

Low-Delay Distributed Source Coding for Time-Varying Sources with Unknown Statistics

Fangzhou Chen
Department of ECE
The Ohio State University
Columbus, OH 43210, USA
Email: chen.1953@osu.edu

Bin Li
Coordinated Science Lab
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
Email: lib@illinois.edu

Can Emre Koksal
Department of ECE
The Ohio State University
Columbus, OH 43210, USA
Email: koksal@ece.osu.edu

Abstract—We consider a system in which two nodes take correlated measurements of a random source with time-varying and unknown statistics. The observations of the source at the first node are to be losslessly replicated with a given probability of outage at the second node, which receives data from the first node over a constant-rate channel. We develop a system and associated strategies for joint distributed source coding (encoding and decoding) and transmission control in order to achieve low end-to-end delay. Slepian-Wolf coding in its traditional form cannot be applied in our scenario, since the encoder requires the joint statistics of the observations and the associated decoding delay is very high. We analytically evaluate the performance of our strategies and show that the delay achieved by them are order optimal, as the conditional entropy of the source approaches to the channel rate. We also evaluate the performance of our algorithms based on real-world experiments using two cameras recording videos of a scene at different angles. Having realized our schemes, we demonstrated that, even with a very low-complexity quantizer, a compression ratio of approximately 50% is achievable for lossless replication at the decoder, at an average delay of a few seconds.

Index Terms—Lossless distributed source coding, universal algorithms, delay optimal control, heavy-traffic analysis

I. INTRODUCTION

In many applications, multiple nodes take measurements from the same source to be combined later in order to obtain a high resolution representation of the source. In order to achieve that, nodes encode their digitized observations and share information for the observations to be replicated at a common location. Lossless distributed source coding aims to encode the observations in a way to minimize the rate of exchanged information for all observations to be perfectly replicated at a certain location. The nodes exploit the correlations in their observations to build an efficient code. Following the seminal work by Slepian and Wolf [17], there has been a vast interest in lossless distributed source coding (DSC) including applications in quantum key distribution [12], distributed video coding [8] and wireless sensor networks [20].

We address the lossless distributed source coding problem for a pair of nodes, observing a random source with time-varying statistics, unknown to the nodes before the session starts. Our objective is to minimize the delay for the second node to losslessly replicate the observations of the first node, subject to a given desired probability of outage. Communication from the first node to the second node occurs over a finite-

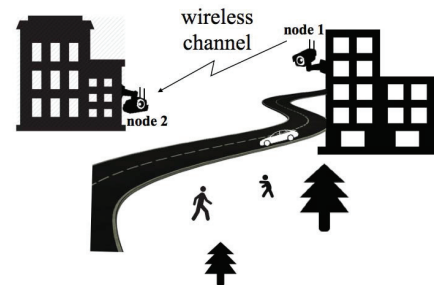


Fig. 1: A sample scenario for our problem

rate channel. Slepian-Wolf coding in its traditional form cannot be applied in our scenario, since, with traditional SW, the encoder requires the joint statistics of the observations causally to encode information. The alternative is to encode across a time period, long enough to exploit long-term variations of the source. However, with this approach, the corresponding decoding delay is very high. To that end, we develop a system and associated novel strategies for joint distributed source coding (encoding and decoding) and transmission control in order to achieve low end-to-end delay. We evaluate the performance of our strategies both analytically and via real-world experimentation. We first derive upper and lower bounds on the expected delay. Our bounds show that the delay achieved by our strategies are order optimal as the conditional entropy of the source (given the observation at the second node) approaches to the channel capacity. Next, we use two cameras recording videos of a common scene at different angles to obtain experimental data. After evaluating the possible joint source distributions based on the observations, we apply our schemes and show major improvements in end-to-end delay over existing traditional Slepian-Wolf based coding schemes. The *analytical* results show us that the delay performance of our schemes are very close to that of a highly-optimistic imaginary scenario, in which the joint distribution of the source observations are causally available to encoder, without seeing the observation at the decoder. The *experimental* results show that, even if we use highly coarse quantization for the source statistics, the average data rate at which an encoder shares information to achieve lossless replication at the decoder is reduced by $\sim 50\%$ (compression ratio), at an average end-to-end delay of 6-9 secs.

Sample scenario: In Fig. 1, we provide an example for a

typical setting¹ that we consider in this paper. Here, two security surveillance cameras observe the action in the environment, which is time-varying and uncertain. Therefore, the joint distribution of the observations of the two cameras is not causally available at these two locations. One of the nodes, holding a camera would like to replicate the video taken by the other camera to increase the resolution, in order to enhance the detection performance of an intruder, for example. The brute force approach would be to have the whole video transmitted to the replicator node over the wireless channel. However, this would lead to a waste of communication resources, especially if the wireless channel has a limited capacity. Instead, the first node can exploit the existence of highly correlated observation at the second (replicating) node to send information merely “sufficient” for its video to be replicated. This process is known as lossless distributed source coding. In security applications, delay is of critical importance. Therefore, traditional schemes that are based on the utilization of the long-term statistical regularities are not viable. Furthermore, due to time-varying statistics (non-stationarity), the knowledge of the joint statistics at the encoder is also not a valid assumption, which calls for *universal* solutions for coding.

Background, Related Work, and our Contributions: Lossless distributed source coding was first introduced by Slepian and Wolf in [17]. There, two independently and identically distributed (i.i.d.) random source sequences, $\{X_i\}_{i=1}^{\infty}, \{Y_i\}_{i=1}^{\infty}$ with joint cumulative distribution function (cdf) $F_{XY}(x, y)$ are observed at the encoder and the decoder, respectively. The objective is to reconstruct sequence $\{X_i\}_{i=1}^{\infty}$ losslessly at the decoder. Instead of encoding $\{X_i\}_{i=1}^{\infty}$ with its full entropy rate $H(X)$, SW coding enables lossless replication at an encoding rate $R_X = H(X|Y)$, regarding $\{Y_i\}_{i=1}^{\infty}$ as side information at the decoder. To achieve this rate, SW coding (i) encodes the source sequence across infinitely many blocks to achieve an arbitrarily low probability of decoding error; (ii) requires the knowledge of the joint cdf, F_{XY} at the encoder. These two assumptions make direct application of SW coding inappropriate in some delay-sensitive applications (e.g. live video meeting, online video streaming etc.) or for cases in which the joint statistics is unknown, and/or time-varying/non-stationary in certain situations (e.g. the encoder and decoder are moving while communicating).

These issues were addressed in a variety of studies, following [17]. Many practical coding schemes for SW problem have been proposed, e.g. [15], [19]. And most of them are based on channel codes [18], especially low-density parity check (LDPC) codes [11], [13]. Nevertheless, most of them still assume perfect knowledge of joint cdf. Csiszar and Korner first extended SW coding to achieve universality [4], where neither encoding nor decoding depends on source statistics. They also analyzed the finite-block behavior and provided the universally attainable error exponent when encoding rates are within the SW-region, i.e., the region of rates for which

lossless replication of sources is possible with arbitrarily low probability of outage as the block size goes to infinity. In [2], end-to-end delay is studied in a SW coding setup. There, decoding error exponent is derived for a given end-to-end delay. If a feedback channel from the decoder to the encoder is present, [14] proposed a scheme under lack of the knowledge for the joint cdf. There, by carefully choosing a sequence of codes, the average encoding rate is minimized over each frame at the encoder. Similarly, [9] exploited the feedback channel to send some information to the encoder for a better performance. With unknown statistics, [16] studied the outage capacity of the system and [6] proposed a compound source model to achieve arbitrarily low probability of outage. In another direction, [5] proposed universal incremental SW coding, in which an incremental transmission and a universal sequential decision test is applied by the encoder and decoder, respectively. The system developed requires the availability of ACK/NAK feedback from decoder to encoder.

