Network Vector Quantization

Michael Fleming, *Student Member, IEEE*, Qian Zhao, *Member, IEEE*, and Michelle Effros, *Senior Member, IEEE*

***Abstract*—We present an algorithm for designing locally optimal vector quantizers for general networks. We discuss the algorithm's implementation and compare the performance of the resulting "network vector quantizers" to traditional vector quantizers (VQs) and to rate-distortion (R-D) bounds where available. While some special cases of network codes (e.g., multiresolution (MR) and multiple description (MD) codes) have been studied in the literature, we here present a unifying approach that both includes these existing solutions as special cases and provides solutions to previously unsolved examples.**

***Index Terms*—Broadcast, generalized Lloyd algorithm, multiple access, multiterminal, side information, source coding.**

I. Introduction

A. Network Source Coding

THE growth in the amount of data being transferred through networks motivates us to improve network communication techniques. Efficient and reliable communication requires both source and channel coding. While network channel coding (including channel coding for multiple access, broadcast, and relay channels as special cases) has received a fair amount of attention (e.g., [1] and the references therein), network source coding has been largely overlooked. This work treats network source code design.

We define a network to be any collection of nodes joined by communication links. In the most general case, every node can communicate with every other node, and any particular message transmitted by a node may be intended for one or more of the other nodes. Every node has a collection of sources to be encoded for transmission and a collection of sources for which it receives descriptions and builds reproductions.

At first glance, it is unclear why the way we compress a source should depend on the network topology when an "independent" coding approach works for all networks. In the independent coding paradigm, each node compresses each of its outgoing sources and decompresses each of its incoming sources independently, using only traditional (point-to-point) data compression techniques. The way we encode a source does not depend on either the number of intended recipients of our data or the number of other nodes sending information to the same receivers. Many current networks rely exclusively on independent coding.

In this work, we demonstrate the benefits of the alternative "network" approach, in which we incorporate network topology into compression system design. In particular, we describe how to optimize a source code for a given network topology and demonstrate the resulting improvements in both rate-distortion performance and system functionality. Network codes yield rate-distortion benefits by exploiting the correlation between network messages; independent coding removes the redundancy in each individual source, but fails to remove the redundancy *between* sources. Consider a sensor network in which spatially separated nodes make measurements in a shared environment. We expect correlation between the measurements made by different nodes, and we would like our source code to use the correlation to reduce either the network message rates or the reproduction distortions. To that end, each node should jointly encode all outgoing messages and jointly decode all incoming messages. Also, any side information available to a node (for example, measurements taken at that node) should be used in its decoding process. The design techniques in this paper demonstrate how to optimize the joint encoders and decoders and how to incorporate side information into code design.

Benefits in functionality arise in network codes that incorporate features useful in the given network topology. For example, consider a website that is accessed by users with very different connection speeds. To cater to all users effectively using an independent approach, the website must store several source descriptions, each optimized for a different speed. In contrast, a network source code, in this case a multiresolution (MR) source code [2], [3], can meet the needs of all users with a single (embedded) source description.

A network coding approach can be used on all systems that can be cast into a network model. We summarize common examples below and illustrate them in Fig. 1. The point-to-point system is provided as an illustration reference in Fig. 1(a).

- **Broadcast (BC) Codes [4], [5]:** In a BC code, shown in Fig. 1(b), a single sender describes a collection of sources to several receivers. A different message can be transmitted to each possible subset of the receivers.
- **Multiple Access (MA) Codes [6]:** In an MA code, shown in Fig. 1(c), many senders transmit information to a single receiver.
- **Wyner–Ziv (WZ) Codes [7], [8]:** WZ codes, shown in Fig. 1(d), apply to point-to-point communication systems in which side information is available at the decoder.

Manuscript received January 22, 2002; revised July 31, 2003. This material is based upon work supported by the F.W.W. Rhodes Memorial Scholarship, a Redshaw Award, the National Science Foundation under Grant CCR-9909026, the Lee Center for Advanced Networking at Caltech, and the Intel Technology for Education 2000 program. The material in this paper was presented in part at the IEEE Data Compression Conference, Snowbird, UT, 1999, 2000, 2001; the 2000 Conference on Information Sciences and Systems, Princeton, NJ, 2000; and the 33rd Asilomar Conference on Signals, Systems, and Computers, Asilomar, CA, 2000.

M. Fleming and M. Effros are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA.

Q. Zhao is with the Cluster and Parallel Storage Technology Department, Oracle Corporation, Redwood City, CA 94065 USA.

Communicated by R. Zamir, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2004.831832

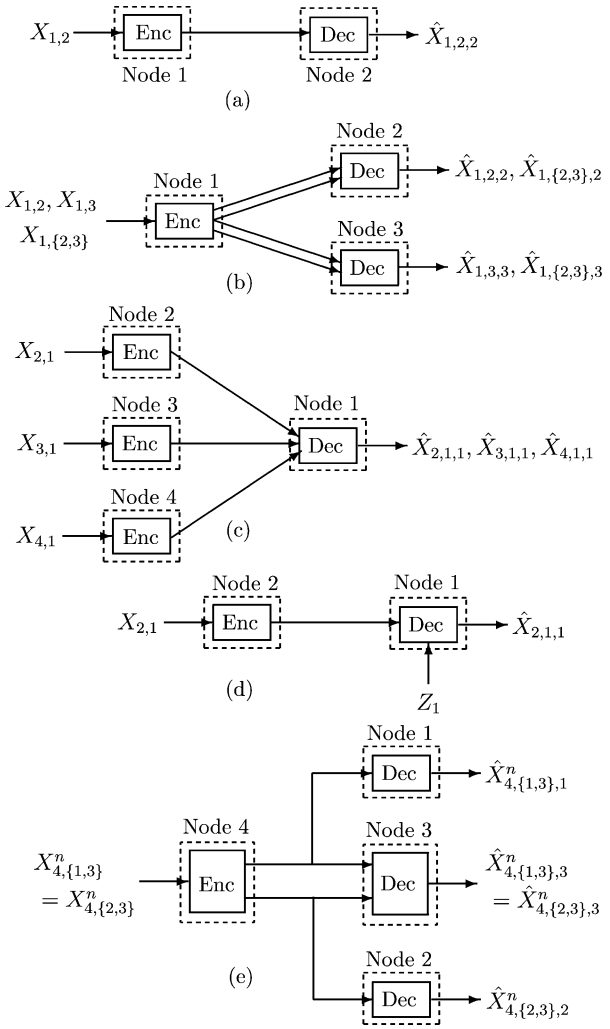


Fig. 1. (a) A point-to point code. (b) A two-receiver BC code. (c) A three-user MA code. (d) A WZ code with side information Z_1. (e) A two-channel MD code. The notation, explained in detail in Section II, is $X_{t,S}$ for a source and $\hat{X}_{t,S,r}$ for a reproduction; here t is the transmitter, S the set of receivers, and r the reproducer.

- **Multiple Description (MD) Codes [9]–[11]:** An MD code can be used for point-to-point communication over multiple, unreliable communication channels (or over a lossy, packet-based channel in which lost packets cannot be retransmitted). Each channel's source description may be lost, and the decoder reproduces the source by combining all received descriptions. In Fig. 1(e), we model a two-channel system and represent the different decoding scenarios with three separate decoders.

Fig. 2 shows the relationship between point-to-point, MR, MD, BC, WZ, and MA codes.

Interesting applications for network coding also arise in systems for which a network model is not obvious but can be applied to advantage, as in the following two examples.

- **Independent-frame video coding:** A weather satellite sends one image of Earth every hour to a base station. The satellite has limited memory and cannot store the images. If it could, we could exploit the high correlation between images using a standard video code. Without storage, each image must be encoded separately. We exploit the correlation using a WZ code; the base station decodes each transmission using the previous $k \geq 1$ received images as side information. If the satellite occasionally goes "off-line," so that we cannot count on the availability at the decoder of all k previous images, then we build several decoders, each optimized for a different available subset of side information. This is now a combination of WZ and MD coding.
- **Distributed data storage:** Suppose many devices together store a large amount of data for a server, and system users typically combine information from several devices with each system use. If some of the devices are occasionally unavailable, then we can use a network source code (e.g., an MD code) to set the balance between the amount of redundancy added to the stored data and the fidelity of the data reproductions.

The rate-distortion and functionality benefits illustrated by these examples come at the cost of increased design complexity. Correlation between sources creates dependencies in the design of joint encoders and decoders, and we must optimize all components of the network simultaneously. This contrasts with the independent coding approach, where optimization over the entire network is suboptimally broken up into many smaller optimizations, each with low complexity. Once design is complete, however, the run-time complexity of the two approaches is similar. Joint decoding, like independent decoding, can be done using table lookup, albeit with a larger table. Joint encoding is more complex than independent encoding, but if encoding complexity is critical, then it can be greatly reduced by approximating the optimal encoder with a hierarchical structure of tables following the approach of [12].

