output alphabet \mathcal{Y} , governed by the distribution $Q_{Y|X}$, and consider any distribution $B(y)$ on \mathcal{Y} . Then we have that the capacity of the channel is upper-bounded by

$$\max_x D(Q_{Y|X=x} \| B) = \sum_{y \in \mathcal{Y}} Q(y|x) \log[Q(y|x)/B(y)].$$

This upper bound approach generalizes to the setting of capacity per unit cost [1]. Specifically, for a nonnegative cost function $c(x)$ and distribution $B(y)$, the capacity per unit cost C' has the upper bound

$$C' \leq \max_{x \in \mathcal{X}} \frac{D(Q_{Y|X=x} \| B)}{c(x)}.$$

While one could in fact use any distribution B to obtain an upper bound, for achieving tight bounds, we are naturally guided to use the information we have about near-optimal input distributions from our lower bound approach. We therefore look for distri-

butions B derived from a suitable input distribution A on \mathcal{X} . Specifically, we have $B(y) = \sum_{x \in \mathcal{X}} A(x)Q(y|x)$, and let

$$I_A(x) \equiv D(Q_{Y|X} \| B)l = \sum_{y \in \mathcal{Y}} Q(y|x) \log[Q(y|x)/B(y)].$$

We then have the upper bound

$$C' \leq \max_{x \in \mathcal{X}} \frac{I_A(x)}{c(x)}.$$

As our lower bound computation naturally gives us a truncated input distribution that is near capacity, for this distribution $I_A(x)/c(x)$ should be close to the true capacity for x in the support of the input distribution. Truncated distributions are unsuitable, however, because giving a zero probability to some output symbol yields a trivial infinite upper bound. We must therefore extend the truncated distribution by adding a suitable tail, with the hope that such an addition will still yield an upper bound close to the true capacity.

This approach poses nontrivial challenges in this setting. Because we must bound the maximum of $I_A(x)/c(x)$ over all symbols in the input alphabet, we may have to choose the tail based on the ability to analyze it for x growing toward infinity, rather than its actual performance, weakening the bound. At the same time, there is generally a ‘‘bump’’ in $I_A(x)/c(x)$ caused by the transition between the original distribution and the added tail; one must try to make this transition between distributions as smooth as possible so as to minimize the increase in the upper bound due to this bump. Because of this, it appears difficult to achieve tight bounds generally. We have successfully used geometric distributions for a tail for the input for the elementary i.i.d. duplication channel, as we now describe.

Suppose that for the elementary i.i.d. duplication channel we start with an input distribution $L(x)$, such as the distribution derived from the lower bound calculation, and augment it with a geometric tail. That is, we take the derived distribution $L(x)$ and replace it with $A(x)$ given by

$$A(x) = \begin{cases} L(x), & \text{if } x < m \\ \gamma \left(1 - \sum_{i=1}^{m-1} L(x)\right) (1-\gamma)^{x-m}, & \text{if } x \geq m \end{cases}$$

for some constants $0 < \gamma < 1$ and m . Suitable values of γ and m can be searched for numerically. We now consider the behavior of $I_A(x)/c(x)$ for $x \geq 2m$. First, note that for $y \geq 2m$, letting $\delta = \gamma \left(1 - \sum_{i=1}^{m-1} L(x)\right) (1-\gamma)^{-m}$, we have

$$\begin{aligned} B(y) &= \sum_{x=\lceil y/2 \rceil}^y A(x) \binom{x}{y-x} p^{y-x} (1-p)^{2x-y} \\ &= \delta \left(\frac{p}{1-p}\right)^y \sum_{x=\lceil y/2 \rceil}^y \binom{x}{y-x} \left(\frac{(1-p)^2(1-\gamma)}{p}\right)^x. \end{aligned}$$

We simplify this expression by using the connection to the Fibonacci polynomials. Recall the Fibonacci polynomial $F_n(x)$ satisfies for $n \geq 1$ the identity

$$F_n(z) = \sum_{j=0}^{\lfloor (n-1)/2 \rfloor} \binom{n-j-1}{j} z^{n-2j-1}$$

and further that

$$F_n(z) = \frac{1}{\sqrt{z^2 + 4}} \left(\left(\frac{z + \sqrt{z^2 + 4}}{2} \right)^n + \left(\frac{z - \sqrt{z^2 + 4}}{2} \right)^n \right).$$