None of the above studies addressed the time varying and unknown source statistics in the context of delay minimization, without an active use of a feedback channel. Indeed, if the prior knowledge of the source statistics is not available, it may not be possible to feed back the joint statistics without actually feeding back the actual observation sequence itself. With that motivation, the main contributions of our paper can be listed as follows:

- (1) We extend the existing studies on universal distributed source coding by integrating time-varying and unknown joint statistics, the use of finite-capacity channel from the encoder to the decoder, and minimization of end-to-end delay.
- (2) We specify two different classes of joint encoding, decoding, and transmission control schemes, Wait-to-Encode and Wait-to-Decode and develop a scheme in each class to achieve low-delay in universal distributed source coding.
- (3) We derive upper and lower bounds on the performance of our schemes. While the bounds are valid in any regime, in the heavy traffic limit as the long-term average source rate converges to the channel capacity, we show that our schemes achieve optimal delay scaling. To achieve that, we develop and utilize new techniques in heavy-traffic analysis of queues. We also point out a phase-transition phenomenon on the achievable delay scaling to show that even a minor degradation in the knowledge of joint statistics leads to a different scaling regime.
- (4) We implement our schemes in a setup that involves two cameras recording videos of a common scene at different angles. This setup enabled the real-world experimental evaluation of the performance of our schemes. We demonstrated that, even with a low-complexity quantizer, a compression ratio of $\sim 50\%$ is achievable for lossless replication at the decoder, at an average delay of a few seconds.

II. SYSTEM MODEL

In this paper, we use boldface to represent vectors, upper case to represent random variables and vectors, and lower case

¹In fact, the illustrated scenario is the very case we evaluate in our experimental observations.

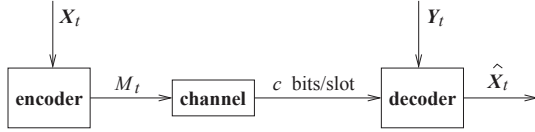


Fig. 2: System model

to represent realizations of random variables and vectors, or deterministic parameters.

First, we introduce our **source model**. In our system, there are two nodes that take correlated measurements of a random source. Time is slotted and a single source symbol is taken by each node in each time slot. We further group time slots into blocks of size n time slots and denote the random measurement taken by the nodes in block t as \mathbf{X}_t and \mathbf{Y}_t , respectively. Hence, we will refer to the nodes as node X and node Y in the sequel. Source symbols are discrete, taking on values from associated finite sets \mathcal{X} and \mathcal{Y} , for node X and node Y , respectively. We denote the joint cumulative distribution function (cdf) of the observation in block t nodes with $F_{\mathbf{X}_t \mathbf{Y}_t}(\mathbf{x}, \mathbf{y})$, and the associated probability mass function (PMF) with $P_{\mathbf{X}_t \mathbf{Y}_t}(\mathbf{x}, \mathbf{y})$. We assume the joint statistics of the blocks to be time varying from one block to another, but to remain constant within each block (analogous to block fading models in channel coding) of n symbols, large enough to invoke random source coding arguments. Also, the symbols observed by node X and Y are i.i.d. in each block:

$$F_{\mathbf{X}_t \mathbf{Y}_t}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^n F_{X_t Y_t}(x_i, y_i),$$

$$P_{\mathbf{X}_t \mathbf{Y}_t}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^n P_{X_t Y_t}(x_i, y_i),$$

where $F_{X_t Y_t}(x, y)$ and $P_{X_t Y_t}(x, y)$ denote the joint cdf and the associated PMF of a symbol in block t , respectively. In the sequel, we simply use $F_{(t)}(x, y)$ and $P_{(t)}(x, y)$. We further assume that $F_{X_t Y_t}(x, y)$ takes values from a finite set \mathcal{F} , of possible joint cdfs. Without loss of generality, we group the joint cdfs in \mathcal{F} into m groups as follows: $\mathcal{F} = \{F_{11}, \dots, F_{1l_1}; F_{21}, \dots, F_{2l_2}; \dots; F_{m1}, \dots, F_{ml_m}\}$, where each group, $\{F_{ij}\}_{j=1}^{l_i}$, of joint cdfs have identical marginal cdfs, i.e., $F_i(x) \triangleq F_{i1}(x, \infty) = \dots = F_{il_i}(x, \infty)$, $F_i(y) \triangleq F_{i1}(\infty, y) = \dots = F_{il_i}(\infty, y)$ for all $i, 1 \leq i \leq m$. Since nodes X and Y merely have their own observations \mathbf{X}_t and \mathbf{Y}_t , they only have the knowledge of the marginals $F_{X_t}(x)$ and $F_{Y_t}(y)$, available at the end of block t , but they do not have any knowledge of $F_{X_t Y_t}(x, y)$ beyond the possible group that it belongs to \mathcal{F} . However, the nodes have the knowledge of the PMF, $P_F(F_{ij})$ of each possible joint cdf of the node. Note that, the existence of such a PMF implies a doubly stochastic, stationary, and ergodic structure in our source model: the joint cdf in each block t is chosen at random, i.i.d. with probabilities parameterized by PMF $P_F(F_{ij})$. Then the source symbols are chosen at random, according to the associated joint cdf F_{ij} , chosen by the PMF.

Next, we present the **system** that we consider, as illustrated

in Fig. 2. Node X is connected to node Y via a noiseless channel with constant transmission rate c bits per slot. Node X (source) encodes the observed symbols $\{\mathbf{X}_t\}$ into a string of bits, which we refer to as the message. We denote the message created in block t with M_t . For each block, node X and Y also (possibly) exchange the indices of the marginal distributions (between 1 and m)², i.e., node X sends the index of \mathbf{X}_t and node Y sends the index of \mathbf{Y}_t . Note that, to form a message, the encoder can possibly combine multiple blocks of source symbols. Consequently, there exist times in which the encoder chooses to make further observations to combine with the existing ones and not encoding the current block at the time. For such blocks, a blank message is generated. For instance, in traditional SW coding, the encoder waits to observe infinitely many blocks of symbols that are encoded jointly to take advantage of long-term statistical averaging. The size of the string dictates the instantaneous rate of the encoder, denoted by R_t bits/slot. Thus, the total number of bits in message $M_t = nR_t$. In case of a blank message after block t , $R_t = 0$ for that block. We ignore the number of bits the encoder uses to encode the index of \mathbf{X}_t in calculating the rate (i.e., $\log_2 m \ll nR_t$ for all non-blank messages). These messages are transmitted over the channel to be (source) decoded by the decoder at node Y . The decoder combines the received messages with its own observation sequence $\{\mathbf{Y}_t\}$ (as well as the indices of the marginals of blocks \mathbf{X}_t if available) in order to *losslessly* decode $\{\mathbf{X}_t\}$. We denote the decoded sequence with $\{\hat{\mathbf{X}}_t\}$. If for certain block t , $\hat{\mathbf{X}}_t \neq \mathbf{X}_t$, we say that an outage occurred for block t .

