Lower Bounds for Asymmetric Communication Channels and Distributed Source Coding

Micah Adler* Erik D. Demaine† Nicholas J. A. Harvey†‡ Mihai Pătraşcu†

Abstract

We prove nearly tight lower bounds on the number of rounds of communication required by efficient protocols over asymmetric channels between a server (with high sending capacity) and one or more clients (with low sending capacity). This scenario captures the common asymmetric communication bandwidth between broadband Internet providers and home users, as well as sensor networks where sensors (clients) have limited capacity because of the high power requirements for long-range transmissions. An efficient protocol in this setting communicates n bits from each of the k clients to the server, where the clients' bits are sampled from a joint distribution D that is known to the server but not the clients, with the clients sending only $O(H(D)+k)$ bits total, where $H(D)$ is the entropy of distribution D. In the single-client case, there are efficient protocols using $O(1)$ rounds in expectation and $O(\lg n)$ rounds in the worst case. We prove that this is essentially best possible: with probability $1/2^{O(t \lg t)}$, any efficient protocol can be forced to use t rounds. In the multi-client case, there are efficient protocols using $O(\lg k)$ rounds in expectation. We prove that this is essentially best possible: with probability $\Omega(1)$, any efficient protocol can be forced to use $\Omega(\lg k/ \lg \lg k)$ rounds. Along the way, we develop new techniques of independent interest for proving lower bounds in communication complexity.

1 Introduction

Many network systems have an inherent asymmetry between two (or more) classes of devices that leads to asymmetric communication channels between devices of different class. For simplicity we call the two classes "servers" (powerful) and "clients" (weak), with faster communication from server to client than from client to server. In some cases the asymmetry arises from the network medium: for example, Internet service providers (servers) recognize that most home users and some business users (clients) download significantly more than they upload, so the network is built (or artificially limited) to have higher server-to-client bandwidth than client-to-server bandwidth. The main examples are broadband connections—asymmetric digital subscriber lines (ADSLs), cable modems, and satellite (fast downlink for more rural areas, with uplink via phone-line modems)—which have a download/upload bandwidth ratio typically between a factor of 5 and 15. Even modern phone-line modems are asymmetric, because the upstream traffic must be passed through a PCM encoder before transmission on the digital backbone, resulting in quantization noise that reduces upstream capacity. In other cases the asymmetry arises from the devices: because there are usually fewer servers than clients and servers are usually tethered, the network can afford to place more resources, particularly computation and power, on the servers. For example, in wireless networks, mobile devices (clients) such as sensors, cell phones, laptops, vehicles, and spacecraft have limited power for long-distance transmission, whereas reception is relatively cheap and tethered base stations (servers) can afford to transmit long distances at high bandwidth.

Adler and Maggs [3] considered the simplest scenario of one client and one server, and showed that, in certain circumstances, the server can use the client's fast downlink to reduce the expected number of bits sent by the client across a slow uplink to significantly less than the length of the client's message. Specifically, in the ***asymmetric transmission problem***, the client wants to send an n-bit string x that is drawn from a probability distribution D (as is assumed for source codes such as Shannon-Fano codes [26] or Huffman codes [16]), but only the server knows the distribution D (e.g., by gathering statistics that the client cannot afford to maintain, or using global knowledge that the client cannot obtain). The client and the server must exchange enough information that the server learns the string x. There are three important objectives to minimize: the number of bits sent by the client, the number of bits sent by the server, and the number of rounds of communication.

*Department of Computer Science, University of Massachusetts, Amherst. micah@cs.umass.edu. Supported by NSF awards EIA-0080119, CCR-0133664, and ITR-0325726.

†MIT Computer Science and Artificial Intelligence Laboratory. {edemaine,nickh,mip}@mit.edu.

‡Supported by a Natural Sciences and Engineering Research Council of Canada PGS Scholarship.

In the original solution proposed in [3], the client sends at most $1.71\,H(D)+1$ bits in expectation, the server sends $O(n)$ bits in expectation, and the number of rounds is $O(1)$ in expectation. By Shannon's Theorem [26], the client must send at least $H(D)$ bits in expectation, so in this measure the protocol is optimal up to constant factors. The expected number of bits sent by the client has been improved to $1.089\,H(D)+1$ by Laber and Holanda [15] (using the same protocol), to $(1+\epsilon)\,H(D)+1$ by Ghazizadeh et al. [14], and to $H(D)+2$ by Watkinson et al. [29, 30] (though in the last, the server sends a factor of $H(D)$ more bits). On the other hand, it is shown in [3] that, for many distributions, the total amount of communication between the client and server must be at least n bits; thus, assuming that the client achieves some savings and sends at most $(1-\epsilon)n$ bits, the server must send $\Omega(n)$ bits. Thus the original protocol is optimal up to constant factors by this measure as well. Techniques from these protocols are useful in circumventing web censorship and surveillance [10], as well as in the design of websites [5].

The focus of this paper is the number of rounds in the communication protocol. This measure is particularly important because of latency. Any time savings obtained from reducing the number of bits sent by the client could easily be lost by the extra latency cost induced by multiple rounds in the protocol, particularly in long-distance networks, such as satellites, where communication has very high latency. The $O(1)$ expected guarantee and the $O(\lg n)$ worst-case guarantee of all existing protocols are not ideal. However, we show that these results are nearly the best possible: no $o(\frac{\lg n}{\lg\lg n})$ bound on the number of rounds can hold with high probability, assuming that the client and server transmit near-optimal numbers of bits. The only previously known lower bound is that one round does not suffice, for a distribution depending on the protocol [3].

The ***multi-client*** version of the asymmetric transmission problem was introduced in [1] as the ***sensor transmission problem***. Here there are k clients, each wanting to transmit an n-bit string to the server. The additional catch is that the strings may be arbitrarily correlated; the probability distribution D is now over length-k lists of n-bit strings. Thus we can expect to save substantially more than if we just ran a single-client solution seperately for each client. As before, only the server knows the distribution D; the clients do not know their correlation to each other, though we can allow them to communicate with each other. Again there are three important objectives to minimize: the total number of bits sent by the clients, the number of bits sent by the server, and the number of rounds of communication.