Then simple algebraic manipulation yields

$$\begin{aligned} \sum_{x=\lceil y/2 \rceil}^y \binom{x}{y-x} \mu^x &= \sum_{x'=0}^{\lfloor y/2 \rfloor} \binom{y-x'}{x'} \mu^{y-x'} \\ &= (\sqrt{\mu})^y \sum_{x'=0}^{\lfloor y/2 \rfloor} \binom{y-x'}{x'} (\sqrt{\mu})^{y-2x'} \\ &= (\sqrt{\mu})^y F_{y+1}(\sqrt{\mu}). \end{aligned}$$

Hence, simplifying for $B(y)$, we have

$$B(y) = \delta(p(1-\gamma))^{y/2} F_{y+1}((1-p)\sqrt{(1-\gamma)/p}).$$

For convenience, we leave $B(y)$ in this form, and note the key point, which is that $B(y) \approx \delta' \rho^{-y}$ for some constants ρ and δ' ; specifically, we may take

$$\rho = \frac{2}{(1-p)(1-\gamma) + \sqrt{4p(1-\gamma) + (1-p)^2(1-\gamma)^2}}.$$

If $F_n(z)$ depended on a single exponential term in n , instead of the sum of two, this approximation would be an equality, and our work would be much simpler; however, in practice, we can simply note that for suitably large y , we have that

$$\log(1/B(y)) \leq \log(1/\delta') + y \log \rho + \epsilon$$

for an appropriately small choice of ϵ which can be determined numerically. We therefore find

$$\begin{aligned} I_A(x) &= \sum_{y=x}^{2x} Q(y|x) \log[Q(y|x)/B(y)] \\ &= -H(Y|X=x) + \sum_{y=x}^{2x} Q(y|x) \log(1/B(y)) \\ &\leq -H(Y|X=x) + \log(1/\delta') + \epsilon \\ &\quad + \sum_{y=x}^{2x} y Q(y|x) \log \rho \\ &= -H(Y|X=x) + \log(1/\delta') + \epsilon + (1+p)x \log \rho. \end{aligned}$$

A straightforward calculation shows that $H(Y|X=x) = o(x)$. We therefore have that $I_A(x)/x$ converges to $(1+p) \log \rho$. Moreover, $H(Y|X=x)$ is increasing in x , since it corresponds to a binomial random variable with x trials. Therefore, for large enough x , so that

$$H(Y|X=x) > \log(1/\delta') + \epsilon,$$

we find as $c(x) = x$

$$I_A(x)/c(x) < (1+p) \log \rho.$$

Using these facts, we can determine upper bounds numerically with this approach.

This approach should be useful for other sticky channels, such as the geometric i.i.d. duplication channel, but there are both

theoretical and numerical challenges. First, studying the asymptotics of $D(Q_{Y|X=x}||B)/c(x)$ becomes harder, since technically for all values of y the output distribution $B(y)$ depends on values of $A(x)$ with $x < m$, for which $A(x) = L(x)$. We could avoid this in analyzing the elementary i.i.d. duplication channel once $y \geq 2m$. Second, it is not clear geometric tail distributions are suitable; one may have to massage the transition for the distribution $A(x)$ around the value m more carefully. Simplifying and generalizing this upper bound argument for additional sticky channels therefore remains an open question.

V. NUMERICAL CAPACITY CALCULATIONS

We perform calculations for lower bounds on the capacity of the elementary i.i.d. duplication channel and the geometric i.i.d. duplication channel based on Section III. We also consider upper bounds for the elementary i.i.d. duplication channel as described in Section IV.