We refer to a strategy as a method that jointly selects the encoder and the decoder. In particular, a strategy, parameterized with π , chooses the mapping from the sequence of blocks $\{\mathbf{X}_t\}$ observed thus far to the message M_t , and the mapping from the received sequence $\{M_t\}$ to the decoded blocks $\{\hat{\mathbf{X}}_t\}$, at the end of each block. The set of all strategies is denoted with Π . Moreover, we do not impose any limitation on the strategy space such as stationarity or ergodicity.

Finally, we provide the **delay model**. We measure the delay experienced by a block of source symbols as the time elapsed between the slot that the first source symbol is observed and the slot that all the symbols of the block is decoded. There are three different components of the delay. Firstly, for a given strategy π , a block may experience a delay, $W_E^{(\pi)}(t)$, at the encoder. This is due to the fact that the encoder decides to group the symbols of the block with the symbols of the subsequent blocks. Next, the messages (encoded symbols) need to wait to be transmitted over the channel, since the channel has a finite rate. For instance, if the message at time t has a rate $R_t = 2$ Mbits/slot and the channel has a rate $c = 1$ Mbits/slot, then it takes the message (encapsulating all blocks encoded) at least $2n$ slots (or 2 blocks) to be transmitted over the channel, even if there are no other messages in transmission at the time the message generated at time t arrives

²This consumes a negligible amount of resources, compared to the size of the message, since n is assumed to be large.

at the transmission queue. We denote the transmission delay associated with a strategy π with $W_C^{(\pi)}(t)$. Lastly, depending on the encoder strategy, the decoder may choose to accumulate further information on a source block through future messages and thus defer the decoding decision until later. We denote the decoding delay associated with strategy π with $W_D^{(\pi)}(t)$. The overall delay experienced in the system with strategy π is thus, $W^{(\pi)}(t) = W_E^{(\pi)}(t) + W_C^{(\pi)}(t) + W_D^{(\pi)}(t)$.

We finalize the section noting that, systems studied in [14] and [2] can be regarded as two special cases of our system.

III. PROBLEM STATEMENT

In this paper, our objective is to develop strategies that minimize the end-to-end delay observed by the source, while keeping the rate of blocks experiencing outage below a certain desired threshold $\epsilon \in (0, 1)$. This goal can be achieved by solving the following problem:

$$\begin{aligned} \min_{\pi \in \Pi} \quad & \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[W_E^{(\pi)}(t) + W_C^{(\pi)}(t) + W_D^{(\pi)}(t)] \quad (1) \\ \text{s.t.} \quad & \mathbb{P}\{\tilde{\mathbf{X}}_t \neq \mathbf{X}_t\} \leq \epsilon, \forall t = 1, 2, \dots, \infty, \end{aligned}$$

where the expectation is taken over the PMF, $P_F(F_{ij})$, of the source cdfs and the probability of outage is dictated by the strategy π , as well as $P_F(F_{ij})$. Note that, while the source is stationary and ergodic, the strategies need not be stationary nor ergodic. One thing to be careful about our formulation is that, the outage probability is imposed on *every single* block individually, rather than on average.

Illustrative scenarios:

(1) *Known joint cdfs*: It is well-known that if the joint cdf, $F_{(t)}(x, y)$, of the source were known at the encoder, then one could apply SW encoding and decoding [17] (based on random binning and typicality decoding) on a block-by-block basis. Thus, the rate of the message M_t would be the conditional entropy, $H_{(t)}(X|Y)$, of the source associated with joint cdf $F_{(t)}(x, y)$, which would lead to a long-term average encoding rate of

$$\mathbb{E}[H_{(t)}(X|Y)] = \sum_{1 \leq i \leq m} \sum_{1 \leq j \leq l_i} P_F(F_{ij}) \cdot H_{ij}(X|Y).$$

where $H_{ij}(X|Y)$ is the conditional entropy given F_{ij} .

With this approach, an arbitrarily low probability of outage can be achieved as $n \rightarrow \infty$ and thus the constraint is met. The encoding delay is merely a single block for all t , since each block is immediately encoded. The decoding delay is 0 for all t , since each block is immediately decoded. Thus, the only component of the delay experienced is the transmission delay, which is finite if $c > \mathbb{E}[H_{(t)}(X|Y)]$.

(2) *Accumulate and encode*: Without the knowledge of the joint cdfs, one possibility is to accumulate infinitely many blocks at the encoder and encode them jointly. That way, node X can exploit the law of large numbers as the empirical PMF of the source statistic, converges to $P_F(F_{ij})$ with probability 1. Thus, the situation becomes that of known joint statistics, and the long-term average encoding rate of $\mathbb{E}[H_{(t)}(X|Y)]$ can still

be achieved at an arbitrarily low decoding error probability. However, clearly the encoding delay will be arbitrarily large with this approach and it cannot be a viable solution for our problem.

(3) *Block-by-block encoding*: In the other extreme, where each block is encoded separately, one has to pick the encoding rate large enough to ensure the outage constraint is met in each block. For instance, to achieve arbitrarily low probability of outage, the encoding rate has to be picked as

$$R_t = \max_{1 \leq i \leq m, 1 \leq j \leq l_i} H_{ij}(X|Y) \triangleq H_{\max}(X|Y).$$

While the encoding delay is a single block, the highly conservative choice of encoding rate increases transmission delay significantly, potentially to ∞ if $c < H_{\max}(X|Y)$.

The above observations motivate us to find a solution somewhere in between the two extreme approaches depicted in (2) and (3). Note that, if $c \gg \mathbb{E}[H_{(t)}(X|Y)]$ the problem becomes uninteresting (trivial if $c \geq H_{\max}(X|Y)$). The problem becomes interesting for $\mathbb{E}[H_{(t)}(X|Y)] < c < H_{\max}(X|Y)$. The major focus in performance analysis will be the case in which c is very close to, but slightly larger than $\mathbb{E}[H_{(t)}(X|Y)]$.

IV. PROPOSED APPROACHES

With the mere assumption that \mathcal{F} is a finite set, there does not exist a well-structured closed-form solution for optimization problem (1), for all possibilities of \mathcal{F} . However, we propose two structured class of strategies, Wait-to-Encode and Wait-to-Decode and show that they are both able to achieve optimal delay scaling as $\mathbb{E}[H_{(t)}(X|Y)]$ approaches channel rate c . Also in both sets of strategies, we provide ways to jointly encode and decode multiple (can be single if needed) blocks of source symbols together. In the rest of this section, our proposed strategies are presented in detail.