This problem captures a fundamental and well-studied task in sensor networks: collecting correlated information that is distributed across a set of clients to a central server. Information collected from sensor networks can be correlated in various scenarios, such as sensing the image of an object from similar but distinct viewpoints or measuring weather data from points in the same geographic region. Collecting such correlated information was first studied by Slepian and Wolf [27], who introduced ***distributed source coding***. The impractical nature of Slepian and Wolf's encoding technique has inspired considerable recent work on distributed source coding; see, e.g., [12, 24, 28, 4, 2, 19, 20, 21, 18, 13, 23, 7]. A comprehensive survey of this work is found in [31]. All of the work described in [31] makes significant restrictions on the distribution D, and [31] stresses handling of more general distributions as the main technical issue to overcome in order to deploy distributed source coding to sensor networks. On the positive side, most of these results use only a single round, with no feedback from the server to the clients.

Adler [1] develops a solution to the k-client sensor transmission problem that is effective for any distribution D, at the cost of requiring several rounds of communication. In this protocol, the total number of bits sent by the clients is $O(H(D)+k)$, which is optimal up to constant factors: Shannon's Theorem gives a lower bound of $H(D)$, and each client may be required to send 1 bit. The number of bits sent by the server is $O(kn+H(D)\lg n)$, which in most cases is optimal up to constant factors: the lower bound from [3] shows that at least kn bits total must be sent. The number of rounds is $O(1+\lg\min\{H(D),k\})$ in expectation. We prove here that this number of rounds is nearly the best possible: there is a family of distributions with $H(D)\geq k$ (as usual) and on which any protocol must use $\Omega(\frac{\lg k}{\lg\lg k})$ rounds with constant probability.

To stress the importance of the number of rounds in the character of this problem, we show how the (multi-client) sensor transmission problem can be solved in a serial fashion using a solution to the (single-client) asymmetric transmission problem. The server collects the n-bit string from each client in turn, conditioning the distribution of the next client on the values received from the previous clients. By the definition of conditional entropy, this protocol is optimal up to constant factors in terms of the total number of bits sent in either direction. However, the number rounds is $\Omega(k)$. Thus the main contribution of [1] was to parallelize this process, achieving $O(1+\lg k)$ rounds, while maintaining (rough) optimality up to constant factors for the number of bits sent in each direction. Our lower bounds show that this parallelization is nearly the best possible.

1.1 Our Results. In this paper, we prove lower bounds on the number of communication rounds required for both the sensor transmission problem and the asymmetric transmission problem. These lower bounds are nearly tight with respect to the upper bounds from [1] and [3] respectively. In particular, we demonstrate that $\Omega(\frac{\lg k}{\lg\lg k})$ rounds are required for any protocol in which the clients send $O(H(D)+1)$ bits in each round (in expectation), even if the server sends up to $2^{n^{1-\epsilon}}$ bits, for any $\epsilon > 0$. In the single-client setting we demonstrate that, for any $t = O(\lg n/\lg\lg n)$, there is a distribution D such that any efficient protocol for the asymmetric transmission problem requires t rounds with probability $2^{-O(t\lg t)}$. As a consequence, there is no high-probability bound of $o(\lg n/\lg\lg n)$ on the number of rounds. Again, these results assume that the client sends $O(H(D)+1)$ bits and allow the server to send up to $2^{n^{1-\epsilon}}$ bits.

To prove our lower bounds, we reformulate the problems in the usual communication-complexity framework where two parties collaboratively compute a function. To do so, we define a new problem, called the ***string-color problem*** SC_n. The setup is identical to the asymmetric transmission problem, except that the server also has a map that associates a color with each binary string of length n, i.e., the server's inputs are a distribution D and a map $\phi : \{0,1\}^n \to \{\mathsf{red}, \mathsf{blue}\}$. The objective is for both parties to learn the color of the client's string. Clearly, SC_n reduces to the asymmetric transmission problem with only one additional round, and one additional bit of communication.

Our main theorem for the single-client asymmetric transmission problem, phrased in terms of the string-color problem, is as follows. Here H is an upper bound on the entropy of any distribution D which the server can receive as an input.

THEOREM 1.1. *For any $\epsilon > 0$, there is a constant $c > 0$ such that, for any $t \le c\frac{\lg n}{\lg\lg n}$, there is a distribution $\mathscr{D}$ on inputs to SC_n for which any communication protocol for SC_n, in which the client sends $O(t\cdot(H+1))$ bits in expectation and the server sends at most $2^{n^{1-\epsilon}}$ bits in expectation, uses at least t rounds with probability $2^{-O(t\lg t)}$.*

It is instructive to understand why the constraints on the numbers of bits sent are necessary: relaxing them too much trivializes the problem. First, if the client is allowed to send n bits (or if $H(D) = n$), then the problem can be solved trivially in one round by direct encoding of the string. Second, if the server is allowed to send $\Omega(2^n)$ bits, then the server can send an encoding of the distribution D (or at least an approximation thereof), and the client can send a Huffman encoding [9] of its string to the server using only $O(H(D)+1)$ bits, for a total of one round. Finally, imposing a hard bound on the message lengths, rather than a bound in expectation, would significantly simplify our lower-bound proof. However, it would also make the problem meaningless because there are distributions with constant entropy such that representing a sample from the distribution requires $\Omega(n)$ bits in the worst case.

Let $\mathsf{SC}_{n,k}$ denote the k-client version of the string-color problem. The setup is identical to the sensor transmission problem, and the server again has a map $\phi : \{0,1\}^n \to \{\mathsf{red}, \mathsf{blue}\}$. The objective is for the server to learn the color of all clients' strings, and for each client to learn the color of its own string. Clearly, $\mathsf{SC}_{n,k}$ reduces to the asymmetric transmission problem with only one additional round, and k additional bits of communication from the server. Our main theorem for the sensor transmission problem, phrased in terms of the string-color problem, is as follows.

THEOREM 1.2. *For any $\epsilon > 0$, there is a constant $c > 0$ such that, for any $k \le n^c$, there is a distribution $\mathscr{D}$ on inputs to $\mathsf{SC}_{n,k}$ for which any communication protocol for $\mathsf{SC}_{n,k}$, in which the clients send $O((H+1)\cdot\frac{\lg k}{\lg\lg k})$ bits in total in expectation and the server sends at most $2^{n^{1-\epsilon}}$ bits in expectation, uses $\Omega(\lg k/\lg\lg k)$ rounds with probability $\Omega(1)$. This lower bound holds even if all messages, in particular those from clients, are broadcast (seen by everyone), and even if the string of each client is chosen independently (no correlation).*

1.2 Technical Contributions. Our lower bounds are proved in the realm of communication complexity; for a general introduction to the field, see [17]. We develop several new techniques and ideas, which we feel are of independent interest and we expect are useful for other problems in this field.