Scalability of network code design depends on the interconnectivity of the network. For an M-node network in which the in-degree of each node is constant as M grows, design complexity increases linearly in M, and code design for large networks is feasible. If, however, the in-degree of each node increases with M, then the design complexity increases exponentially in M and our approach is useful for small networks only.

B. Network Vector Quantizers (NVQs)

In this work we focus on the design of vector quantizers (VQs) for networks (NVQs). The choice of VQs (which include scalar quantizers (SQs) as a special case) is motivated by their practicality, generality, and close relationship to theory. We extend the generalized Lloyd algorithm to develop an iterative approach for the design of NVQs for any network topology. Our primary result is an algorithm that guarantees local, but not necessarily global, rate-distortion optimality in the resulting NVQs for some systems. For other systems, approximations required for practical implementation remove this guarantee, although we do observe convergence in all of our experimental work.

We build on previous fixed-rate and entropy-constrained SQ and VQ design algorithms. An algorithm to design locally optimal fixed-rate VQs for the simplest network, the point-to-point network, appears in [13], and its extension to variable-rate coding via the inclusion of an entropy constraint appears in [14]. The approaches of [13] and [14] have been generalized for application to MR and MD networks. Examples

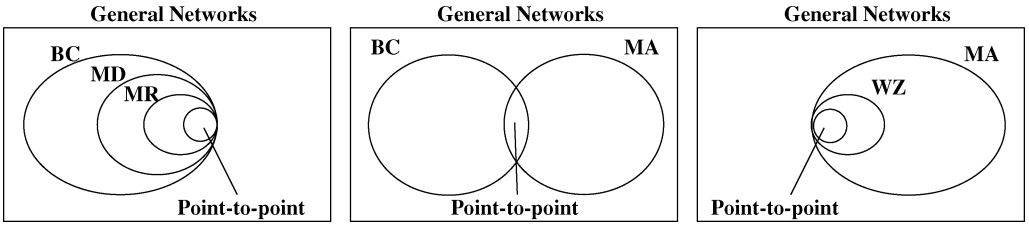


Fig. 2. The relationship between the systems. WZ systems are considered as special cases of MA systems for which one source (the side information) has no rate constraint.

of MR coding include tree-structured VQ [15], [16], locally optimal fixed-rate multiresolution SQ (MRSQ) [17], variable-rate MRSQ [18], and locally optimal variable-rate MRSQ and multiresolution VQ (MRVQ) [19], [20]. Examples of MD coding include locally optimal two-description fixed-rate multiple-description VQ (MDVQ) [21] and locally optimal two-description fixed-rate [22] and entropy-constrained [23] MDSQ. Since the submission of this paper, algorithms have been developed that achieve globally optimal VQ design for many of the systems studied here [24]; those algorithms rely directly on the optimal encoder and decoder definitions discussed in this work.

The work in this paper, introduced in part in [5], [25]–[28], extends locally optimal VQ coding to general networks. This extension requires us to solve the problem of locally optimal VQ design for BC, MA, and WZ systems, since each node in a general network can transmit and receive multiple messages and use side information. Although two VQ design schemes for MA coding appear in [29], they are not designed explicitly for rate-distortion optimality. In this work, we explicitly optimize for rate-distortion performance. Our work both unifies existing results into a single framework and provides new results for previously unsolved NVQ design scenarios. Examples of new results include fixed-rate quantizer design for arbitrary networks and variable-rate quantizer design for M-description ($M \geq 2$) MD, two-user MA, and two-receiver BC systems.

Our initialization method borrows from a network coding scheme outside of VQ. Algebraic binning approaches (e.g., [30], [31]) for WZ and MA systems yield codes with a structure that is well suited to the coding of symmetric or Gaussian sources under quadratic distortion. When appropriate, we adopt that structure in NVQ initialization. The resulting NVQs have no obvious advantages over binning codes for the specific applications mentioned earlier, but the unconstrained nature of our NVQ design algorithm allows them to better fit more general source distributions and distortion measures.

Following the entropy-constrained coding tradition (see, for example, [14], [20], [32], [33]), we describe lossy code design as quantization followed by entropy coding. The only loss of generality associated with the entropy-constrained approach is the restriction to solutions lying on the lower convex hull of achievable entropies and distortions. We here focus exclusively on the quantizer design,[1] considering entropy codes only insofar as their index description rates affect quantizer optimization.

[1]The topic of entropy code design for network systems is a rich field deserving separate attention; see, for example, [4], [26], [34]–[40].

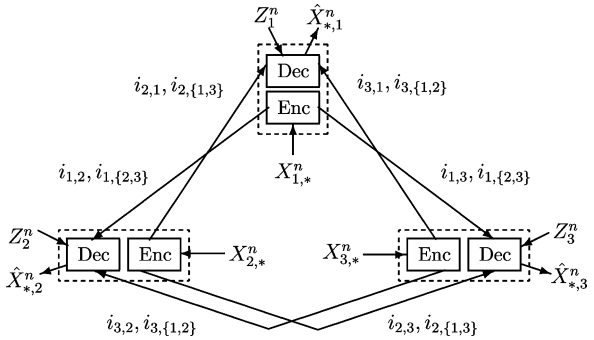


Fig. 3. A general three-node network.

While the entropy codes of [14], [20] are lossless codes, entropy coding for many networks requires the use of codes that are lossless in the regular Shannon sense; that is, they require entropy codes with asymptotically negligible (but nonzero) error probabilities [34], [40], [41]. We here call those codes *near-lossless* codes and assume their use where necessary. As in [14], [20], [32], [33], we approximate entropy code performance using the asymptotically optimal values—reporting rates as entropies and assuming zero error probability. This approach is consistent with past work. It is also convenient since tight bounds on the nonasymptotic performance are not currently available and the current high level of interest in entropy coding for networks promises rapid changes in this important area. The extension of our approach to include entropy code optimization and account for true (possibly nonzero) error probabilities in the iterative design is straightforward, and we give the optimization equations in their most general form to allow for this extension.

Although our algorithm allows fixed- and variable-rate quantizer design for arbitrary networks, potential optimality in variable-rate design is currently limited to a select group of systems. It requires the availability of optimal theoretical entropy constraints and/or optimal practical entropy codes. These are available for MR, MD, WZ, two-user MA, and two-receiver BC systems. For multiuser MA systems, the optimal theoretical entropy constraints are not yet known, nor have optimal practical near-lossless codes been created. For multireceiver BC and general networks (e.g., the general three-node network of Fig. 3), even the asymptotically optimal near-lossless rates are unknown. For such networks, we must design variable-rate quantizers using rates that are known to be achievable in place of the unknown optimal rates. The resulting quantizers are necessarily suboptimal.

The paper is organized as follows. We develop a framework for network description in Section II. The optimal design equations for an NVQ are presented in Section III, and we discuss their implementation in Section IV. Section V presents experimental results for specific network design examples, and we draw conclusions in Section VI.

II. Network Description

In this section, we develop a framework for describing network components and define optimality for network source codes. Due to the complexity of a general network, this discussion requires a significant amount of notation; we simplify where possible.

Consider a dimension-n code for an M-node network. In the most general case, every node communicates with every other node, and a message may be intended for any subset of nodes. Let $X^n_{t,S} \in \mathcal{X}^n_{t,S}$ denote the source to be described by node t to the nodes in set $S \subseteq \mathcal{M} = \{1, \ldots, M\}$. For example, $X^n_{1,\{2,3\}}$ is the source described by node 1 to nodes 2 and 3. If S contains only one index, we write $X^n_{t,\{r\}} = X^n_{t,r}$. Let $\hat{X}^n_{t,S,r} \in \hat{\mathcal{X}}^n_{t,S,r}$ denote the reproduction of source $X^n_{t,S}$ at node $r \in S$. Thus, $\hat{X}^n_{1,\{2,3\},2}$ is node 2's reproduction of source $X^n_{1,\{2,3\}}$. Reproductions $\hat{X}^n_{1,\{2,3\},2}$ and $\hat{X}^n_{1,\{2,3\},3}$ can differ since nodes 2 and 3 jointly decode the description of $X^n_{1,\{2,3\}}$ with different source descriptions. The source and reproduction alphabets can be continuous or discrete, and typically, for each (t,S), $\hat{\mathcal{X}}^n_{t,S,r} = \mathcal{X}^n_{t,S}, \forall r \in S$.