A. Wait-to-Encode Strategies

A strategy, π_{WE} , is called Wait-to-Encode if it accumulates blocks of symbols at the encoder and the blocks in the group are encoded simultaneously. With Wait-to-Encode, a source block is encoded only in a single message and an encoded message is no longer kept at the encoder. At the end of each block t , the encoder makes a decision about generating message M_t , which is based on whether a condition associated with strategy π_{WE} is satisfied or not. This condition is parameterized with $C(\pi_{WE})$. Any time a block, \mathbf{X}_t , is deferred for future encoding, the associated message, M_t , at that instant is blank. Node Y decodes M_t immediately after receiving it, thus $W_D^{(\pi_{WE})}(t) = 0$. Denoting the set of Wait-to-Encode strategies with Π_{WE} , we can summarize the general procedure as follows:

Algorithm 1 (Wait-to-Encode Strategy):

Observation:

At the beginning of block t , suppose $K - 1$ blocks of source symbols have been accumulated thus far, for $K = 1, 2, \dots$, waiting to be encoded. By the end of block t , node X and

Y observes \mathbf{x}_t and \mathbf{y}_t respectively and the marginals $F_{(t)}(x)$ and $F_{(t)}(y)$ are interchanged between nodes.

Encoding:

Node X generates message M_t as follows:

If $C(\pi_{WE})$ holds for the accumulated set of blocks
Blank message M_t is generated, i.e. $R_t = 0$;
Else

Message M_t is generated via SW encoding at rate
 $R_t = R_X(F_X^K, F_Y^K, \epsilon)$,

where F_X^K and F_Y^K represent the sequence of marginals for the observations of node X and node Y , respectively and the encoding rate function $R_X(F_X^K, F_Y^K, \epsilon)$ is chosen as:

$$R_X(F_X^K, F_Y^K, \epsilon) = \min_{R_X} R_X \quad (2)$$

$$s.t. \sum_{F^K \in \mathcal{F}^K} P_F(F^K | F_X^K, F_Y^K) \leq \epsilon,$$

where $\mathcal{F}^K \triangleq \left\{ F^K \in \mathcal{F}^K: \sum_{\tau=t-K+1}^t H_{(\tau)}(X|Y) \geq R_X \right\}$. SW encoding is a random coding strategy, in which $2^{\sum_{\tau=t-K+1}^t H_{(\tau)}(\mathbf{X})}$ possible typical node X observations are mapped into a binning structure with $2^{nR_X(F_X^K, F_Y^K, \epsilon)}$ bins. Then, given the vector of observations, node X finds the bin number of the associated vector in the binning structure and uses it as message M_t .

Theorem 1: Given \mathbf{x}^K , \mathbf{y}^K , F_X^K , F_Y^K and the outage probability constraint $\epsilon > 0$, the minimum achievable joint encoding rate for M_t is the solution of the constrained optimization problem (2).

Proof: The detailed proof can be found in Appendix A of our Technical Report [1]. To prove this theorem, we utilized random coding and typicality decoding ideas. ■

Possible choices for Condition $C(\pi_{WE})$: It is clear that the main differentiator between different Wait-to-Encode schemes is the choice of Condition $C(\pi_{WE})$. For instance, the two extreme cases for this class of algorithms are the ones in which $C(\pi_{WE})$ is chosen such that (i) $K = 1$ and (ii) $K = \infty$, which correspond to Illustrative Scenario (2) and (3), respectively. In general, Condition $C(\pi_{WE})$ dictates the achievable point in the tradeoff between $\mathbb{E}[W_E^{(\pi_{WE})}(t)]$ and $\mathbb{E}[W_C^{(\pi_{WE})}(t)]$. If we try to keep one of them small, then the other will increase. With this observation, we propose the following Wait-to-Encode strategy π_{WE}^* , in which the condition $C(\pi_{WE}^*)$ is chosen to be:

$$\frac{1}{K} R_X(F_X^K, F_Y^K, \epsilon) > c. \quad (3)$$

Though π_{WE}^* is not necessarily optimal with respect to Problem (1), we prove in Section V that it achieves optimal delay scaling as the entropy rate, $\mathbb{E}[H_{(t)}(X|Y)]$, of the observation at node X approaches to the channel rate c .

B. Wait-to-Decode Strategy

In a Wait-to-Decode strategy, π_{WD} , the encoder generates message M_t with encoding rate $R_t = c$, at the end of each time block t . For some blocks, this rate will be sufficient to decode \mathbf{X}_t at the desired probability of outage. For those blocks that

this is not true, block \mathbf{X}_t is not dropped at the encoder, and it is jointly encoded with the subsequent blocks of source observations. Once the encoder decides that sufficient information is accumulated at the decoder so that the group of blocks encoded can be decoded with the desired probability of outage, the blocks at the encoder are removed and the next block is encoded by itself, to start the process afresh. With Wait-to-Decode strategy, the main delay is experienced at the decoder, since the messages are accumulated there. Once the delay is calculated at the decoder, the other components of delay can be written as $W_E^{(\pi_{WD})}(t) = 1$ and $W_C^{(\pi_{WD})}(t) = \frac{n \cdot R_t}{n \cdot c} = 1$.

Motivated by the sequential binning strategy proposed in [2], we propose the following Wait-to-Decode strategy:

Algorithm 2 (Wait-to-Decode Strategy):

Observation:

At the beginning of block t , suppose $K - 1$ blocks of source symbols and messages have been respectively accumulated at node X and Y thus far, for $K = 1, 2, \dots$, waiting to be decoded. By the end of block t , node X and Y observes \mathbf{x}_t and \mathbf{y}_t respectively and the marginals $F_{(t)}(x)$ and $F_{(t)}(y)$ are interchanged between nodes.

Encoding:

Node X generates message M_t for jointly encoding \mathbf{x}^K via sequential SW encoding at rate $R_t = c$. Sequential SW encoding is also a random coding strategy, in which $2^{\sum_{\tau=t-K+1}^t H_{(\tau)}(\mathbf{X})}$ possible typical node X observations are mapped into a binning structure as with $2^{n \cdot c}$ bins. Then, given the vector of observations, node X finds the bin number of the associated vector in the binning structure and uses it as message M_t . After accumulating the bin number sequence, i.e. message sequence (M_{t-K+1}, \dots, M_t) , this sequence of bin numbers is sequentially connected and treated as the ‘bin number’ in traditional SW coding.

Decoding:

After receiving message M_t , node Y decodes the accumulated messages (M_{t-K+1}, \dots, M_t) as follows:

If $\frac{1}{K} R_X(F_X^K, F_Y^K, \epsilon) \leq c$ holds

Jointly decode all the accumulated messages (M_{t-K+1}, \dots, M_t) , remove \mathbf{x}^K from the encoder and start afresh;

Else

All the messages (M_{t-K+1}, \dots, M_t) keep being accumulated at node Y , waiting for the subsequent message(s).

Notice that, we described only one Wait-to-Decode strategy, without any control condition. The main reason is that, here, we use a constant encoding rate (as opposed to the *variable-rate encoding* in Wait-to-Encode class of strategies) and at the same time node Y decides to decode the accumulated messages as soon as the probability of outage goes below the desired threshold. We show in Section V that our Wait-to-Decode strategy also achieves optimal delay scaling as $\mathbb{E}[H_{(t)}(X|Y)]$ approaches channel rate c .