For context, we describe two standard message-elimination techniques which we use: message switching and the round-elimination lemma. To state these results, it is necessary to define some variants of a given communication problem. An abstract communication problem is specified by a function $f : \mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$, where $\mathcal{X}$ is the domain of Alice's input and $\mathcal{Y}$ is the domain of Bob's input. We let f^A denote the communication problem in which Alice communicates first; f^B is defined similarly. We define $f^{(k)}$ to be a communication problem based on f, where Alice is given $x_1,\ldots,x_k \in \mathcal{X}$ and Bob is given $y\in\mathcal{Y}$ and $i\in\{1,\ldots,k\}$. Their goal is to compute $f(x_i, y)$. Given a distribution $\mathscr{D}$ on inputs to f, we may define a related distribution $\mathscr{D}^{(k)}$ on inputs to $f^{(k)}$. First we draw k pairs $(x_1, y_1), \ldots, (x_k, y_k)$ from the distribution $\mathscr{D}$. Next, we choose i uniformly from $\{1, \ldots, k\}$. Finally, Alice is given $(x_1, y_1), \ldots, (x_k, y_k)$ and Bob is given y_i and i. The resulting distribution is $\mathscr{D}^{(k)}$.

The ***message-switching lemma*** of Chakrabarti and Regev [6] states that Bob can postpone sending his first message if instead Alice sends him every possible reply.

Lemma 1.1. (Message-switching lemma [6]) *If f^B has a t-message protocol in which Bob's messages have size b and Alice's messages have size a, then f^A has a $(t-1)$-message protocol with the same error probability in which Alice's first message has size $2^b a$, Bob's next message has size $2b$, and all subsequent messages are of the usual size.*

Now consider a protocol for $f^{(k),A}$ in which Alice's first message has much fewer than k bits. Because Alice does not know i, we would expect her message to be rather useless, as she cannot send even one bit about each of her k inputs. This intuition is formalized by the ***round-elimination lemma***, originally due to Miltersen et al. [22], and refined by Sen [25]. The following is a slight reformulation of Sen's statement. Let $\varepsilon^P_{\mathscr{D}}$ be the error made by protocol P under distribution $\mathscr{D}$.

Lemma 1.2. (Round-elimination lemma [25]) *Suppose the communication game $f^{(k),A}$ has a t-message protocol P in which Alice's first message has a bits. Then f^B has a $(t-1)$-message protocol Q where all messages have the same size and $\varepsilon^Q_{\mathscr{D}} \leq \varepsilon^P_{\mathscr{D}^{(k)}} + \sqrt{a/k}$.*

One of the most serious issues in applying these lemmas to our problem is that we do not have hard bounds on the message sizes, but only expected bounds. In principle, one could simply apply a Markov bound and impose a hard limit on all messages, by introducing some error. However, this is not feasible in our case, because of the following characteristic of our lower bound. The probability that t rounds are needed decreases exponentially with t. Hence, after each round elimination, our proof needs to restrict itself to a rapidly shrinking probability space. Thus, the error that we can afford to introduce must be sufficiently small to be negligible in this exponentially small probability space. Consequently, a simple Markov bound would cause the client's messages to be too large.

Instead of imposing a hard limit on message sizes from the beginning, we need more careful control over the clients' expected message size in the course of message elimination. To achieve this, we use a bicriterion round-elimination lemma, which can eliminate the first message while bounding both the error and some other parameter (in our case, the expected length of future messages).

However, simply delaying a hard restiction on the message lengths in this fashion turns out to be insufficient. The alternative we propose, which is one of the most interesting technical aspects of our proof, is to deal with two types of errors simultaneously. The first type is the ordinary distributional error $\varepsilon^P_{\mathscr{D}}$, which is an error made by protocol P when the inputs are distributed according to $\mathscr{D}$. This error can depend on both Alice's and Bob's inputs. The second type is ***unilateral error***, denoted $\mu^P_{\mathscr{D}}$, which is a type of error which can depend only on the input of one player (say Bob). When considering the natural matrix describing the communication protocol over $\mathcal{X} \times \mathcal{Y}$, this error consists of entire rows of the matrix. By definition, any error made by the protocol must be accounted for by either the ordinary or the unilateral error (or both). Because of the structure of our proof, we can deal with much larger unilateral error than unrestricted error, which justifies our interest in it.

2 The Hard Input Distribution

We now describe the distribution over the problem inputs under which the lower bound of Theorem 1.1 comes true. Specifically, we describe a collection $\mathcal{D}_{n,t}$ of distributions over n-bit strings, parameterized by t and n. In the hard instance, the server receives a uniformly random $D \in \mathcal{D}_{n,t}$ and the client receives a sample from $\{0,1\}^n$ chosen according to D. We view such a distribution D as a binary tree where each leaf has depth n. A leaf represents a sample from the support of the distribution; the bits of the sample are determined by the directions of the children on the root-to-leaf path.

An example of a binary tree corresponding to a distribution is shown in Figure 1. The tree consists of several instances of a structure which we call T. The root of an instance of T has one ***vestigial child*** and one ***nonvestigial child***. When generating a distribution D from $\mathcal{D}_{n,t}$, we choose at random which child is vestigial. The vestigial child's subtree consists of single path directly to a sample at depth n. All nodes on this path are chosen to be left or right children at random. The nonvestigial child's subtree is a complete binary tree, whose size will be specified later.