For each node $t \in \mathcal{M}$, let $\mathcal{S}(t)$ denote a collection of sets such that for each $S \in \mathcal{S}(t)$, there exists a source to be described by node t to precisely the members of $S \subseteq \mathcal{M}$. Then $X^n_{t,*} = (X^n_{t,S})_{S \in \mathcal{S}(t)}$ gives the collection of sources described by node t. Similarly, for each $r \in \mathcal{M}$, let

$$\mathcal{T}(r) = \{(t,S) \in \mathcal{S} : r \in S\}$$

be the set of source descriptions received by node r, where

$$\mathcal{S} = \{(t,S) : t \in \mathcal{M}, S \in \mathcal{S}(t)\}$$

is the set of sources in the network. Then

$$\hat{X}^n_{*,r} = (\hat{X}^n_{t,S,r})_{(t,S) \in \mathcal{T}(r)}$$

gives the collection of reconstructions at node r. Finally, let

$$\mathcal{T} = \{(t,S,r) : r \in \mathcal{M}, (t,S) \in \mathcal{T}(r)\}$$

denote the set of all transmitter-message-receiver triples.

For each node $r \in \mathcal{M}$, we denote the side information available at node r by $Z^n_r \in \mathcal{Z}^n_r$. Alphabet $\mathcal{Z}_r$ can be continuous or discrete.

Fig. 3 shows the example of a general three-node network. Each node transmits a total of three different source descriptions. Node 1, for instance, encodes a source intended for node 2 only, a source intended for node 3 only, and a source intended for both. These are denoted by $X^n_{1,2}$, $X^n_{1,3}$, and $X^n_{1,\{2,3\}}$, respectively, giving

$$X^n_{1,*} = (X^n_{1,2}, X^n_{1,3}, X^n_{1,\{2,3\}}).$$

Each node in the network receives and decodes four source descriptions. The collection of reproductions at node 1 is

$$\hat{X}^n_{*,1} = (\hat{X}^n_{2,1,1}, \hat{X}^n_{2,\{1,3\},1}, \hat{X}^n_{3,1,1}, \hat{X}^n_{3,\{1,2\},1});$$

their descriptions are jointly decoded with the help of side information Z^n_1. The total number of reproductions is greater than the number of sources since some sources are reproduced at more than one node.

A network encoder comprises two parts: a quantizer encoder, followed by an entropy encoder. A network decoder comprises two complementary parts: an entropy decoder followed by a quantizer decoder. For variable-rate NVQ design, the network's entropy coders may be lossless or near-lossless, and, following [14] and practical implementations employing arithmetic codes, we allow the entropy coders to operate at a higher dimension than the quantizers. For the case of fixed-rate NVQ design, the entropy coders are simply lossless codes operating at a fixed rate.

For any vector $X^n_{t,*}$ of source n-vectors, the quantizer encoder at node t, given by $\alpha_t : \mathcal{X}^n_{t,*} \to \mathcal{I}_{t,*}$, maps $X^n_{t,*}$ to a collection of indexes $i_{t,*}$ in the index set $\mathcal{I}_{t,*}$. In theory, $\mathcal{I}_{t,*}$ may be finite or countably infinite; in practice, we use finite $\mathcal{I}_{t,*}$. Here $i_{t,*} = (i_{t,S})_{S \in \mathcal{S}(t)}$, and for each $S \in \mathcal{S}(t)$, $i_{t,S} \in \mathcal{I}_{t,S}$. The collection of indexes $i_{t,*}$ is mapped by the fixed- or variable-rate entropy encoder at node t to a concatenated string of binary descriptions $c_{t,*} \in \mathcal{C}_{t,*}$. The channel conveys each individual description $c_{t,S}$ to precisely the receivers $r \in S$.

For any $r \in \mathcal{M}$, the entropy decoder at node r receives the codewords $c_{*,r}$ and side information Z^n_r and outputs index reconstructions $\hat{i}_{*,r} \in \mathcal{I}_{*,r}$. Except in a few special cases (e.g., when a coding error occurs), these are identical to the corresponding transmitted indexes. We denote the quantizer decoder at node r by $\beta_r : \mathcal{I}_{*,r} \times \mathcal{Z}^n_r \to \hat{\mathcal{X}}^n_{*,r}$. It maps indexes $\hat{i}_{*,r} \in \mathcal{I}_{*,r}$ and side information Z^n_r to a collection of reproduction vectors $\hat{X}^n_{*,r}$ such that $\hat{X}^n_{t,S,r} \in \hat{\mathcal{X}}^n_{t,S,r}$ for each $(t,S) \in \mathcal{T}(r)$. Let $\beta^n_{t,S,r}(\hat{i}_{*,r}, Z^n_r)$ denote the reproduction of $X^n_{t,S}$ made by receiver r. Then $\beta_r(\hat{i}_{*,r}, Z^n_r) = \hat{X}^n_{*,r}$ implies that $\beta_{t,S,r}(\hat{i}_{*,r}, Z^n_r) = \hat{X}^n_{t,S,r}$ for each $(t,S) \in \mathcal{T}(r)$. Note that $\beta_{t,S,r}$ depends on $\hat{i}_{*,r}$ rather than simply $\hat{i}_{t,S}$ since $\hat{i}_{*,r}$ is jointly decoded.

We associate two mappings with the entropy code. The first, $\ell_t : \mathcal{I}_{t,*} \to [0, \infty)$, is the rate used to describe $i_{t,*}$. In practice, $\ell_t(i_{t,*})$ is the length of the entropy code's corresponding codewords $c_{t,*}$; for entropy-constrained design, we set $\ell_t(i_{t,*})$ according to the entropy bound [14]. The rate used to describe a particular $i_{t,S}$ is given by $\ell_{t,S} : \mathcal{I}_{t,*} \to [0, \infty)$; it depends on all of the indexes from node t because the mapping is done jointly.

The second mapping, given by $f_r : \mathcal{I}_{*,r} \times \mathcal{Z}^n_r \to \mathcal{I}_{*,r}$, maps indexes $i_{*,r}$ transmitted to node r, together with side information Z^n_r, to the indexes $\hat{i}_{*,r}$ received after entropy decoding. Let $\alpha_{t,S}(x^n_{t,*})$ denote the component of α_t that produces codeword $i_{t,S}$. Then $\hat{i}_{*,r} = f_r(i_{*,r}, z^n_r)$, where $i_{*,r} = \left(\alpha_{t',S'}(X^n_{t',*})\right)_{(t',S') \in \mathcal{T}(r)}$. Typically, $f_r(i_{*,r}, z^n_r) = i_{*,r}$. Exceptions are caused by coding errors and the treatment of empty cells in the NVQ design process (as discussed in Section IV).

We restrict the joint encoding of the entropy codes to ensure unique decodability. We require that every entropy decoder be

able to uniquely decode each of its codewords using only the other codewords and the side information available to it. For example, in the MD system of Fig. 1(e), the entropy encoder at node 4 must encode indexes $i_{4,\{1,3\}}$ and $i_{4,\{2,3\}}$ so that they can be individually decoded at nodes 1 and 2. This requires that the coding be independent. However, our restriction is not so severe that all entropy codings in all systems need be done independently. In the system of Fig. 1(b), for example, a conditional entropy code for $i_{1,2}$ given $i_{1,\{2,3\}}$ can be used since the two indexes are jointly decoded at node 1. The restriction that our entropy codes be uniquely decodable does not imply that the encoders are one-to-one mappings; different symbols may be given the same description if the decoder has other information that allows it to distinguish them.

The performance of a network source code

$$Q^n = (\{\alpha_t\}_{t\in\mathcal{M}}, \{\beta_r\}_{r\in\mathcal{M}}, \{\ell_t\}_{t\in\mathcal{M}}, \{f_r\}_{r\in\mathcal{M}})$$

is measured in rate and distortion. In particular, for each $(t,S,r)\in\mathcal{T}$, let $d_{t,S,r} : \mathcal{X}_{t,S}\times\hat{\mathcal{X}}_{t,S,r}\to[0,\infty)$ be a nonnegative distortion measure between the alphabets $\mathcal{X}_{t,S}$ and $\hat{\mathcal{X}}_{t,S,r}$. We define the distortion between vectors of symbols to be additive, so that

$$d^n_{t,S,r}(x^n_{t,S},\hat{x}^n_{t,S,r}) = \sum_{k=1}^{n} d_{t,S,r}(x_{t,S}(k),\hat{x}_{t,S,r}(k)).$$

Here, $x_{t,S}(k)$ and $\hat{x}_{t,S,r}(k)$ denote the kth symbols in vectors $x^n_{t,S}$ and $\hat{x}^n_{t,S,r}$, respectively. Although not required for the validity of our results, for simplicity we assume that the distortion measures are identical and omit the subscripts. We also omit the superscript since it is clear from the arguments whether d is operating on a scalar or a vector.