C. Comparison of Wait-to-Encode and Wait-to-Decode

Both proposed strategies have their own advantages and disadvantages. Under our Wait-to-Encode strategy π_{WE} , we carefully control the encoding rate, which is variable. We choose Condition $C(\pi_{WE}^*)$ in a way to minimize the encoding delay by transmitting the group of blocks, as soon as the encoding rate goes below the channel rate. The optimal design of $C(\pi_{WE})$ is complicated due to the strong coupling between the encoder and the channel, where the instant arrival rate of the channel, i.e. R_t , depends on our control decisions.

On the other hand, under the Wait-to-Decode strategy π_{WD} , we only have messages accumulating at the decoder. Due to the lack of control in our scheme, the end-to-end expected delay can be fairly large for certain blocks (despite being order optimal). For example, once a (very-long) block, demanding large SW encoding rate is observed, i.e., the required encoding rate is way above c , the decoder needs to wait for many subsequent blocks for things to smooth out in the long term. This is identical to the so called *slow truck effect* in First-in-First-out type queue scheduling.

V. PERFORMANCE BOUNDS

In this section, we derive upper and lower bounds for the end-to-end delay under both Wait-to-Encode strategy π_{WE}^* and Wait-to-Decode strategy π_{WD} . While our results are *general* for all possible values of the parameters, one of our main focus will be on the case in which the channel transmission rate c is larger than, but close to $\mathbb{E}[H_{(t)}(X|Y)]$. To formalize this, we define parameter η with $0 < \eta < 1$ such that $\mathbb{E}[H_{(t)}(X|Y)] = (1 - \eta)c$. The results reveal that both strategies achieve the order-optimal delay performance with respect to Problem (1) in heavy-traffic regime, i.e., $\eta \downarrow 0$. This regime is particularly interesting in cases when we want to fully utilize the available channel, while achieving a low finite end-to-end delay. At the end of the section, we illustrate the bounds via a simple numerical example.

Recall that the set of possible joint cdfs $\mathcal{F} = \{F_{11}, \dots, F_{l_1 l_1}; F_{21}, \dots, F_{2 l_2}; \dots; F_{m1}, \dots, F_{m l_m}\}$. Note that if $l_i = 1, \forall 1 \leq i \leq m$, then the marginal distribution information is equivalent to know the joint cdf information, for which the delay analysis becomes trivial. In fact, as we will discuss later, the delay scaling changes in the case of the availability of joint cdf, pointing out a *phase transition* phenomenon. The main focus in this section is in the case where there exists an m^* with $1 \leq m^* < m$ such that $l_i = 1$ for any $i \leq m^*$ and $l_i > 1$ for $m^* + 1 < i \leq m$. Let $\phi_{ij} \triangleq P_F(F_{ij})$, $\phi_i \triangleq \sum_{j=1}^{l_i} \phi_{ij}$, and $H_{\max_i}(X|Y) \triangleq \max_{1 \leq j \leq l_i} H_{ij}(X|Y)$. Next, we define some quantities that we use in expressing the performance. For all (i, j) such that $1 \leq i \leq m$, $1 \leq j \leq l_i$, let

$$\sigma_H^2 \triangleq \sum_{i=1}^m \sum_{j=1}^{l_i} \phi_{ij} (H_{ij}(X|Y) - (1 - \eta) \cdot c)^2,$$

$$c_i \triangleq \frac{1}{1 - \eta} \sum_{j=1}^{l_i} \frac{\phi_{ij}}{\phi_i} H_{ij}(X|Y),$$

$$\sigma_{H_i}^2 \triangleq \sum_{j=1}^{l_i} \frac{\phi_{ij}}{\phi_i} (H_{ij}(X|Y) - (1 - \eta)c_i)^2.$$

Also, let us succinctly denote the objective function of optimization problem (1) with $\mathbb{E}[\overline{W}^{(\pi)}(t)]$, i.e.,

$$\mathbb{E}[\overline{W}^{(\pi)}(t)] \triangleq \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[W_E^{(\pi)}(t) + W_C^{(\pi)}(t) + W_D^{(\pi)}(t)].$$

A. Delay Upper Bounds

In this section, we provide upper bounds on the end-to-end delay achieved by Wait-to-Encode strategy π_{WE}^* and Wait-to-Decode strategy π_{WD} to be valid for all values of η .

Theorem 2: For a given set of possible joint cdfs, \mathcal{F} , expected average end-to-end delay $\mathbb{E}[\overline{W}^{(\pi_{WE}^*)}(t)]$ and $\mathbb{E}[\overline{W}^{(\pi_{WD})}(t)]$ can be upper bounded as follows:

$$\mathbb{E}[\overline{W}^{(\pi_{WE}^*)}(t)] \leq \frac{3\gamma}{2} \cdot \frac{1}{\eta^2} + \frac{1}{2}, \quad (4)$$

$$\mathbb{E}[\overline{W}^{(\pi_{WD})}(t)] \leq \frac{\gamma}{2} \cdot \frac{1}{\eta^2} + \frac{3}{2}, \quad (5)$$

where $\gamma = \frac{-2 \ln \epsilon \cdot \sigma_H^2}{c^2} \cdot \left(1 + \frac{M_H \cdot c}{3\sigma_H^2} \cdot \eta\right)$ and $M_H = H_{\max}(X|Y) - \mathbb{E}[H_{(t)}(X|Y)]$.

Proof: Both strategies achieve their upper performance bounds in the worst case scenario when the encoder and the decoder do not use the marginal cdf of source symbols (or the marginal cdfs are useless, e.g. $m = 1, l_1 > 1$). In such a case, the joint encoding rate function $R_X(F_X^K, F_Y^K, \epsilon)$ no longer depends on F_X^K or F_Y^K , but the value of K . Hence, $R_X(F_X^K, F_Y^K, \epsilon)$ can be simply written as $R_X(K, \epsilon)$. Based on this observation, we define constant K_c as:

$$K_c \triangleq \min K \quad (6)$$

$$\text{s.t. } \mathbb{P} \left\{ \sum_{\tau=t}^{t+K-1} H_{(\tau)}(X|Y) > K \cdot c \right\} \leq \epsilon.$$

Since there always exists a solution to (6) for all $\eta \in (0, 1)$, it is clear that $R_X(K_c, \epsilon) \leq c$. Thus, in Wait-to-Encode strategy π_{WE}^* , the encoder will always accumulate K_c blocks to jointly encode; while in Wait-to-Decode strategy π_{WD} , the decoder will always accumulate K_c blocks to jointly decode (because of the lack of side information from the observation of the marginals). Since every K_c blocks form a cycle which will repeat over and over again in both strategies, the expected average end-to-end delay under the worst case scenario, i.e. $\mathbb{E}[\overline{W}_{WC}^{(\pi_{WE}^*)}(t)]$ and $\mathbb{E}[\overline{W}_{WC}^{(\pi_{WD})}(t)]$, can be easily derived as follows:

$$\mathbb{E}[\overline{W}_{WC}^{(\pi_{WE}^*)}(t)] = \frac{1}{K_c} \sum_{\tau=t}^{t+K_c-1} [W_E(\tau) + W_C(\tau) + W_D(\tau)]$$

$$= (3K_c + 1)/2 \quad (7)$$

since $W_E(\tau) = \tau - t + 1$, $W_C(\tau) = \frac{c \cdot K_c}{c}$, and $W_D(\tau) = 0$;

$$\mathbb{E}[\overline{W}_{WC}^{(\pi_{WD})}(t)] = \frac{1}{K_c} \sum_{\tau=t}^{t+K_c-1} [W_E(\tau) + W_C(\tau) + W_D(\tau)]$$

$$= \frac{K_c}{2} + \frac{3}{2} \quad (8)$$

since $W_E(\tau) = W_C(\tau) = 1$, and $W_D(\tau) = \tau - t$.