Each leaf of T in the nonvestigial subtree has beneath it another instance of T. This iterative construction proceeds for t layers. All leaves of T in the last layer generate a path directly to a sample. We say that the topmost instance of T is at ***layer*** 1, the instances of T immediately beneath it are at layer 2, and so on. Each instance of T at layer i is chosen to have height $1 + h_i$, with the nonvestigial subtree of height h_i (to

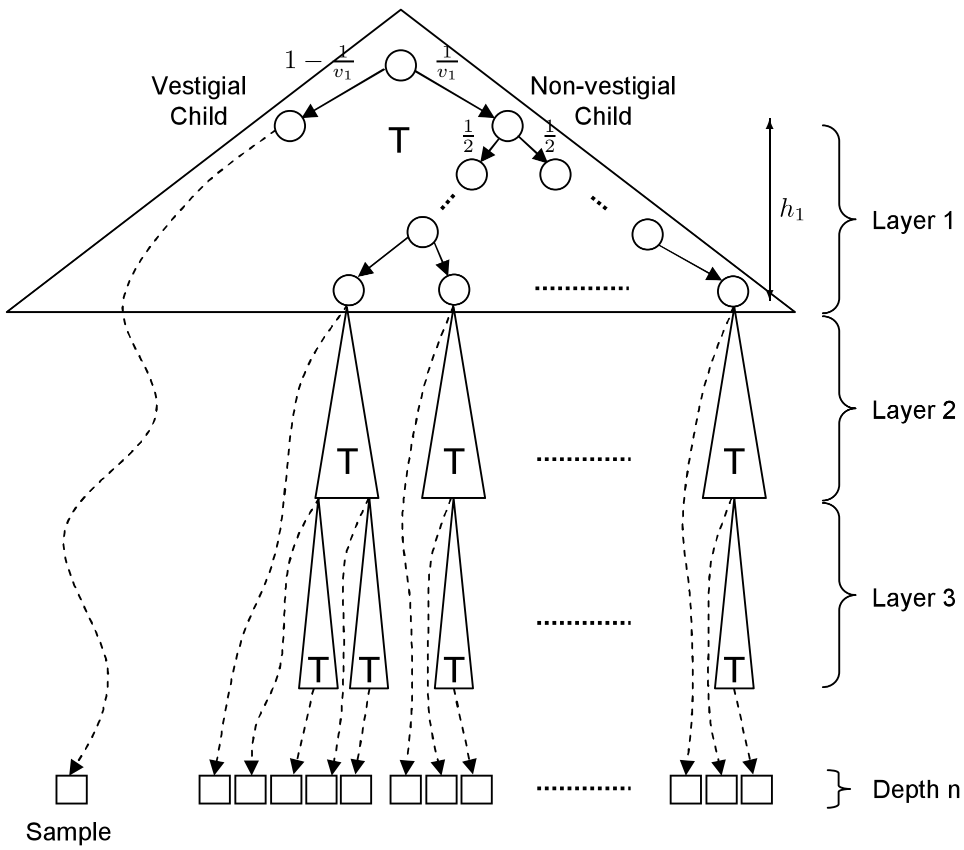

Figure 1: A distribution $D \in \mathcal{D}_{n,t}$, represented as a binary tree. For each instance of the subtree T, one of the root's children is "vestigial" and generates a path directly to a sample at depth n. The other child has beneath it a complete binary tree, underneath which are more instances of T. There are t layers in total.

be determined), and hence the number of nonvestigial leaves in this instance is $\ell_i := 2^{h_i}$.

After the t^{th} layer, there is a final layer $t+1$ that consists solely of vestigal paths down to depth n, with no branching even at the top node. Thus every leaf is at the end of a vestigal path from some layer between 1 and $t+1$, and has depth n as desired.

Thus far we have described D as a tree, but not as a probability space. For every internal node, we specify the probabilities of going to the left and right children (conditioned on having already reached the parent). The probability of a sample (leaf) is then the product of the edge probabilities on the root-to-leaf path. At the root of an instance of T at layer i, the probability that we go down the vestigial path is $1 - \frac{1}{v_i}$, where v_i is to be determined. There is nothing to specify along the vestigial path, because there is no branching. Inside a nonvestigial subtree, the sample chooses between the left and right children uniformly. The result of this construction is that a sample generated from a vestigial node at layer i has probability mass $\left(\prod_{j=1}^{i-1} \frac{1}{v_j \cdot 2^{h_j}}\right) \cdot (1 - \frac{1}{v_i})$.

To summarize, choosing D from $\mathcal{D}_{n,t}$ involves choosing two random features: which child is vestigial for every T root, and how each vestigial path evolves down to level n. As explained above, the server is given a D chosen at random from $\mathcal{D}_{n,t}$ and a coloring ϕ of the leaves of D, while the client is given a sample Y drawn from D. Let $\mathscr{D}$ denote this distribution on triples (D, ϕ, Y); this is the hard input distribution for SC_n. Define $\mathcal{L}_k$ to be the event that the client's string is generated at layer at least k. Define $n_k := n - \sum_{i=1}^{k-1} h_i$. We use the notation $\mathscr{D}_k$ to denote the distribution on strings of length n_k by starting the above construction at layer k. Note that $\mathscr{D}_k$ is a distribution on inputs to SC_{n_k}.

3 Message-Elimination Machinery

In this section, we discuss the message-elimination aspects of our proof, deferring a detailed analysis to Section 4. The general approach of our proof is to take any protocol for the string-color problem, restrict the protocol to t rounds, and then argue that such a protocol must have error $2^{-O(t \lg t)}$. To do so, we argue by contradiction: supposing that the protocol has smaller error, we show how to iteratively eliminate all t rounds, yielding a protocol that solves a nontrivial problem without any communication.

Let A be the expected number of bits communicated by the server per round. Let B the expected total number of bits communicated by the client over all rounds. For simplicity of notation, we will occasionally treat B as its corresponding random variable, not as an expectation; it will be clear from context which meaning of B is intended. The precise values of A and B are irrelevant for our present discussion, and are specified later.

Eliminating a round involves three steps. By adding at most one round we can assume that the client sends the first message.

1. We eliminate the client's message through message switching, at the cost of increasing the server's message. This requires a hard bound on the client's message, which we obtain by keeping track of the expected number of remaining bits to be sent by the client (via the quantity B), and applying a Markov bound. An important observation is that this bound introduces only unilateral error.

2. We restrict ourselves to the probability space in which the sample comes from the nonvestigial child of the current T structure. Because this is a smaller probability space, the ordinary error and B can increase by v_i. However, we can show that the unilateral error does not change because the marginal distribution of the sample does not change (because each vestigial path is chosen uniformly at random).

3. We eliminate the server's message through the round-elimination lemma. The nonvestigial subtree of the T structure naturally defines a collection of subproblems (the T structures in the leaves). A key observation is that the round elimination increases both the ordinary error and the value of B, but not the unilateral error.