We use $\mathcal{Q}^{\mathrm{fr},n}$ and $\mathcal{Q}^{\mathrm{vr},n}$ to denote the classes of n-dimensional fixed- and variable-rate NVQs, respectively. Let $x^n_{*,*} = (x^n_{1,*}, x^n_{2,*},\ldots,x^n_{M,*})$ denote a particular value for the collection of random source vectors $X^n_{*,*} = (X^n_{1,*}, X^n_{2,*},\ldots,X^n_{M,*})$. Similarly, let $z^n_* = (z^n_1, z^n_2,\ldots,z^n_M)$ denote a particular value for the side information $Z^n_* = (Z^n_1, Z^n_2,\ldots,Z^n_M)$. The (instantaneous) rate and distortion vectors associated with coding source vector $x^n_{*,*}$ with code $Q^n\in\mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),n}$ given side information z^n_* are, respectively[2]

$$\begin{aligned}\boldsymbol{r}(x^n_{*,*},Q^n) &= \left(r_{t,S}(x^n_{t,*},Q^n)\right)_{(t,S)\in\mathcal{S}}\\ &= \left(\ell_{t,S}(\alpha_t(x^n_{t,*}))\right)_{(t,S)\in\mathcal{S}}\\ \boldsymbol{d}(x^n_{*,*},z^n_*,Q^n) &= \left(d(x^n_{t,S},\hat{x}^n_{t,S,r})\right)_{(t,S,r)\in\mathcal{T}}\\ &= \left(d\left(x^n_{t,S},\beta_{t,S,r}(\hat{i}_{*,r},z^n_r)\right)\right)_{(t,S,r)\in\mathcal{T}}.\end{aligned}$$

Assume that the source and side information vectors together form a strictly stationary[3] ergodic random process with source distribution P. Let E denote the expectation with respect to P. The expected rate and distortion in describing n symbols from P with code Q^n are

$$\boldsymbol{R}(P,Q^n) = (R_{t,S}(P,Q^n))_{(t,S)\in\mathcal{S}}$$

and

$$\boldsymbol{D}(P,Q^n) = (D_{t,S,r}(P,Q^n))_{(t,S,r)\in\mathcal{T}}$$

where

$$\begin{aligned}R_{t,S}(P,Q^n) &= Er_{t,S}\left(X^n_{t,*},Q^n\right)\\ &= E\ell_{t,S}\left(\alpha_t(X^n_{t,*})\right)\\ D_{t,S,r}(P,Q^n) &= Ed\left(X^n_{t,S},\hat{X}^n_{t,S,r}\right)\\ &= Ed\left(X^n_{t,S},\beta_{t,S,r}(\hat{I}_{*,r},Z^n_r)\right).\end{aligned}$$

Given an M-node network with source distribution P, the fixed-rate achievable rate-distortion region is defined as

$$\mathcal{J}^{\mathrm{fr}}(P) = \overline{\bigcup_n \mathcal{J}^{\mathrm{fr},n}(P)}$$

where $\overline{A}$ denotes the closure of A and $\mathcal{J}^{\mathrm{fr},n}(P)$, the nth-order fixed-rate achievable rate-distortion region, is defined as

$$\mathcal{J}^{\mathrm{fr},n}(P) = \overline{\left\{\left((R_{t,S})_{(t,S)\in\mathcal{S}},(D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) : \exists Q^n\in\mathcal{Q}^{\mathrm{fr},n}\ \text{s.t.}\ \overline{R_{t,S}\ge\frac{1}{n}R_{t,S}(P,Q^n)\ \forall(t,S)\in\mathcal{S},}\ \overline{D_{t,S,r}\ge\frac{1}{n}D_{t,S,r}(P,Q^n)\ \forall(t,S,r)\in\mathcal{T}}\right\}}.$$

Analogous definitions hold for $\mathcal{J}^{\mathrm{vr}}(P)$ and $\mathcal{J}^{\mathrm{vr},n}(P)$.

A concatenation code argument demonstrates that any rate-distortion vector that can be achieved with a code of dimension n_0 can also be achieved for all n sufficiently large. Combining that result with a time-sharing argument proves the convexity of $\mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),n}(P)$. Lemmas 1 and 2, proved in Appendix I, make these ideas concrete.

Lemma 1: If P is a stationary source, then

$$\mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P) = \overline{\lim_{n\to\infty}\mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),n}(P)}.$$

Lemma 2: If P is a stationary source, then $\mathcal{J}^{\mathrm{fr}}(P)$ and $\mathcal{J}^{\mathrm{vr}}(P)$ are convex sets.

Since $\mathcal{J}^{\mathrm{fr}}(P)$ and $\mathcal{J}^{\mathrm{vr}}(P)$ are closed (by definition) and convex (by Lemma 2), each is entirely characterized by its support functional [42, p. 135]. These support functionals, here denoted by $j^{\mathrm{fr}}(P,\boldsymbol{a},\boldsymbol{b})$ and $j^{\mathrm{vr}}(P,\boldsymbol{a},\boldsymbol{b})$, are called the weighted fixed-rate and variable-rate operational rate-distortion functionals, respectively, where

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) = \inf_{(R_{*,*},D_{*,*})\in\mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)}\sum_{(t,S)\in\mathcal{S}}\left[a_{t,S}R_{t,S}+\sum_{r\in S}b_{t,S,r}D_{t,S,r}\right]$$

and $\boldsymbol{a} = (a_{t,S})_{(t,S)\in\mathcal{S}}$, $\boldsymbol{b} = (b_{t,S,r})_{(t,S,r)\in\mathcal{T}}$.

[2]The superscript $(\mathrm{fr}|\mathrm{vr})$ implies that the given result applies in parallel for fixed- and variable-rate.

[3]The condition of strict stationarity could be replaced by a condition of asymptotic mean stationarity in the results that follow. Strict stationarity is used for simplicity.

Lemma 3, proved in Appendix I, shows that $j^{(\mathrm{fr}|\mathrm{vr})}(P, \boldsymbol{a}, \boldsymbol{b})$ may also be described in terms of their nth-order equivalents.

Lemma 3: For any source P

$$j^{(\mathrm{fr}|\mathrm{vr})}(P, \boldsymbol{a}, \boldsymbol{b}) = \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P, \boldsymbol{a}, \boldsymbol{b})$$

where the nth-order fixed- and variable-rate weighted operational rate-distortion functionals are given by

$$j^{(\mathrm{fr}|\mathrm{vr}),n}(P, \boldsymbol{a}, \boldsymbol{b}) = \inf_{Q^n \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),n}} \sum_{(t,S)\in\mathcal{S}} \frac{1}{n}\left[a_{t,S} R_{t,S}(P, Q^n) + \sum_{r\in S} b_{t,S,r} D_{t,S,r}(P, Q^n)\right]. \quad (1)$$

The weighted operational rate-distortion functionals may be viewed as Lagrangians for minimizing a weighted sum of distortions subject to a collection of constraints on the corresponding rates. They can also be viewed as Lagrangians for minimizing a weighted sum of rates subject to a collection of constraints on the corresponding distortions. The weights $\boldsymbol{a}$ and $\boldsymbol{b}$ embody the code designer's priorities on the rates and distortions. They are constrained to be nonnegative, so that higher rates and distortions yield a higher Lagrangian cost. Practical code design depends on the *relative* values of these weights, and hence without loss of generality we set

$$\sum_{(t,S)\in\mathcal{S}} \left[a_{t,S} + \sum_{r\in S} b_{t,S,r}\right] = 1.$$

Together $\boldsymbol{a}$ and $\boldsymbol{b}$ also find an interpretation as a vector describing the direction of a hyperplane supporting the convex region $\mathcal{J}^{\mathrm{fr}}(P)$ or $\mathcal{J}^{\mathrm{vr}}(P)$ at a single point. Traversing all possible values traces out all points on the convex hull of the achievable rate-distortion region.

In practice, $\boldsymbol{a}$ and $\boldsymbol{b}$ cannot easily be chosen to guarantee specific rates or distortions: they reflect tradeoffs over the entire network. Often, the rates $\boldsymbol{R}(P, Q^n)$ of a network are desired to have particular values $\boldsymbol{R}^*$, and the distortions can be assigned relative priorities. We set $\boldsymbol{b}$ using those relative priorities and adopt a gradient descent approach to find appropriate values for $\boldsymbol{a}$. Denote by $Q^n(\boldsymbol{a}, \boldsymbol{b})$ the quantizer produced by our algorithm (described in the following section) when the Lagrangian constants are $(\boldsymbol{a}, \boldsymbol{b})$. The gradient descent minimizes the rate misfit $\chi(\boldsymbol{a}) = |\boldsymbol{R}(P, Q^n(\boldsymbol{a}, \boldsymbol{b})) - \boldsymbol{R}^*|^2$ as a function of $\boldsymbol{a}$.