In general, K_c is difficult to be exactly evaluated. Yet, we can make use of Chernoff bound to approximately calculate K_c , denoted by \tilde{K}_c . From Theorem 2.11 in [3], we have

$$\mathbb{P} \left\{ \sum_{\tau=t}^{t+K-1} H_{(\tau)}(X|Y) > K \cdot c \right\} \leq \exp \left\{ -\frac{K\eta^2 c^2}{2(\sigma_H^2 + M_H \cdot c \cdot \eta/3)} \right\}.$$

Therefore, if we define \tilde{K}_c as:

$$\tilde{K}_c = \min K \quad \text{s.t.} \quad \exp \left\{ -\frac{K\eta^2 c^2}{2(\sigma_H^2 + M_H \cdot c \cdot \eta/3)} \right\} \leq \epsilon,$$

then we have

$$\tilde{K}_c = \frac{-2 \ln \epsilon \cdot \sigma_H^2}{c^2 \cdot \eta^2} \cdot \left(1 + \frac{M_H \cdot c}{3\sigma_H^2} \cdot \eta \right). \quad (9)$$

By substituting (9) into (7) and (8), and replacing K_c with \tilde{K}_c , we complete the proof. ■

Theorem 2 indicates that without any marginal distribution information, the expected end-to-end delay under both WE and WD strategies scales as $O(1/\eta^2)$. Next, we derive lower bounds for the same delay and show that the lower bounds also have the same scaling law. This indicates that our algorithms are order optimal.

B. Delay Lower Bounds

Next, we evaluate lower bounds on the end-to-end expected delay achieved for all $\pi_{WE} \in \Pi_{WE}$ and π_{WD} .

Theorem 3: Given the set of possible joint cdfs \mathcal{F} , expected delays $\mathbb{E}[\bar{W}^{(\pi_{WE}^*)}(t)]$ and $\mathbb{E}[\bar{W}^{(\pi_{WD})}(t)]$ can be lower bounded as:

$$\mathbb{E}[\bar{W}^{(\pi_{WE})}(t)] \geq \max_{m^*+1 \leq i^* \leq m} \left\{ (1-\eta) \sum_{i \neq i^*} \frac{\phi_i c_i}{c} + \phi_{i^*} \left(\frac{4\gamma_{i^*} c_{i^*}}{27 \cdot c(1+\beta_{i^*})^2} \cdot \frac{1}{\eta^2} + \frac{1}{6} \right) \right\} \quad (10)$$

$$\mathbb{E}[\bar{W}^{(\pi_{WD})}(t)] \geq \max_{m^*+1 \leq i^* \leq m} \left\{ (1-\eta) \sum_{i \neq i^*} \frac{\phi_i c_i}{c} + \phi_{i^*} \left(\frac{2\gamma_{i^*}}{27(1+\beta_{i^*})^2} \cdot \frac{1}{\eta^2} + \frac{1}{2} \right) \right\}, \quad (11)$$

where $\gamma_{i^*} = \frac{-2 \ln \epsilon \cdot \sigma_{H_{i^*}}^2}{c_{i^*}^2} \cdot \left(1 + \frac{M_{H_{i^*}} \cdot c_{i^*}}{3\sigma_{H_{i^*}}^2} \cdot \eta \right)$, $M_{H_{i^*}} =$

$H_{\max_{i^*}}(X|Y) - (1-\eta) \cdot c_{i^*}$, and $\beta_{i^*} = \frac{\sum_{i \neq i^*} \phi_i c_i}{\phi_{i^*} c_{i^*}}$.

Proof: The complete proof can be found in *Appendix B* of [1]. Here, we just provide a brief sketch: We bound the performance of both strategies from below by considering a genie-aided scenario, in which part of the sequence of joint cdfs is provided to both nodes by a genie. In particular, all joint cdfs are available to node X and Y , except for the blocks with joint cdf F_{i^*j} ($m^*+1 \leq i^* \leq m$). Then we

applied the following steps for the lower bound: (i) Encode the blocks with cdf F_{ij} ($i \neq i^*$) block by block and in calculating the end-to-end delay, ignore all, except for the transmission delay; (ii) Assume the channel is always idle whenever a message arrives; (iii) Accumulate blocks with cdf F_{i^*j} to jointly encode, which leads to the same situation as discussed in the proof of *Theorem (2)* for the blocks with that particular cdf. Then, we can use a similar line of techniques to derive the lower bound. Finally, by taking the maximum over all possible lower bounds, the tightest lower bound is evaluated. ■

Discussion: Theorem 3 reveals that the end-to-end delay scales as $O(1/\eta^2)$ under both WE and WD strategies. Along with Theorem 2, we showed that both WE and WD strategies achieve order-optimal delay performance in the heavy-traffic regime (see [7], [10]), i.e., $\eta \downarrow 0$. Also, one interesting observation is that the marginal distribution information does not necessarily improve the delay scaling. This is in contrast to the case where the perfect knowledge of the joint cdf is available. If the joint cdf is known, the experienced delay is equivalent to the queuing delay with random arrivals $H_{(t)}(X|Y)$ and constant service rate c , which scales with $O(1/\eta)$ (see [7, Lemma 4]). This points out a **phase transition phenomenon**: Even a minor degradation in the knowledge of joint statistics at the encoder -from perfect knowledge to a slightly imperfect knowledge- leads to a different scaling regime in the expected delay.

C. Numerical Evaluation

To illustrate how the performance bounds vary with η , we study a simple example as follows: (i) $\mathcal{F} = \{F_{11}, F_{21}, F_{22}, F_{31}, F_{32}, F_{33}\}$, $\epsilon = 0.01$; (ii) $\phi_{11} = 0.1$, $\phi_{21} = \phi_{22} = 0.2$, $\phi_{31} = 0.12$, $\phi_{32} = \phi_{33} = 0.19$; (iii) $H_{ij}(X|Y) = i + j$. Thus the source entropy varies between 2–6 bits/symbol. In Fig 3, we plot the upper and lower bounds on the expected delay for the observations taken at node X to be replicated at node Y , measured in number of blocks (n slots), as a function of the heavy-traffic parameter η^{-1} .