In order to apply the round-elimination lemma, we need a hard bound on the server's message length, as well as on the random variable B. These bounds need not be too tight, so we can apply them in the beginning of the proof, at the cost of a small error. Note that this hard bound on B is much looser than the hard bound we apply as part of message switching in each round.

Throughout a round elimination, we need to pay close attention to two quantities: the expected error ε and the expected number of bits sent by the client, B. To handle these two criteria, we define a random variable $Z := \varepsilon + \gamma B$, where γ will be chosen later. (Here, ε and B are treated as random variables.) Our goal is to eliminate a round without increasing Z too much. This implies a good bound on the increase in the error and B, with a tradeoff controlled by γ.

3.1 Imposing Hard Bounds. Suppose we have a protocol for SC_n with no errors. For convenience, we define $s := t^{6t}$. One may think of $1/s$ as roughly the probability of choosing a sample at layer t from a distribution in $\mathcal{D}_{n,t}$. We impose the following hard bounds on this protocol:

- the protocol has t rounds,
- each of the server's messages has at most $\overline{A} := 4stA$ bits, and
- each of the client's messages has at most $\overline{B} := 4stB$ bits.

If one of these bounds is ever violated, the protocol returns an arbitrary answer. We will show that for $t \leq c \cdot \frac{\lg n}{\lg \lg n}$, such a protocol has error at least $\frac{1}{s}$. Notice that the probability any message exceeds its hard bound is at most $\frac{1}{2s}$ by a Markov bound on each message and a union bound over all. Then, the probability that the original protocol exceeds t rounds has to be at least $\frac{1}{2s} = 2^{-O(t \lg t)}$, which is our desired conclusion. Finally, note that the expected number of bits sent by the client cannot increase as a result of this transformation, because we only discard some messages.

3.2 Eliminating the Client's Message. If the client sends at most B bits in expectation, the first message is also bounded by B in expectation. By Markov, we can bound it to $8tB$ bits with probability $1-\frac{1}{8t}$. Note that the error of $\frac{1}{8t}$ that is introduced is unilateral error μ, because the size of the message is a random variable that is a function of the client's input, but *not* of the server's input. Now we can eliminate the client's message by message switching (Lemma 1.1). This change does not introduce error. Also, it does not affect B, because we only postpone sending the client's message.

3.3 Reduction from $\mathsf{SC}^{(\ell)}$ to SC. The following lemma identifies the desired reduction in a T structure:

LEMMA 3.1. *Let Z be an arbitrary positive random variable. Given a protocol P for SC_{n_k}, one can construct a protocol Q for $\mathsf{SC}^{(\ell_k)}_{n_{k+1}}$ with the same unilateral error and* $\mathrm{E}_{\mathscr{D}^{(\ell_k)}_{k+1}}\left[Z^Q \right] \leq v_k \cdot \mathrm{E}_{\mathscr{D}_k}\left[Z^P \right]$.

Proof. The two parties construct an instance of SC_{n_k} starting with an instance of $\mathsf{SC}_{n_{k+1}}^{(\ell_k)}$, and then run protocol P. The server constructs a new distribution D where the root is a new instance of T with ℓ_k leaves. Beneath the leaves in the nonvestigial half of T, we attach the subtrees correspond to the server's ℓ_k inputs. The client prepends the variable i to its sample, which effectively selects among these inputs. The remaining random features are these: which child is vestigial, the vestigial path, and the color of the vestigial sample. We imagine selecting these features randomly through public coins. Then, we can fix these coins deterministically to achieve the same expectation on Z. Note that the client must know which child is vestigial, because it must set the first bit of its sample to point to the nonvestigial subtree.

By construction, if the instance of $\mathsf{SC}_{n_{k+1}}^{(\ell_k),A}$ is distributed according to $\mathscr{D}_{k+1}^{(\ell_k)}$, the induced distribution on instances of $\mathsf{SC}_{n_k}^{A}$ is $\mathscr{D}_k$, conditioned on the event $\mathcal{L}_{k+1}$. Recall that $\mathcal{L}_{k+1}$ is the event that a sample is vestigial at layer at least $k+1$, and that $\Pr[\mathcal{L}_{k+1} \mid \mathcal{L}_k] = \frac{1}{v_k}$. Then

$$\begin{aligned} \mathrm{E}_{\mathscr{D}_{k+1}^{(\ell_k)}}\left[Z^Q\right] &= \mathrm{E}_{\mathscr{D}_k}\left[Z^P \mid \mathcal{L}_{k+1}\right] \\ &\leq \frac{\mathrm{E}_{\mathscr{D}_k}\left[Z^P\right]}{\Pr_{\mathscr{D}_k}\left[\mathcal{L}_{k+1}\right]} = v_k \cdot \mathrm{E}_{\mathscr{D}_k}\left[Z^P\right]. \end{aligned}$$

It remains to analyze the unilateral error. By the above, $\mu^Q_{\mathscr{D}_{k+1}^{(\ell_k)}} = \mu^P_{\mathscr{D}_{k|\mathcal{L}_{k+1}}}$. Now observe that the marginal distribution on the client's sample is uniform under both $\mathscr{D}_k$ and $\mathscr{D}_{k|\mathcal{L}_{k+1}}$, because the vestigial child and the vestigial path are chosen uniformly at random. Because unilateral error depends only on the client's sample, $\mu^P_{\mathscr{D}_k} = \mu^P_{\mathscr{D}_{k|\mathcal{L}_{k+1}}} = \mu^Q_{\mathscr{D}_{k+1}^{(\ell_k)}}$, as required. □

3.4 Eliminating the Server's Message. Observe that our variable Z satisfies $0 \leq Z \leq 1+\gamma\cdot(t\cdot\overline{B})$. We use the following modification of Sen's Round Elimination Lemma (Lemma 1.2), with Alice being the server and Bob being the client:

LEMMA 3.2. *Let Z be a random variable whose range is contained in the interval $[0, Z_{max}]$. Suppose there is a protocol P for $f^{(\ell),A}$ such that Alice's first message has a bits. Then there is a protocol Q for f^B in which Alice's first message is never sent, such that $\mathrm{E}_{\mathscr{D}}\left[Z^Q\right] \leq \mathrm{E}_{\mathscr{D}^{(\ell)}}\left[Z^P\right] + Z_{max}\cdot\sqrt{a/\ell}$. Any unilateral error depending on Bob's input is unchanged.*

Proof sketch. The proof is similar to Sen's, with two differences. First, we can avoid Yao's minimax principle, because we already know the distribution on the inputs. Second, we do not focus specifically on the error, but consider a general random variable Z. Full details are deferred to the full paper. □

4 Proof of the Single Client Lower Bound

To recapitulate our proof outline, we assume that we have an error-free protocol for the string-color problem. We restrict this protocol to t rounds, and then assume that the resulting protocol has ordinary error at most $1/s$. The goal of this section is to eliminate messages from this protocol and carefully analyze the increase in error. We arrive at a contradiction by eliminating all messages, obtaining at a protocol that solves a nontrivial problem without any communication. The conclusion is that the original protocol required t rounds with probability at least $1/2s = 2^{-O(t\lg t)}$. In Section 4.1, we analyze the range of parameters for which this proof is valid, thereby establishing Theorem 1.1.