Before beginning, for comparison purposes it is worth noting that the i.i.d. duplication channel has a simple, zero-error coding scheme that works for any duplication probability p . Using the correspondence with the SBC, we lay out symbols as blocks; by using alternating blocks of 1 bit or 3 bits, we guarantee no confusion in decoding even if duplications occur. We can use a 1-bit block to represent a 0 and a 3-bit block to represent a 1. Using a random input distribution, this already yields a capacity per unit cost for the SBC (and hence a capacity for the sticky channel) of 0.5; the optimal input distribution increases the capacity to nearly 0.5515. More generally, one could handle any finite or countably infinite alphabet without confusion by using blocks of length $1, 3, 7, \dots, 2^i - 1, \dots$ as needed. The optimal distribution in this case yields a capacity lower bound of nearly 0.6025. These schemes provides a nice baseline when considering our results. Of course, this approach does not work for more general channels such as the geometric i.i.d. duplication channel.

Also, for the i.i.d. duplication channel, the capacity is trivially 1 when $p = 0$ or $p = 1$, and in these cases the optimal input distributions are uniform, corresponding to block lengths that are geometrically distributed with mean two.

A. Lower Bounds

For the elementary i.i.d. duplication channel, we present results of capacity lower bound calculations where the input stream is limited to maximal blocks of 8 and 16 bits in Fig. 1; these curves are labeled by ‘‘Opt.’’ (The maximal blocks in the received streams are naturally limited to 16 and 32 bits, respectively.) As can be seen, the difference is small enough that the two curves are almost indistinguishable in the figure, and further increases in the maximal block length yield similarly small gain. For example, calculations of capacity with maximal blocks of 20 bits in the input rounded to four decimal places agree with the result from 16 bits, except where the rounding leads to a difference in the fourth decimal place. This provides evidence for our conjecture that the calculations yield results quite close to the actual capacity. The capacity lower bound is minimized at around $p \approx 0.31$, where it is approximately 0.64.

A further interesting behavior is worth noting. For the elementary i.i.d. duplication channel, for values of p near 0 and 1, the optimal input distribution remains close to geometrically

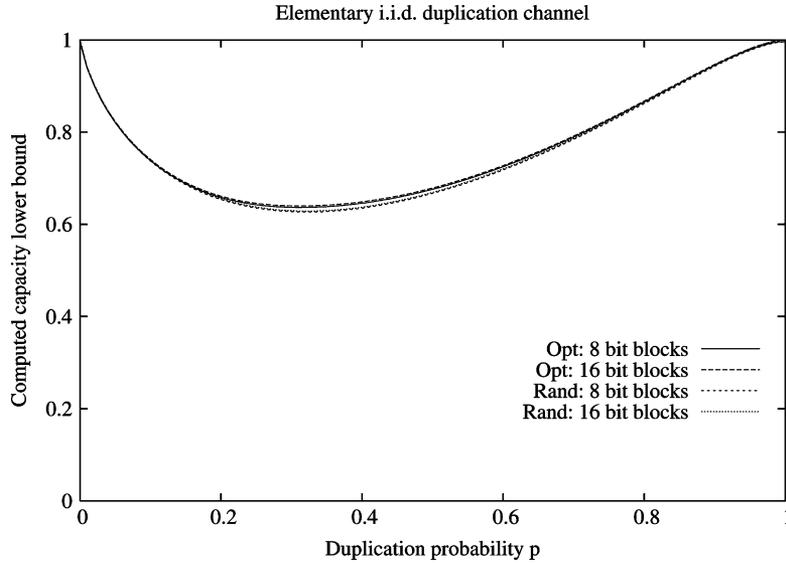


Fig. 1. Capacity lower bounds for the elementary i.i.d. duplication channel based on numerically calculated optimal input distributions and random (truncated) input distributions for maximal block lengths of 8 and 16 bits.

TABLE I
COMPUTED OPTIMAL DISTRIBUTIONS, TO FOUR DECIMAL PLACES, FOR THE ELEMENTARY I.I.D DUPLICATION CHANNEL

Block Length	1	2	3	4	5	6	7	8
$p = 0.1$	0.5138	0.2330	0.1210	0.0625	0.0321	0.0201	0.0045	0.0121
$p = 0.5$	0.5804	0.1888	0.1128	0.0502	0.0298	0.0225	0.0001	0.0155
$p = 0.9$	0.5169	0.2527	0.1215	0.0581	0.0277	0.0133	0.0063	0.0035

distributed with mean 2, as they would be if each bit were independent and uniform. The distribution drifts from this at intermediate values of p ; some examples (for blocks up to 8 bits) are shown in Table I. The distributions are nearly geometric at $p = 0.1$ and $p = 0.9$; the deviation from the geometric is stronger but still slight for intermediate values such as $p = 0.5$.