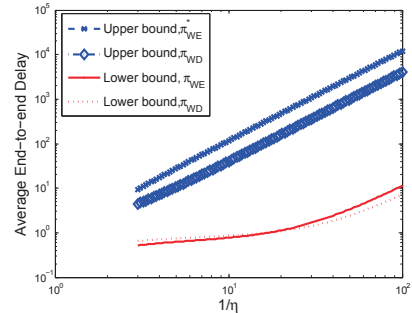


Fig. 3: Delay bounds for Wait-to-Encode and Wait-to-Decode schemes: both x-axis and y-axis are in logarithmic scale

Examining the plots, one can see that both upper and lower bounds scale as $O(\frac{1}{\eta^2})$ as $\eta \rightarrow 0$, which reflect the order optimality of our proposed strategies. Also, for the above set of parameters, the bounds slightly favor Wait-to-Decode strategy, but note that, this is not a typical trend and one can come up with another set of parameters for which the opposite holds.

VI. EXPERIMENTAL EVALUATION

To illustrate the performance of average end-to-end delay and compare the proposed encoding and decoding strategies, we set up an experiment which emulates a possible application of our problem in real-time streaming networks. As shown in Fig. 1 and discussed in Section I, we set two cameras at different locations at still to record people walking by in a busy street. The distance between the cameras is 20 meters and the directions they face are at a 90° angle to one another.

After synchronization, we end up with two correlated uncompressed video frame sequences. For the video signals, each frame has a size 1024×1024 , where each pixel is represented with 3 bytes (associated with RGB-index), hence each frame is a 3 MB sample. The frame rate is 60 frame/sec, i.e. the streaming rate of the uncompressed video is 180 Mbps. We regard each frame as one video symbol (a very large one) and pile up every $n = 180$ symbols to form blocks $\{\mathbf{X}_t\}$, $\{\mathbf{Y}_t\}$ in our model, i.e., a time slot is 1/60 secs long and a block is 3 secs long. The empirically generated time-varying joint (marginal) cdf sequences are represented by $\{F_{(t)}(x, y)\}$, $\{F_{(t)}(x)\}$, and $\{F_{(t)}(y)\}$.

To form the set \mathcal{F} , we use a pilot shot over a certain duration. Due to the huge size of the symbols and the unpredictable environment, developing the exact representation of set \mathcal{F} is not possible. Thus, we quantize the set of all possible distributions: Firstly, using the pilot sequence, we calculate the relative entropy of each block t with respect to block 1, i.e. $D(F_{(t)}(x, y) || F_{(1)}(x, y))$, which gives us the sequence of relative entropies $\{D_{(t)}(x, y)\}$. Next, we quantize the values of $\{D_{(t)}(x, y)\}$ into 128 intervals and treat each quantization level as a single joint distribution, one which is randomly picked from each interval. Hence, we end up with 128 different quantized joint cdfs $\{F_{ij}(x, y)\}$ in order to form the set \mathcal{F} . Finally, we repeat an identical quantization process for the marginal cdf sequences $\{F_{ij}(x)\}$ and $\{F_{ij}(y)\}$, but we set the number of quantization intervals to 8 this time. Thus, we have 64 different combinations of marginal cdf pairs. Recall that $\mathcal{F} = \{F_{11}, \dots, F_{1l_1}; F_{21}, \dots, F_{2l_2}; \dots; F_{m1}, \dots, F_{ml_m}\}$, hence $m = 64$ after the above quantization process. The empirical PMF $P_F(F_{ij})$ can be calculated with respect to each $F_{ij} \in \mathcal{F}$, using the pilot shot. Hence, we use an **extremely coarse quantizer** in the representation of the sources. One of our main objectives is to show that, even with such coarse quantization, it is possible to achieve a substantial compression ratio (e.g., reduction in the rate of data transmission from node X to node Y) for \mathbf{X}_t to be replicated at node Y . After the pilot shot, we first quantize the observed new frames as described above. After quantization, the conditional entropy rate we observed from the combined pilot shots turned out to be 76.5 Mbps, which we take as the basic limit for the minimum rate node X needs to transmit for lossless recovery (0-outage) at node Y . Note that, once we obtain the video traces based on real-world data and construct set \mathcal{F} , we use simulations to evaluate the performance of the system. We ran these real-world data-driven simulations for 10^6 blocks of video symbols.

In our first evaluation, we focus on the heavy-traffic sce-

nario. Transmission rate is fixed at $c = \frac{1}{1-\eta} \times 76.5$ Mbps, for a given heavy-traffic parameter η . The end-to-end delay experienced by each block was stored and then used to calculate the average delay over all 10^6 blocks. Average delay is plotted vs. η^{-1} in Fig. 4a for Wait-to-Encode strategy π_{WE}^* , Wait-to-Decode strategy π_{WD} , along with their associated upper/lower bounds we derived in Section V. In the plot, ‘WE’ and ‘WD’ stand for ‘Wait-to-Encode’ and ‘Wait-to-Decode’.

As can be observed in Fig. 4a, the expected delay of all proposed strategies scale as $O(\frac{1}{\eta^2})$ as $\eta \rightarrow 0$. Note that, even though π_{WD} achieves better performance than π_{WE}^* in our experiment, this is not necessarily a common trend for all possible Wait-to-Encode strategies. In fact, in [1], we provide a new adaptive-threshold based scheme, which achieves a better expected delay than π_{WD} when η is relatively large. Also, it is worth mentioning that the bounds become tighter, as the traffic load gets lower.

Next, we plot the encoding rate chosen by our Wait-to-Encode strategy, π_{WE}^* , as a function of time, for various values of η and outage probability constraint ϵ . Note that, in Wait-to-Decode strategy, π_{WD} , the encoding rate is always identical to c , which we choose to be $\frac{1}{1-\eta} \times 76.5$ Mbps in these plots. In Fig. 4b, we illustrate the encoding rate, measured regularly across blocks. In particular, we take one sample per 10^4 blocks through the entire trace. The channel rate (thus the encoding rate for π_{WD}) is $c = 102$ and 77.3 Mbps for $\eta = 0.75$ and 0.99 , respectively. One can see in Fig 4b, if the traffic is light (i.e., the average conditional entropy of the source is much smaller than the channel rate), π_{WE}^* chooses the encoding rate high, since the fixed channel transmission rate is sufficiently large to choose encoding rate more aggressively to reduce delay. Also, as expected, a smaller value of ϵ requires higher encoding rate to keep the outage constraint to be met. Finally, note that the long-term average encoding rate with π_{WE}^* will always be smaller than that with π_{WD} , but the ratio of the average encoding rates will be no less than $1 - \eta$.

Another interesting observation from this plot is the following. Recall that the rate of the uncoded video is 180 Mbps; thus, with the available channel rates, it is not possible for the uncoded video to be transmitted from node X to node Y . However, even with the highly coarse quantization that we use for the source observations, we reduced the average encoding rate (i.e., the rate at which node X transmits to node Y) to between 75-85 Mbps at an outage probability of 0.05, which corresponds to a compression ratio of approximately 50% at a reasonably low block delay, which will be analyzed next.