We say that a fixed- or variable-rate network source code Q^n is *optimal* if it achieves a point on $j^{\mathrm{fr},n}(P, \boldsymbol{a}, \boldsymbol{b})$ or $j^{\mathrm{vr},n}(P, \boldsymbol{a}, \boldsymbol{b})$. Section III considers locally optimal NVQ design.

III. Locally Optimal NVQ Design

The goal in NVQ design is to find a code Q^n that minimizes the weighted cost in (1). Following the strategy of [13] and [14], we consider the necessary conditions for optimality of $\{\alpha_t\}$ and $\{\beta_r\}$ when the other system components are fixed. Using these conditions, we design NVQs through an iterative descent technique functionally equivalent to the generalized Lloyd algorithm. We compare the NVQ conditions with those for the point-to-point system of Fig. 1(a) to highlight the changes involved in moving from independent to network design.

The Lagrangian cost for a code Q^n is

$$j^n(P, \boldsymbol{a}, \boldsymbol{b}, Q^n) = \sum_{(t,S)\in\mathcal{S}} \frac{1}{n}\left[a_{t,S} R_{t,S}(P, Q^n) + \sum_{r\in S} b_{t,S,r} D_{t,S,r}(P, Q^n)\right]. \quad (2)$$

Our algorithm to design a code minimizing this cost is as follows.

Initialize the system components $\{\alpha_t\}$, $\{\beta_r\}$, lengths $\{\ell_t\}$, and mappings $\{f_r\}$.
Repeat
  Optimize each α_t and β_r in turn, holding every other component fixed.
  Update the coding rates $\{\ell_t\}$ and mappings $\{f_r\}$, holding all $\{\alpha_t\}$ and $\{\beta_r\}$ fixed.
Until the code's cost function j^n converges.

Provided that each optimization reduces the cost functional j^n, which is bounded below by zero, the algorithm will converge. In practice, we make approximations to simplify the optimizations and cannot always guarantee a reduction of the cost function (see Section IV for more details). However, except when close to a minimum, we do observe a consistent reduction in j^n in our experiments.

Decoder optimization is simple, even in the most general case. However, encoder optimization is not. Messages produced by an encoder are jointly decoded with messages from other encoders and with side information. However, each encoder knows neither the input to the other encoders nor the side information exactly, so it must operate based on the *expected* behavior of these other quantities. The expectation complicates the design process.

We next examine the component optimizations in detail, beginning with the decoders.

A. Optimal Decoders

Choose some $R \in \mathcal{M}$, and consider necessary conditions for the optimality of β_R when all encoders $\{\alpha_t\}_{t\in\mathcal{M}}$, all other decoders $\{\beta_r\}_{r\in\mathcal{M}\cap\{R\}^c}$, all length functions $\{\ell_t\}_{t\in\mathcal{M}}$, and all mappings $\{f_r\}_{r\in\mathcal{M}}$ are fixed. The optimal decoder $\beta_R^\star = (\beta^\star_{t,S,R})_{(t,S)\in\mathcal{T}(R)}$ for index vector $\hat{i}_{*,R} = (\hat{i}_{t,S})_{(t,S)\in\mathcal{T}(R)}$ and side information z_R^n satisfies

$$\beta^\star_{t,S,R}(\hat{i}_{*,R}, z_R^n) = \arg\min_{\hat{x}^n \in \hat{\mathcal{X}}^n_{t,S,R}} E\left[d(X^n_{t,S}, \hat{x}^n)\,\middle|\, Z_R^n = z_R^n, f_R\left((\alpha_{t',S'}(X^n_{t',*}))_{(t',S')\in\mathcal{T}(R)}, z_R^n\right) = \hat{i}_{*,R}\right]. \quad (3)$$

The expectation is with respect to the source distribution P.

The optimal decoder of the point-to-point system, shown in Fig. 1(a), satisfies

$$\beta^{\star}_{t,R,R}(\hat{i}_{t,R}) = \arg\min_{\hat{x}^n \in \hat{\mathcal{X}}^n_{t,R,R}} E\left[d(X^n_{t,R}, \hat{x}^n) \middle| f_R(\alpha_{t,R}(X^n_{t,R})) = \hat{i}_{t,R} \right] \quad (4)$$

where we have relabeled node 1 as t and node 2 as R. In the point-to-point case, the optimal reproduction for $\hat{i}_{t,R}$ is the vector $\hat{x}^n \in \hat{\mathcal{X}}^n_{t,R,R}$ that minimizes the expected distortion in the Voronoi cell indexed by $\hat{i}_{t,R}$. This Voronoi cell contains all source vectors $X^n_{t,R}$ such that $f_R(\alpha_{t,R}(X^n_{t,R})) = \hat{i}_{t,R}$. In the network case, the equation takes the same form, but with the Voronoi cell now indexed by a *collection* of indexes $\hat{i}_{*,R}$ *and* the side information z^n_R.

In general, the optimal network decoder depends on the full distribution P rather than merely the distribution of the message under consideration. This dependence arises from the joint nature of the decoding process.

We can extend the optimal decoder to allow for channel coding errors.[4] The distribution of channel coding errors is assumed to be independent of the sources and side information given the transmitted indexes. We describe the effect of channel errors on the indexes received by decoder r by the random mapping $G_r : \mathcal{I}_{*,r} \to \mathcal{I}_{*,r}$. Indexes $i_{*,r}$ transmitted by the encoders are transformed into $G_r(i_{*,r})$ by the channel, and decoded as $\hat{i}_{*,r} = f_r(G_r(i_{*,r}), z^n_r)$ by the entropy code. The optimal decoder is then given by (5) (at the bottom of the page).

B. Optimal Encoders

Now choose some $T \in \mathcal{M}$ and consider necessary conditions for the optimality of α_T when $\{\alpha_t\}_{t \in \mathcal{M} \cap \{T\}^c}$, $\{\beta_r\}_{r \in \mathcal{M}}$, $\{\ell_t\}_{t \in \mathcal{M}}$, $\{f_r\}_{r \in \mathcal{M}}$ are fixed. The optimal encoder $\alpha^{\star}_T(x^n_{T,*})$ satisfies the conditions in (6) (at the bottom of the page). Compare this to the equation given in (7) (at the bottom of the page) for optimizing the encoder $\alpha^{\star}_T(x^n_{T,r})$ of the point-to-point system. In the point-to-point case (7), the encoder's choice of index $i_{T,r}$ affects only one reproduction $\hat{X}_{T,r,r}$ at only one node r. In the network case (6), the indexes $\alpha_T(x^n_{T,*})$ chosen by encoder $\alpha^{\star}_T$ have a much more widespread impact. As expected, they affect the reproductions for $X^n_{T,*}$, but they also affect some reproductions for $X^n_{t,*}$ $(t \neq T)$ because each decoder β_r *jointly* maps its set of received indexes to the corresponding vector of reproductions. Thus, $i_{T,*}$ affects *all* reproductions at any node r to which T transmits a message. The minimization considers the weighted distortion over all of these reproductions.

The other major difference between the point-to-point and network equations is the expectation in the distortion term. The encoder in the point-to-point case knows $\hat{X}_{T,r,r}$ exactly for any possible choice of $i_{T,r}$. This is not true in the network case. For example, suppose encoder α_T transmits to node r. It does not know any of the indexes received by r from other nodes, nor the side information available to r. These unknowns are jointly decoded with the message(s) from α_T to produce the reproductions at r, and hence α_T cannot completely determine the reproductions knowing only its own choice of indexes. Encoder α_T must take a conditional expectation over the unknown quantities, conditioned on all of the information it does know, to determine its best choice of indexes. In (6), the use of capitalization for $I_{*,r} = (I_{t',S'})_{(t',S') \in \mathcal{T}(r)}$ denotes the fact that for any $t' \neq T$, $i_{t',S'}$ is unknown to α_T and must be treated as a random variable. The expectation is taken over the conditional distribution on $X^n_{t,S}$, $I_{*,r}$, and the side information Z^n_r given $X^n_{T,*} = x^n_{T,*}$ and $I_{T,*} = i_{T,*}$. For any $t' \neq T$, the distribution on $I_{t',S'}$ is governed by the corresponding (fixed) encoder $\alpha_{t'}$ together with the conditional distribution on the inputs to that encoder. Evaluating the conditional expectations in the equation for α_T is the primary difficulty in implementing the design algorithm, as discussed in Section IV.