Indeed, the capacity achieved by geometric distributions is not too much less than for our computed optimal curves, as shown in Fig. 1; these curves are labeled by “Rand.” (We calculate the capacity with the probability that a block of maximal length k being 2^{-k+1} instead of 2^{-k} .) This fact may prove useful for designing computationally efficient coding schemes that perform close to capacity in practice, by having them use random input distributions instead of specially tailored distributions.

The geometric i.i.d. duplication channel naturally exhibits more complicated behaviors. For our calculations, we require bounds on the maximal block length in the received sequence. Because of this, we limited our calculations for this channel to $p \leq 0.6$. At $p = 0.6$, a received block will be, on average, slightly more than twice as long as the corresponding input block; we allow received blocks to be as long as five times the maximal length of a maximal input block. Longer received blocks are treated as truncated at this length, and the matrix P is computed accordingly.

In Fig. 2, we present results of capacity calculations where the input stream is limited to maximal blocks of 8 and 16 bits. Again, the difference between using maximal blocks of 8 and 16 bits is quite small. For this channel the capacity falls to 0 as p goes to 1.

We also again present results for random codebooks. They perform nearly as well as the optimal calculated truncated distri-

butions for small values of p , but as p grows larger, the difference appears to increase steadily. Indeed, in contrast to the elementary i.i.d. duplication channel, we find that as p increases, the optimal calculated distribution differs significantly and increasingly from a geometric distribution; the numerically computed optimal distributions appear to become more focused on a small and well-spread set of values. Some examples for blocks up to 8 bits are shown in Table II. This perhaps explains the challenge in applying our upper bound techniques to this channel; a more complicated tail distribution may be required.

In summary, our results suggest that our calculated lower bounds are extremely close to the true capacity, as increasing the maximum input and output block lengths leads to only very small differences in the final answers. Also, codes with code-words chosen uniformly at random (even with blocks of limited run lengths) would appear to perform close to capacity for the elementary i.i.d. duplication channel, and for the geometric i.i.d. duplication channel with small p .

B. Upper Bounds

We present numerically derived upper bounds for the elementary i.i.d. duplication channel. In all cases below, we began with a distribution based on our lower bound calculation, with input blocks limited to 30 bits. In all cases, we began by augmenting it with a geometric tail from block length $m = 24$. We attempted to stay with values of p and γ for which the asymptotic values of $I_A(x)/x$ were conservatively far from the derived upper bound. While further optimization of parameter choices might improve things slightly, we found very good results with these choices.

We present results in Table III. For ease of presentation, we have rounded our derived lower bounds down in the fourth decimal place, and the derived upper bounds up in the fourth dec-

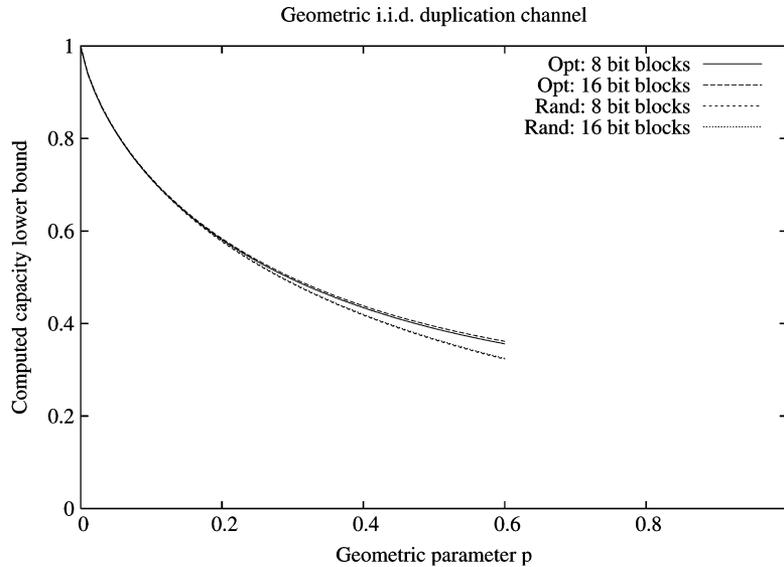


Fig. 2. Capacity lower bounds for the geometric i.i.d. duplication channel based on numerically calculated optimal input distributions and random (truncated) input distributions for maximal block lengths of 8 and 16 bits.