Finally, we focus on the delay for two different values of ϵ at $\eta = 0.25$, which corresponds to a 75% utilization. We plot the sample path for the end-to-end block delay in Fig. 4c for strategies π_{WE}^* and π_{WD} . In these experiments, $c = 102$ Mbps. With only a delay of 6-9 secs, we achieve an outage probability of 0.05. This delay achieved, combined with the 50% compression ratio demonstrates the efficacy of our system. We believe much higher compression ratios could be possible with finer quantizers and higher correlations between sources. Finally, note that the end-to-end delay becomes

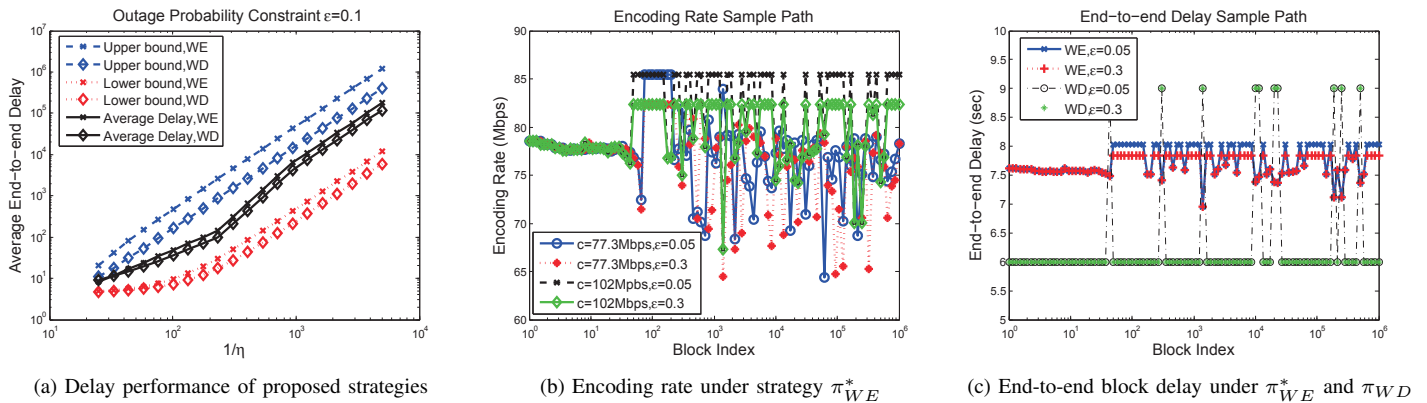


Fig. 4: Experimental Results. With a delay of a few blocks, the compression ratio of around 50% is achievable for lossless replication subject to a probability of outage of 0.05.

constant as the transmission rate increases (low utilization), leading to illustrative scenario (3), depicted in Section III. In that case, block-by-block encoding becomes the best scheme. **Discussion:** There is another alternative for our system design that we did not consider here, as it pertains to video compression. The observed video signal at node X can be compressed via standard video compression techniques (non-distributed), independently of the observation of node Y . The basic limit for the rate after that process is $\mathbb{E}[H_{(t)}(X)]$, while one can achieve a much smaller rate $\mathbb{E}[H_{(t)}(X|Y)]$ (i.e., a much higher compression ratio) with the distributed approach. One important question that we are planning to answer as a part of future work is the comparison of our schemes with standard video compression. We will study systematic ways to develop quantizers (possibly more complex than we have here) that are simple, yet provide significant gains over the best available non-distributed video compression techniques.

VII. CONCLUSIONS

We studied the lossless distributed source coding problem in which there exists a pair of nodes, observing a random source with time-varying statistics, unknown to the nodes before the session starts. We formulated the problem as that of minimization of end-to-end delay for the observations of one of the nodes to be replicated at the other node, subject to a certain desired outage probability. Even though, it is not possible to come up with well-structured solutions to the basic problem, due to the generality of the set of distributions, we developed two different classes of strategies, Wait-to-Encode and Wait-to-Decode that are *provably* order optimal in the heavy traffic limit. After analytically deriving general bounds for the expected delay achieved by our schemes, we further experimentally demonstrate the efficacy of the schemes, using a setup involving two cameras, obtaining videos of a common scene at different angles. We showed that, even with a very low-complexity quantizer, a compression ratio of approximately 50% is achievable for lossless replication at the decoder, at an average delay of 5-10 seconds.

ACKNOWLEDGMENT

This work is supported in part by NSF grant CNS-1054738.

REFERENCES

- [1] Technical report for low-delay distributed source coding for time-varying sources with unknown statistics. <https://www.dropbox.com/s/lhzy4ls1iy12o7v/TechnicalReportInfocom2015.pdf>.
- [2] C. Chang. Streaming source coding with delay. In *Thesis, EECS Department University of California, Berkeley*, 2007.
- [3] F. Chung and L. Lu. Complex graphs and networks. In *American Mathematical Society*, 2006.
- [4] I. Csiszar and J. Korner. Towards a general theory of source networks. *IEEE Trans. on Inf. Th.*, 26(2):155–165, Mar 1980.
- [5] S. Draper. Universal incremental slepian-wolf coding. In *Proc. 2nd Int. Conf. Quality Service Heterogeneous Wired/Wireless Networks*, 2004.
- [6] E. Dupraz, A. Roumy, and M. Kieffer. Source coding with side information at the decoder: Models with uncertainty, performance bounds, and practical coding schemes. 40:586–595, 2012.
- [7] A. Eryilmaz and R. Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, 72:311–359, 2012.
- [8] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, Jan 2005.
- [9] S. Jaggi and M. Effros. Universal linked multiple access source codes. In *IEEE International Symposium on Information Theory*, 2002.
- [10] B. Li, R. Li, and A. Eryilmaz. Heavy-traffic-optimal scheduling with regular service guarantees in wireless networks. In *ACM MOBIHOC*, Bangalore, India, July 2013.
- [11] A. Liveris, Z. Xiong, and C. Georghiadis. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Commun. Lett.*, 6:440–442, 2002.
- [12] R. Matsumoto. Problems in application of ldpc codes to information in quantum key distribution protocols. In *IEICE Technical Report*.
- [13] T. Matsuta, T. Uyematsu, and R. Matsumoto. Universal slepian-wolf source codes using low-density parity-check matrices. In *Proc. IEEE International Symposium on Information Theory*, pages 186–190, 2010.
- [14] R. Parseh and F. Lahouti. Rate adaptation for slepian-wolf coding in presence of uncertain side information. In *The Fifth International Conference on Sensor Technologies and Applications*, 2011.
- [15] R. Puri and K. Ramchandran. Prism: a new robust video coding architecture based on distributed compression principles. In *Proc. Allerton Conference on Communications Control and Computing*, pages 665–674, 2002.
- [16] S. Shriram, D. Baron, and R. Baraniuk. Variable-rate universal slepian-wolf coding with feedback. In *Asilomar Conference*, 2005.
- [17] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. on Inf. Th.*, 19(4):471–480, Jul 1973.
- [18] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiadis. Design of slepian-wolf codes by channel code partitioning. In *Proc. Data Compression Conference*, pages 302–311, 2004.
- [19] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiadis. On code design for the slepian-wolf problem and lossless multiterminal networks. *IEEE Trans. Inf. Theory*, 52(4):1495–1507, 2006.
- [20] Z. Xiong, A. Liveris, and S. Cheng. Distributed source coding for sensor networks. *Signal Processing Magazine, IEEE*, 21(5):80–94, Oct 2004.