We can extend the optimal encoder to allow for channel coding errors, representing their effects by a random mapping

[4]By incorporating the stochastic effects of channel coding errors into quantizer design we control the sensitivity of the source code to channel errors. We also allow for quantizer design in the case of a joint source–channel code. Since the source–channel separation theorem does not hold for network coding (see, for example, [1, pp. 448–449]), joint source–channel codes are required for optimal performance in some networks.

$$\begin{aligned}\beta^{\star}_{t,S,R}(\hat{i}_{*,R}, z^n_R) &= \arg\min_{\hat{x}^n \in \hat{\mathcal{X}}^n_{t,S,R}} E\left[d(X^n_{t,S}, \hat{x}^n) | Z^n_R = z^n_R, f_R(G_R(I_{*,R}), z^n_R) = \hat{i}_{*,R} \right] \\ &= \arg\min_{\hat{x}^n \in \hat{\mathcal{X}}^n_{t,S,R}} \sum_{i_{*,R} \in \mathcal{I}_{*,R}} \Big(E[d(X^n_{t,S}, \hat{x}^n) | Z^n_R = z^n_R, I_{*,r} = i_{*,R}] \\ &\qquad \cdot \Pr\Big(I_{*,R} = i_{*,R} | Z^n_R = z^n_R, f_R(G_R(I_{*,R}), z^n_R) = \hat{i}_{*,R} \Big) \Big). \end{aligned} \quad (5)$$

$$\begin{aligned}\alpha^{\star}_T(x^n_{T,*}) = \arg\min_{i_{T,*} \in \mathcal{I}_{T,*}} \Bigg[&\sum_{S \in S(T)} a_{T,S} \ell_{T,S}(i_{T,*}) \\ &+ \sum_{r \in S' : S' \in S(T)} \sum_{(t,S) \in \mathcal{T}(r)} b_{t,S,r} E[d(X^n_{t,S}, \beta_{t,S,r}(f_r(I_{*,r}, Z^n_r), Z^n_r)) | X^n_{T,*} = x^n_{T,*}, I_{T,*} = i_{T,*}] \Bigg] \end{aligned} \quad (6)$$

$$\alpha^{\star}_T(x^n_{T,r}) = \arg\min_{i_{T,r} \in \mathcal{I}_{T,r}} [a_{T,r} \ell_{T,r}(i_{T,r}) + b_{T,r,r} d\left(X^n_{T,r}, \beta_{T,r,r}(f_r(i_{T,r}))\right)] \quad (7)$$

G_r as before. The optimal encoder is then given by (8) (at the bottom of the page), where the expectation is over both the source and the channel error distributions.

C. Entropy Coding Rates

We now consider how the state of the art in lossless and near-lossless coding affects entropy-constrained design. Networks fall into three categories in this regard.

First are systems for which there exist practical codes achieving arbitrarily close to the entropy bounds and for which we also know the theoretically optimal codeword lengths. For example, in point-to-point coding (Fig. 1(a)), the entropy bound $R_{1,2} \geq H(I_{1,2})$ can be approximated using either Huffman or arithmetic coding. In addition, the codeword lengths given by $\ell_1(i_{1,2}) = -\log_2 p(i_{1,2})$ yield an expected rate equal to $H(I_{1,2})$ and satisfy Kraft's inequality.[5] For systems in this category (including MR, MD, and two-receiver BC systems), we follow [14] and design entropy-constrained NVQs using the theoretically optimal lengths.

Second are systems for which we cannot assign theoretical optimal codeword lengths, but we can still design practical lossless codes with rates close to the entropy bounds. This category includes WZ and two-access (2A) systems. Slepian and Wolf [6] give the achievable rate region for near-lossless coding in a 2A system (the generalization to M-encoder MA systems appears in [43]). However, these bounds alone are insufficient to determine the optimal codeword lengths. Generalizations of the point-to-point solution can yield lengths achieving points such as $(R_{2,1}, R_{3,1}) = (H(I_{2,1}), H(I_{3,1}|I_{2,1}))$ on the boundary of the achievable rate region. However, there is no 2A-equivalent of Kraft's inequality with which to prove that there could exist uniquely decodable codes with those lengths. We turn to practical codes, such as the 2A code in [44], and use their codeword lengths in entropy-constrained design.

Third are systems for which we cannot assign theoretically optimal codeword lengths and lack techniques for designing optimal codes. For example, in lossless M-receiver BC coding, even the optimal performance is unknown when $M > 2$. However, we can assign theoretical lengths and even design practical codes to achieve rates unobtainable by an independent approach. For systems in this category (which includes the three-node network of Fig. 3), we use the best known achievable rates, practical or theoretical, in the entropy constraint. Improved entropy constraints for these systems will likely become available as the field of lossless network source coding develops.

[5] As in [14], we allow nonintegral codeword lengths in our entropy constraints.

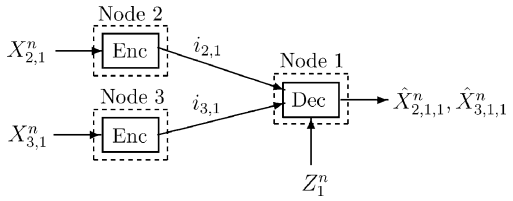


Fig. 4. The 2AWZ network.

If we assume that the near-lossless entropy codes achieve their asymptotic error probability of zero and that the distortion measure cannot be infinite, then there is no need to consider the increase in distortion resulting from entropy coding errors. For practical codes, which have a small but nonzero probability of error, the distortion increase should be taken into account in the near-lossless code optimization. This distortion increase is easily calculated in the case of squared-error distortion, for which an error that results in reproduction $\hat{x}'$ instead of $\hat{x}$ generates an additional distortion $(\hat{x} - \hat{x}')^2$ that is independent of the source distribution or training set. For distortion measures other than squared error, the distortion caused by an error must be calculated using the training set and the fixed encoders and decoders. The near-lossless design algorithm presented in [44] minimizes a weighted sum of rate and probability of error, but it can easily be altered to weigh each error by the expected distortion it generates as opposed to simply the error probability. This then ensures that our entropy code optimization is conducted with the same priorities as the quantizer design and will never result in an increased Lagrangian cost.

D. Network Scalability

Scalability is a key issue for some network source coding applications. If we train a code for a particular network, but that network is then altered by adding or deleting a node, how much code redesign is required to accommodate the change?

Starting with the WZ system of Fig. 1(d), consider adding a third node that describes a new source $X_{3,1}$ to the decoder at node 1. This yields a 2A system with side information at the decoder (a 2AWZ system), as depicted in Fig. 4. We describe two options for updating the code. In the first, the encoder at node 2 stays the same, the decoder retains its previous codebook for jointly decoding $(\hat{i}_{2,1}, Z_1)$, and we train a new conditional codebook for decoding $\hat{i}_{3,1}$, conditioned on $(\hat{i}_{2,1}, Z_1)$. This greedy approach allows us to keep our previous system intact and simply add a new component. However, the correlation between $X_{3,1}$ and $X_{2,1}$ is exploited only in the decoding of $X_{3,1}$ and not $X_{2,1}$. Optimally exploiting the correlation so as to minimize our Lagrangian cost (2) requires a second, global

$$\alpha^\star_T(x^n_{T,*}) = \underset{i_{T,*} \in \mathcal{I}_{T,*}}{\arg\min} \left[\sum_{S \in S(T)} a_{T,S} \ell_{T,S}(i_{T,*}) + \sum_{r \in S' : S' \in S(T)} \sum_{(t,S) \in \mathcal{T}(r)} b_{t,S,r} E[d(X^n_{t,S}, \beta_{t,S,r}(f_r(G_r(I_{*,r}), Z^n_r), Z^n_r)) | X^n_{T,*} = x^n_{T,*}, I_{T,*} = i_{T,*}] \right]. \quad (8)$$

approach, in which we extend the original WZ codebook to jointly decode all three inputs $(\hat{i}_{2,1}, \hat{i}_{3,1}, Z_1)$. For this, the whole system must be retrained.

Now consider deleting node 2 from the 2A system of Fig. 4, so that the decoder no longer receives $\hat{i}_{2,1}$. We can form a new decoder from the existing one by simply averaging over the various codewords for different values of $\hat{i}_{2,1}$, for each fixed $(\hat{i}_{3,1}, Z_1)$. Provided that the existing decoder cells are convex with respect to $\hat{i}_{2,1}$, this is a good strategy. Otherwise, global retraining is necessary.

These cases exemplify canonical network alterations: we can form a new code with little cost by making local, greedily designed component additions or subtractions, or we can aim for optimal performance by retraining the entire network at a greater cost.

IV. Implementation

In this section, we consider the implementation of the design algorithm. We focus on practical evaluation of the terms in the optimality conditions represented by (3) and (6) from the previous section. We also discuss the use of side information at the decoders.

In practice, we optimize our codes with respect to a training data set. A key assumption of that approach is that the empirical joint distribution of the training set is close to the true joint source distribution. Under this assumption, the training set allows us to evaluate the expectations required to optimize the encoders and decoders of the network.