TABLE II
COMPUTED OPTIMAL DISTRIBUTIONS, TO FOUR DECIMAL PLACES, FOR THE GEOMETRIC I.I.D DUPLICATION CHANNEL

Block Length	1	2	3	4	5	6	7	8
$p = 0.1$	0.5078	0.2384	0.1200	0.0626	0.0331	0.0197	0.0057	0.0127
$p = 0.2$	0.5192	0.2202	0.1142	0.0609	0.0379	0.0218	0.0000	0.0257
$p = 0.3$	0.5369	0.1928	0.1132	0.0487	0.0627	0.0044	0.0000	0.0413
$p = 0.4$	0.5631	0.1526	0.1115	0.0545	0.0624	0.0000	0.0000	0.0559
$p = 0.5$	0.6017	0.0907	0.1168	0.0773	0.0424	0.0000	0.0000	0.0712
$p = 0.6$	0.6547	0.0000	0.1416	0.0104	0.0131	0.0000	0.0000	0.0862

TABLE III
COMPARING UPPER AND LOWER BOUNDS FOR THE ELEMENTARY I.I.D. DUPLICATION CHANNEL

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
γ	0.40	0.36	0.36	0.36	0.37	0.39	0.40	0.40	0.40
Lower Bound	0.7405	0.6611	0.6400	0.6488	0.6788	0.7273	0.7914	0.8674	0.9469
Upper Bound	0.7406	0.6618	0.6404	0.6499	0.6797	0.7277	0.7915	0.8675	0.9479
$\lim I_A(x)/x$	0.7218	0.6276	0.6263	0.6269	0.6504	0.6977	0.7243	0.7284	0.7326

imal place. In some cases, the results were actually the same to four decimal places. We emphasize that our point here is not the numbers themselves, which could be improved with further calculation, but that for this and possibly other sticky channels, quite tight bounds are possible with these techniques. Table III demonstrates that the derived lower bounds are quite close to the derived upper bounds across the range of values of p .

VI. CONCLUSION

We have given numerical methods for calculating bounds on the capacity of sticky channels, explicitly showing quite tight bounds in the case of the elementary i.i.d. duplication channel. These capacity results should allow this channel to serve as a test case for future work on more general methods for capacity bounds for insertion–deletion channels. Similarly, these results should serve as a goalpost for efficient codes, either for these specific channels or more general insertion–deletion channels. Many open questions remain, include generalizing or simplifying our upper bound approach, finding closed-form expres-

sions for capacity bounds for sticky channels, and providing efficient coding techniques for sticky channels.

REFERENCES

- [1] K. A. S. Abdel-Ghaffar, “Capacity per unit cost of a discrete memoryless channel,” *IEE Electron. Lett.*, vol. 29, pp. 142–144, 1993.
- [2] S. Diggavi and M. Grossglauser, “Information transmission over a finite buffer channels,” *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1226–1237, Mar. 2006.
- [3] S. Diggavi, M. Mitzenmacher, and H. Pfister, “Capacity upper bounds for deletion channels,” in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jun. 2007, pp. 1716–1720.
- [4] 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. 2007.
- [5] E. Drinea and M. Mitzenmacher, “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.
- [6] M. Jimbo and K. Kunisawa, “An Iteration Method for Calculating the Relative Capacity,” *Inf. Contr.*, vol. 43, pp. 216–233, 1979.
- [7] M. Mitzenmacher and E. Drinea, “A simple lower bound for the capacity of the deletion channel,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4657–4660, Oct. 2006.
- [8] S. Verdú, “On channel capacity per unit cost,” *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1019–1030, Sep. 1990.