We now consider iterating the previous process until all rounds have been eliminated. To perform this analysis, we introduce the following more convenient notation:

- b_i is the hard length restriction imposed on the client's i^{th} message.
- a_i is the hard length of the server's i^{th} message, after applying message switching to the client's previous message.
- ε_i is the ordinary error (under $\mathscr{D}_{i+1}$) after eliminating the server's i^{th} message.
- μ_i is the unilateral error (under $\mathscr{D}_{i+1}$) after eliminating the server's i^{th} message.
- B_i is the expected remaining number of bits that the client has left to send, after eliminating the server's i^{th} message.

LEMMA 4.1. *If $h_i \geq 8ti\cdot\left(1+\frac{2}{t}\right)^{i-1}\cdot\left(\prod_{j=1}^{i-1} v_j\right)\cdot B + 10t^2 + \lg A$, the message-elimination techniques from above lead to the following bounds:*

$$\begin{aligned} &b_i \leq 8ti\cdot B_{i-1}, && a_i \leq 2^{b_i}\cdot\overline{A}, \\ &\mu_i \leq \frac{i}{8t}, && \varepsilon_i \leq \frac{(t+2)^i}{s}\cdot\left(\textstyle\prod_{j=1}^{i} v_j\right), \\ &B_i \leq \left(1+\frac{2}{t}\right)^i\cdot\left(\textstyle\prod_{j=1}^{i} v_j\right)\cdot B. \end{aligned}$$

Proof. The proof is by induction, the base case $i=0$ being that $b_0 = a_0 = \mu_0 = 0$, $\varepsilon_0 = \frac{1}{s}$, $B_0 \leq B$. Now we consider $i \geq 1$. To obtain a hard limit on the client's i^{th} message, we assume (pessimistically) that it has expected size B_{i-1}. That is, we assume that all of the client's remaining bits are used in this next message, but also assume that the number of remaining bits does not decrease. Using the Markov bound, we impose a hard limit of $8tB_{i-1}$. This introduces unilateral error $\frac{1}{8t}$, and is the only source of unilateral error. Hence we see that $\mu_i \leq \mu_{i-1} + \frac{1}{8t} \leq \frac{i}{8t}$. The client's i^{th} message also has to include message $i-1$ because message switching only postpones the sending of a message. Thus $b_i \leq b_{i-1} + 8tB_{i-1} \leq 8tiB_{i-1}$. Now, applying message switching, we obtain the size of the server's message: $a_i \leq 2^{b_i}\overline{A}$.

When performing the i^{th} round elimination, we choose $\gamma = \frac{t^i}{sB}$. Consequently, we have $Z_{\max} = 1 + \gamma t\overline{B} = 1 + \frac{t^i}{sB}t(4stB) \leq 5t^{i+2}$. Applying the reduction from $\mathsf{SC}^{(\ell)}$ to SC increases Z by a factor of v_i. Lemma 3.2 increases Z additively by $Z_{\max} \cdot \sqrt{(a_i+1)/\ell_i}$, so we obtain that

$$\varepsilon_i + \gamma \cdot B_i \;\leq\; v_i(\varepsilon_{i-1} + \gamma B_{i-1}) + 5t^{i+2} \cdot \sqrt{(a_i+1)/\ell_i}.$$

To analyze this, we first observe that

$$\begin{aligned}
&\varepsilon_{i-1} + \gamma B_{i-1} \\
&\leq \Big(\textstyle\prod_{j=1}^{i-1} v_j\Big)\left(\frac{(t+2)^{i-1}}{s} + \gamma \cdot \left(\left(1+\frac{2}{t}\right)^{i-1} B\right)\right) \\
&\leq \Big(\textstyle\prod_{j=1}^{i-1} v_j\Big) \cdot \frac{(t+2)^{i-1}}{s} \cdot (1+t).
\end{aligned}$$

Next, recall that $\overline{A} = 4stA$, $\ell_i = 2^{h_i}$, and $s = t^{6t} = 2^{6t\lg t}$. Thus we obtain that

$$\begin{aligned}
\lg \frac{a_i+1}{\ell_i} \leq\; & \big(b_i + \lg(4stA) + 1\big) \\
& - \left(8ti \cdot \left(1+\frac{2}{t}\right)^{i-1} \cdot \Big(\textstyle\prod_{j=1}^{i-1} v_j\Big) \cdot B + 10t^2 + \lg A\right) \\
& \leq \big(\lg(4stA) + 1\big) - \big(10t^2 + \lg A\big) \;\leq -9t^2.
\end{aligned}$$

This inequality implies that $5t^{i+2} \cdot \sqrt{(a_i+1)/\ell_i} \leq 5t^{i+2} \cdot 2^{-4t^2} \leq 2^{-3t^2} \leq \frac{1}{s}$. In conclusion, we obtain that

$$\varepsilon_i + \gamma \cdot B_i \;\leq\; \Big(\textstyle\prod_{j=1}^{i} v_j\Big) \cdot \left(\frac{(t+2)^{i-1}}{s}\right) \cdot (2+t).$$

Because ε_i and B_i are both nonnegative, this inequality immediately implies the desired bound on ε_i. Recalling that $\gamma = \frac{t^i}{sB}$, we have

$$B_i \;\leq\; \Big(\textstyle\prod_{j=1}^{i} v_j\Big) \cdot \frac{(t+2)^i}{\gamma s} \;=\; \Big(\textstyle\prod_{j=1}^{i} v_j\Big)\left(1+\frac{2}{t}\right)^i \cdot B,$$

as required. □

After eliminating all messages, we obtain a protocol for the problem $\mathsf{SC}_{n_{t+1}}$ which sends no messages. The distribution under consideration is $\mathscr{D}_{t+1}$: the support is just one string, of a random color. Clearly, the client cannot guess the color with probability more than $\frac{1}{2}$ without communication. However, Lemma 4.1 shows that $\mu_t \leq \frac{1}{8}$. As explained in the following section, we may choose our parameters such that $\varepsilon_i < \frac{1}{4}$. Thus our final protocol sends no messages and yet the client correctly announces the color of his string with probability strictly greater than $\frac{1}{2}$, which is a contradiction. This contradicts our assumption that the t-round protocol has error at most $1/s$. This completes the framework of our proof. To complete the proof of Theorem 1.1, we now specify the various parameters of our proof.