Several experimental results in Section V assume that the entropy codes achieve their asymptotic bound of error probability zero. This does not imply that $f_r(i_{*,r}, z_r^n) = i_{*,r}$. Nonidentity mappings must be used to deal with empty cells that arise during training. In designing a point-to-point VQ, there may be training iterations in which no training vectors are mapped to a particular Voronoi cell because its codeword is not the nearest neighbor of any of the training vectors. In entropy-constrained VQ (ECVQ), such cells are removed from consideration by associating with their index an infinite rate. Thus, an encoder designed to minimize $aD + bR$ for some $b > 0$ never uses that index. The same empty cell phenomenon occurs in network coding. However, in an MA system, a decoder cell is jointly indexed by two or more encoder indexes. Rates are not associated with individual cells, but with each separate index. Even if index pair (i, j) corresponds to an empty cell, this does not mean that either i or j individually should be assigned infinite rate (this should happen to index i, for instance, only if *all* cells $(i, \cdot)$ were to become empty). Since we cannot, in general, remove cells from consideration by altering their rate, and since the encoders work independently, it is possible that an empty cell may inadvertently be indexed. This must be avoided, since to save rate we usually do not require that the entropy code preserve the indexes of empty cells. In practice, we begin by assuming that no cells are empty, and as cells do become empty we merge each of them with a full neighboring cell (when side information z^n is present, we can choose a different merging for each z^n). The cell merging is incorporated into the entropy code to allow a saving in rate. Any reference to an empty cell is redirected to the nonempty neighbor, and this redirection is made known to the encoders through the mappings $\{f_r\}$. As in ECVQ, no cells that become empty are ever filled again; training vectors mapped to an empty cell indexed by (i_2, i_3) are always redirected to the appropriate nonempty neighbor $f_r(i_2, i_3, z^n)$. Thus, $f_r(i_2, i_3, z^n)$ fills a dual role of handling both empty cells (in a "once empty always empty" manner) and near-lossless coding errors.

The terms in the optimality condition for a network decoder (3) are no more difficult to evaluate than those for the point-to-point decoder (4). For the point-to-point decoder, optimization trains each codeword using the set of training vectors falling into that codeword's Voronoi cell. For example, when the distortion measure is a squared error, evaluating the expectation in (4) places each codeword at the mean value of its associated training vectors. Network decoder optimization (3) requires no change in approach.

The difficulties in NVQ design arise in the optimization of the network encoders. In point-to-point VQ, encoder optimization is implemented through a nearest neighbor search: the encoder chooses the index $i_{T,r}$ that minimizes the Lagrangian in (7). In NVQ design, we again search over all possible encoder indexes, but computing the Lagrangian in (6) may require the evaluation of a conditional expectation. We divide network encoders into two types, one for which the conditional expectation is necessary and one for which it is not. Each encoder's type is determined by the nature of the decoders it transmits to. We call any decoder that uses side information, or that receives messages from two or more encoders, a *joint* decoder. We call all other decoders *individual* decoders.

A *Type I* encoder transmits messages only to individual decoders. Type I encoders know exactly the reproductions associated with each possible index choice, and hence no conditional expectation is necessary. Optimization of Type I encoders is done via a straightforward nearest neighbor search in the same way as point-to-point encoder optimization.

A *Type II* encoder transmits to one or more joint decoders. Since it lacks some of the information used by the joint decoders, a Type II encoder cannot determine the decoders' reproductions. Its optimization therefore requires a conditional expectation over the unknown messages or side information at each joint decoder. The discussion that follows illustrates the implementation of Type II encoders using two examples. The first is a 2A system with side information at the decoder (the 2AWZ system). The second is a three-node network, which adds the additional complication of having Type II encoders that transmit to more than one joint decoder.

In addition to the implementation of Type II encoders, the 2AWZ example addresses the use of side information at a network decoder and the initialization of the components of a network system. The discussion generalizes from 2AWZ to arbitrary networks.

A. Two-User Multiple Access With Side Information (2AWZ)

We here discuss Type II encoder implementation, the use of side information, and initialization methods for the 2AWZ network shown in Fig. 4. The two encoders $\alpha_2 : \mathcal{X}_{2,1}^n \to \mathcal{I}_{2,1}$ and

$\alpha_3 : \mathcal{X}_{3,1}^n \to \mathcal{I}_{3,1}$ operate on sources $X_{2,1}^n$ and $X_{3,1}^n$, respectively, to produce indexes $i_{2,1}$ and $i_{3,1}$. The decoder

$$\beta_1 : \mathcal{I}_{2,1} \times \mathcal{I}_{3,1} \times {\mathcal{Z}_1}^n \to \hat{\mathcal{X}}_{2,1,1}^n \times \hat{\mathcal{X}}_{3,1,1}^n$$

jointly decodes the corresponding received indexes

$$(\hat{i}_{2,1,1}, \hat{i}_{3,1,1}) = f_1(i_{2,1}, i_{3,1}, z_1^n)$$

using side information z_1^n. We denote the decoder reproductions individually as

$$\beta_{2,1,1}(\hat{i}_{2,1,1}, \hat{i}_{3,1,1}, z_1^n) = \hat{x}_{2,1,1}^n$$

and

$$\beta_{3,1,1}(\hat{i}_{2,1,1}, \hat{i}_{3,1,1}, z_1^n) = \hat{x}_{3,1,1}^n.$$

We assume that $(X_{2,1}^n, X_{3,1}^n, Z_1^n)$ are dependent random variables. For notational convenience, we use Z_1^n and Z^n interchangeably.

For now, assume that $\mathcal{Z}^n$ is discrete and that $|\mathcal{Z}^n|$ is small, so that the total number of possible decoder codewords, $|\mathcal{I}_{2,1}||\mathcal{I}_{3,1}||\mathcal{Z}^n|$, is significantly smaller than the size of the training set. Later, we discuss how to work with a large or continuous $\mathcal{Z}^n$.

1) Encoder Implementation: Considering α_2 and rewriting the expectations from (6) in terms of sums over the sets $\mathcal{I}_{3,1}$ and $\mathcal{Z}^n$ gives

$$\begin{aligned}\alpha_2^\star(x_{2,1}^n) = \arg\min_{i_{2,1}} \Bigg[& a_{2,1} \left|\ell_2(i_{2,1})\right| \\ + \sum_{i_{3,1}} \sum_{z^n} & \Pr(\alpha_3(X_{3,1}^n) = i_{3,1}, Z^n = z^n | X_{2,1}^n = x_{2,1}^n) \\ & \cdot \Big(b_{2,1,1} d\left(x_{2,1}^n, \beta_{2,1,1}\left(f_1(i_{2,1}, i_{3,1}, z^n), z^n\right)\right) \\ & + b_{3,1,1} E\left[d(X_{3,1}^n, \beta_{3,1,1}(f_1(i_{2,1}, i_{3,1}, z^n), z^n)) \right| \\ & \left. X_{2,1}^n = x_{2,1}^n, \alpha_3(X_{3,1}^n) = i_{3,1}, Z^n = z^n \right] \Big) \Bigg]. \end{aligned} \quad (9)$$

An analytical model for the source distribution is generally unavailable, so we estimate $\Pr(\alpha_3(X_{3,1}^n) = i_{3,1}, Z^n = z^n | x_{2,1}^n)$ and the expectation over $X_{3,1}^n$ using the training data. Since the alphabet $\mathcal{X}_{2,1}^n$ is in general very large (e.g., $|\mathcal{X}_{2,1}^4| = 256^4$ for an 8-bit greyscale image and VQ dimension 4), the number of conditional probability terms to be estimated is very large. Given a limited training set size, possible estimation techniques use kernels, histograms, and combinations of the two [45]. We here use histograms due to their low computational complexity. We partition $\mathcal{X}_{2,1}^n$ into a finite number of bins and estimate conditional distributions over $\mathcal{I}_{3,1} \times \mathcal{Z}^n$ for each bin. Denote by $\delta_2 : \mathcal{X}_{2,1}^n \to \mathcal{K}_{2,1} = \{1, \ldots, |\mathcal{K}_{2,1}|\}$ the function that maps a sample $x_{2,1}^n$ to the index $k_{2,1}$ of its corresponding histogram bin.

Denote by $\Gamma = \{(x_{2,1}^n, x_{3,1}^n, z^n)\}$ the *list*[6] of training vectors, and define

$$\begin{aligned}\Gamma(k_{2,1}, i_{3,1}, z^n) = \{(x_{2,1}'^n, x_{3,1}'^n, z'^n) \in \Gamma : \delta_2(x_{2,1}'^n) = k_{2,1}, \\ \alpha_3(x_{3,1}'^n) = i_{3,1}, z'^n = z^n\} \\ \Gamma(k_{2,1}) = \{(x_{2,1}'^n, x_{3,1}'^n, z'^n) \in \Gamma : \delta_2(x_{2,1}'^n) = k_{2,1}\}.\end{aligned}$$

[6]Γ is defined as a list rather than a set, because any training vector that appears multiple times in Γ should be counted multiple times in any list size or sum calculation.