4.1 Fixing the Parameters. We now define the three parameters left unspecified in the construction of $\mathcal{D}_{n,t}$, namely v_i, h_i, and B. First we choose h_i to satisfy the hypothesis of Lemma 4.1:

$$h_i \;:=\; 8ti \cdot \left(1+\frac{2}{t}\right)^i \cdot \left(\prod_{j=1}^{i-1} v_j\right) \cdot B + 10t^2 + \lg A.$$

We now explain why this is an appropriate choice. All distributions in $\mathcal{D}_{n,t}$ have the same entropy, which we denote by H. The hypothesis of Theorem 1.1 is that the expected total number of bits sent by the client is $O(t \cdot (H+1))$. Our proof refers to this quantity as B, so we must ensure that $B \geq c_2 \cdot t \cdot (H+1)$, for an arbitrary constant c_2. Because the distributions in $\mathcal{D}_{n,t}$ are on n-bit strings, we must also ensure the heights of the layers are valid, i.e., $\sum_{i=1}^{t}(1+h_i) \leq n$. We begin by computing H, which leads to our choice of v_i and B.

$$\begin{aligned}
H &= \sum_{i=1}^{t} (\text{entropy at layer } i) \cdot \Pr[\,\text{sample's layer} \geq i\,] \\
&\leq \sum_{i=1}^{t} \left(\frac{1}{v_i} \cdot h_i + 1\right) \cdot \prod_{j=1}^{i-1} \frac{1}{v_j} \\
&= \sum_{i=1}^{t} \left(\frac{8ti\,(1+2/t)^i \cdot B}{v_i} + \frac{10t^2 + \lg A}{\prod_{j=1}^{i} v_j} + \frac{1}{\prod_{j=1}^{i-1} v_j}\right)
\end{aligned}$$

We now choose $B := 4c_2t(1+\frac{\lg A}{t^2})$. In order to obtain a small coefficient of $(1+\frac{\lg A}{t^2})$ on the right-hand side, we choose $v_i := 32c_2t^4(1+\frac{2}{t})^i$. Hence we obtain that

$$\begin{aligned}
H &< \sum_{i=1}^{t} \left(\frac{(1+(\lg A)/t^2)}{t} + \frac{t^2 + \lg A}{t^4}\right) + 1 \\
&< \left(2 + \frac{1}{t}\right) \cdot \left(1 + \frac{\lg A}{t^2}\right).
\end{aligned}$$

Thus $H+1 < (3+\frac{1}{t}) \cdot (1+\frac{\lg A}{t^2})$ and $B \geq c_2 t(H+1)$. This establishes the desired bound on B.

We now focus on the constraint on the heights of the layers. Note that $\prod_{j=1}^{i-1} v_j = \prod_{j=1}^{i-1} 32c_2t^4(1+\frac{2}{t})^j = 2^{O(i \lg t)}$ because $(1+\frac{2}{t})^j < e^2$ for $1 \leq j \leq t$. Thus,

$$\begin{aligned} \sum_{i=1}^{t}(1+h_i) &\leq \sum_{i=1}^{t} \left(1+8ti\, e^2\, 2^{O(i \lg t)}\, B + 10t^2 + \lg A\right) \\ &= 2^{O(t \lg t)} \cdot \left(1+\frac{\lg A}{t^2}\right) + O(t^3 + t \lg A) \\ &= 2^{O(t \lg t)} \cdot \lg A, \end{aligned}$$

which is less than n for $t \leq c \cdot \frac{\lg n}{\lg \lg n}$ and $\lg A = O(n^{1-\epsilon})$, for any constant $\epsilon > 0$ and for an appropriate constant $c > 0$ (dependent on ϵ).

Finally, now that v_i has been chosen, we must verify our claim from the previous section that $\varepsilon_i < \frac{1}{4}$. By Lemma 4.1, we obtain that $\varepsilon_i \leq (t+2)^i \cdot \left(\prod_{j=1}^{i} v_j\right)/s$. Hence,

$$\begin{aligned} \varepsilon_i &\leq 2^{i \lg(t+2)} \cdot \left(\prod_{j=1}^{i} 32c_2t^4\left(1+\frac{2}{t}\right)^j\right) \cdot 2^{-6t \lg t} \\ &= 2^{i \lg t + O(i)} \cdot \left(\prod_{j=1}^{i} 2^{4 \lg t + O(1)}\right) \cdot 2^{-6t \lg t} \\ &= 2^{i \lg t + 4i \lg t + O(i) - 6t \lg t} = 2^{O(i) - t \lg t}, \end{aligned}$$

which is less than $\frac{1}{4}$ for sufficiently large t.

5 The Lower Bound for Multiple Clients

Assume $k \leq n^c$ and let $t = c \cdot \frac{\lg k}{\lg \lg k}$, for a small enough constant c. For each $i \in \{1, \ldots, k\}$, the server receives D_i chosen independently at random from $\mathcal{D}_{n,t}$, and client i receives a sample y_i from D_i. Clearly, the entropy of the joint distribution is $H(D) = kH(D_i)$. Let the bound on the total communication from the clients be $c_1 k H(D_i)$. Consider the expected number of bits sent by each client. By Markov, there is a set of $\frac{k}{2}$ ***light clients***, each of which sends at most $2c_1 H(D_i)$ bits in expectation.

We now argue that, if a client i sends $O(H(D_i)+1)$ bits in expectation in a multiclient scenario, the single-client lower bound applies. Indeed, our previous lower bound allowed for nonuniformity, so the client and the server can nonuniformly fix some good setting of (D_j, y_j) for $j \neq i$, and simulate the other clients without communication (because both the client and the server know the assumed inputs of the other clients). By a bicriterion averaging argument, there is a setting of the other (D_j, y_j) which simultaneously doubles both the expected communication from client i and the probability that client i needs t rounds. Thus, we have probability $2^{-O(t \lg t)}$ that client i cannot finish in t rounds. For a sufficiently small c, this probably is at least $\frac{1}{k}$.

Now we want to show that, with constant probability, some light client cannot finish in t rounds. We prove by induction that the probability the first i light clients can all finish in t rounds is at most $(1-\frac{1}{k})^i$. Setting $i = \frac{k}{2}$ then proves the theorem, as $(1-\frac{1}{k})^{k/2} = e^{-1/2} - O(\frac{1}{k}) = \Omega(1)$. Assume the induction hypothesis for i, and let us prove it for $i+1$. If the probability of the first i light clients finishing in t rounds is already at most $(1-\frac{1}{k})^{i+1}$, we are done. Otherwise, condition on this event. Because $i \leq k$, the probability of the event is $e^{-1} - O(\frac{1}{k}) = \Omega(1)$, so the expected number of bits sent by the $(i+1)^{\text{st}}$ light client increases by at most a constant factor when conditioning. By the previous paragraph, we have probability at most $1-\frac{1}{k}$ that the $(i+1)^{\text{st}}$ light client finishes in t rounds. This proves the claim, concluding the proof of Theorem 1.2.

References

[1] M. Adler, *Collecting Correlated Information from a Sensor Network*, Proc. 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) 2005.

[2] A. Aaron and B. Girod, *Compression with side information using turbo codes*, Proc. DCC 2002.

[3] M. Adler and B. Maggs, *Protocols for asymmetric communication channels*, Journal of Computer and System Sciences 63(4): 573–596, Dec 2001. Special issue dedicated to FOCS 1998.

[4] J. Bajcsy and P. Mitran, *Coding for the Slepian-Wolf problem with turbo codes*, Proc. GlobeCom 2001.

[5] P. Bose, D. Krizanc, S. Langerman, and P. Morin, *Asymmetric communication protocols via hotlink assignments*, Theory of Computing Systems 36(6): 655-661, 2003. Special issue dedicated to the IX$^{\text{th}}$ International Colloquium on Structural Information and Communication Complexity (SIROCCO 2002).

[6] A. Chakrabarti and O. Regev, *An optimal randomised cell probe lower bound for approximate nearest neighbour searching*, Proc. 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2004.

[7] J. Chou, D. Petrovic, and K. Ramchandran, *A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consuumption in Sensor Networks*, Proc. INFOCOM 2003.

[8] T. Coleman, A. Lee, M. Medard, and M. Effros, *On some new approaches to practical Slepian-Wolf compression inspired by channel coding*, Proc. DCC'04.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, 1991.

[10] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger, *Infranet: Circumventing Censorship and Surveillance*, Proc. 11th USENIX Security Symposium, 2002.

[11] L. M. Feeney and M. Nilsson, *Investigating the energy consumption of a wireless network interface in an ad hoc network*, Proc. INFOCOM 2001.

[12] J. Garcia-Frias and Y. Zhao, *Compression of correlated binary sources using turbo codes*, IEEE Communications Letters 5(10): 417-419, Oct 2001.

[13] J. Garcia-Frias and W. Zhong, *LDPC codes for compression of multiterminal sources with hidden Markov correlation*, IEEE Communications Letters 7(3): 115-117, Mar 2003.

[14] S. Ghazizadeh, M. Ghodsi, and A. Saberi, *A New Protocol for Asymmetric Communication Channels, Reaching Lower Bounds*, Scientia Iranica 8(4), 2001.

[15] E. S. Laber and L. G. Holanda, *Improved bounds for asymmetric communication protocols*, Information Processing Letters 83(4): 205–209, 2002.

[16] D. A. Huffman, *A method for the construction of minimum redundancy codes*, Proc. IRE 40(10): 1099–1101, 1952.

[17] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, 1996.

[18] C. Lan, A. Liveris, K. Narayanan, Z. Xiong, and C. Georghiades, *Slepian-Wolf coding of multiple M-ary sources using LDPC codes*, Proc. DCC 2004.

[19] A. Liveris, Z. Xiong, and C. Georghiades, *Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes*, Proc. DCC 2003.

[20] A. Liveris, Z. Xiong and C. Georghiades, *Compression of binary sources with side information at the decoder using LDPC codes*, IEEE Communications Letters 6(10): 440-442, Oct 2002.

[21] A. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades, *Slepian-Wolf coding of three binary sources using LDPC codes*, Proc. Intl. Symp. Turbo Codes and Related Topics 2003.

[22] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson, *On data structures and asymmetric communication complexity*, Journal of Computer and System Sciences 57(1): 37–49, 1998.

[23] S. Pradhan, J. Kusuma, and K. Ramchandran, *Distributed compression in a dense microsensor network*, IEEE Signal Processing Mag., 19: 51-60, Mar 2002.

[24] D. Schonberg, S. Pradhan, and K. Ramchandran, *Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources*, Proc. DCC 2004.

[25] P. Sen, *Lower bounds for predecessor searching in the cell probe model*, Proc. 18th IEEE Conference on Computational Complexity (CCC) 2003.

[26] C. E. Shannon, *A mathematical theory of communication*, Bell System Technical Journal, 27: 379–423 and 623–656, 1948.

[27] D. Slepian and J. K. Wolf, *Noiseless encodings of correlated information sources*, IEEE Trans. on Information Theory, IT-19: 471–480, July 1973.

[28] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiades, *Design of Slepian-Wolf codes by channel code partitioning*, Proc. DCC 2004.

[29] John Watkinson, *New Protocols for Asymmetric Communication Channels*, Master's Thesis, U. Toronto, 2000.

[30] John Watkinson, Micah Adler, and Faith Fich, *New Protocols for Asymmetric Communication Channels*, Proc. 8th Intl. Colloquium on Structural Information and Communication Complexity (SIROCCO) 2001.

[31] Zixiang Xiong, Angelos D. Liveris, and Samuel Cheng, *Distributed Source Coding for Sensor Networks*, IEEE Signal Processing Magazine, Sep 2004.