We estimate $\Pr(\alpha_3(X_{3,1}^n) = i_{3,1}, Z^n = z^n | x_{2,1}^n)$ in (9) by replacing the condition on $x_{2,1}^n$ with a condition on $\delta_2(x_{2,1}^n)$ giving

$$\begin{aligned}\Pr(\alpha_3(X_{3,1}^n) = & i_{3,1}, Z^n = z^n | x_{2,1}^n) \\ & \approx \Pr(\alpha_3(X_{3,1}^n) = i_{3,1}, Z^n = z^n | \delta_2(x_{2,1}^n)) \\ & = \frac{\left|\Gamma(\delta_2(x_{2,1}^n), i_{3,1}, z^n)\right|}{\left|\Gamma(\delta_2(x_{2,1}^n))\right|}\end{aligned}$$

which we evaluate from the training data using the current (fixed) α_3. We evaluate the expectation over $X_{3,1}^n$ using the known mappings (from the previous optimization[7]) for all of the training vectors.

2) Convergence: In the preceding discussion, we make approximations that allow us to reduce the number of conditional distributions to be estimated. These approximations represent a deviation from the optimal encoder as specified by the design equations, and, as a result, convergence of the algorithm is no longer guaranteed (although it is observed in practice given a training set of appropriate size). We now show that by altering our cost function, convergence can be guaranteed at the cost of some performance degradation.

Let $K_{2,1} = \delta_2(X_{2,1}^n)$ and $K_{3,1} = \delta_3(X_{3,1}^n)$. Suppose

$$\begin{aligned}X_{2,1}^n \to K_{2,1} \to K_{3,1} \to X_{3,1}^n \\ (X_{2,1}^n, X_{3,1}^n) \to (K_{2,1}, K_{3,1}) \to Z^n\end{aligned}$$

where $X \to Y \to Z$ specifies that (X, Y, Z) forms a Markov chain. Then our approximation of conditioning on $k_{2,1} = \delta_2(x_{2,1}^n)$ and $k_{3,1} = \delta_3(x_{3,1}^n)$ becomes exact, and we can implement the optimal encoder exactly. These Markov properties do not hold in general, but by building a probability model in which they are forced to hold, we get a design algorithm that is guaranteed to converge. For any source distribution $P(x_{2,1}^n, x_{3,1}^n, z^n, k_{2,1}, k_{3,1})$, there exists a corresponding distribution $\hat{P}(x_{2,1}^n, x_{3,1}^n, z^n, k_{2,1}, k_{3,1})$ that satisfies the Markov properties, where

$$\begin{aligned}\hat{P}(x_{2,1}^n, x_{3,1}^n, z^n, k_{2,1}, k_{3,1}) = & P(x_{2,1}^n) P(k_{2,1} | x_{2,1}^n) \\ & \times P(k_{3,1} | k_{2,1}) P(x_{3,1}^n | k_{3,1}) \\ & \times P(z^n | k_{2,1}, k_{3,1}).\end{aligned}$$

Define a new cost function $\hat{j}^n(\hat{P}, \boldsymbol{a}, \boldsymbol{b}, Q^n)$ that differs from $j^n(P, \boldsymbol{a}, \boldsymbol{b}, Q^n)$ in (2) in that we take expectations with respect to $\hat{P}$ rather than P.[8] Thus, $\hat{j}^n$ gives the expected system performance with respect to $\hat{P}$, where $\hat{P}$ has the properties we desire. Both the optimal encoders and the optimal decoder for $\hat{j}^n$ can be implemented exactly (in a computationally feasible manner), and hence convergence is guaranteed. However, the code is now optimized with respect to $\hat{P}$ rather than the true distribution P, and it does not perform as well in practice as a code designed with the nonconvergent algorithm on P, as shown by the experimental results in Sections V-A and V-B.

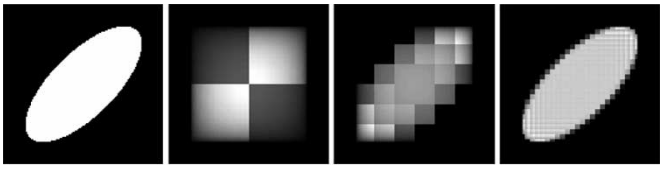
We give two examples to build intuition of how enforcement of the Markov property alters the joint distribution. For simplicity, we omit the side information.

[7]All components except α_2 are held fixed from the previous optimization.

[8]In (2), we take expectations over a distribution of the form $P(x_{2,1}^n, x_{3,1}^n, z^n)$. This can easily be extended to the form $P(x_{2,1}^n, x_{3,1}^n, z^n, k_{2,1}, k_{3,1})$, since $k_{2,1}$ and $k_{3,1}$ are deterministic functions of $x_{2,1}^n$ and $x_{3,1}^n$.

TABLE I
APPLICATION OF THE MARKOV CONSTRAINT TO A PAIR OF GAUSSIAN SOURCES

$x_{2,1}\backslash x_{3,1}$	$x_{3,1} \leq 0$	$x_{3,1} > 0$
$x_{2,1} > 0$	$(2-\eta)P(x_{2,1})P(x_{3,1})$	$\eta P(x_{2,1})P(x_{3,1})$
$x_{2,1} \leq 0$	$\eta P(x_{2,1})P(x_{3,1})$	$(2-\eta)P(x_{2,1})P(x_{3,1})$



(a) (b) (c) (d)

Fig. 5. A discrete Markov constraint example. (a) The distribution P, uniform over an ellipse. (b)–(d) $\hat{P}$ for $|\mathcal{K}_{2,1}| = |\mathcal{K}_{3,1}| = K$ when: (b) $K = 2^1$, (c) $K = 2^3$, (d) $K = 2^5$.

Let vector $(X_{2,1}, X_{3,1})$ be Gaussian with mean 0, variance 1, and correlation ρ, i.e.,

$$P(x_{2,1}, x_{3,1}) = \frac{1}{2\pi(1-\rho^2)} \exp\left(-\frac{x_{2,1}^2 + 2\rho x_{2,1}x_{3,1} + x_{3,1}^2}{2(1-\rho^2)}\right)$$

giving marginals

$$P(x_{2,1}) = (1/\sqrt{2\pi})\exp(-x_{2,1}^2/2)$$

and

$$P(x_{3,1}) = (1/\sqrt{2\pi})\exp(-x_{3,1}^2/2).$$

We consider scalar quantization ($n = 1$) and allocate one bit to each of $k_{2,1}$ and $k_{3,1}$. Choose $\delta_2(x) = \delta_3(x) = 1(x > 0)$, where $1(\cdot)$ is the indicator function, and let

$$\eta = 4\int_0^\infty \int_0^\infty P(x_{2,1}, x_{3,1}) dx_{2,1} dx_{3,1}.$$

Then $\hat{P}$ for the four possible values of $(k_{2,1}, k_{3,1})$ is given by Table I. It is a weighted product of the marginals of P. The weights, which reflect the correlation, are such that the integral over each quadrant is the same for $\hat{P}$ and P. For independent sources, $\rho = 0$, $\eta = 1$, and we find $P = \hat{P}$. For highly correlated sources with $\rho \approx 1$, we have $\eta \approx 2$, and $\hat{P}$ smears the positive correlation over the first and third quadrants. The second and fourth quadrants have little or no probability mass, consistent with the original distribution.

In general, $\hat{P}$ is a weighted product of the marginals of P in which the weighting can be different for different values of $(K_{2,1}, K_{3,1})$. As $|\mathcal{K}_{2,1}|$ and $|\mathcal{K}_{3,1}|$ grow, $\hat{P}$ more closely resembles P. Fig. 5 illustrates this point for a discrete distribution on a square grid with $|\mathcal{X}_{2,1}| = |\mathcal{X}_{3,1}| = 2^7$ ($n = 1$ as before). The original uniform distribution (Fig. 5(a)) is approximated to greater accuracy (Fig. 5(b)–(d)) by increasing $|\mathcal{K}_{2,1}|$ and $|\mathcal{K}_{3,1}|$.

For our experiments we use 8-bit greyscale images at dimension 4, with $|\mathcal{X}_{2,1}^4| = |\mathcal{X}_{3,1}^4| = 2^{32}$ and $|\mathcal{K}_{2,1}| = |\mathcal{K}_{3,1}| \approx 2^9$. Thus, $\hat{P}$ will only coarsely model the source correlation, but this appears to be sufficient for the particular sources and rates that we use in the experiments.

3) Initialization: The first step in implementing generalized Lloyd design is to initialize the system encoders and decoders. Since iterative descent design can at best give only a locally optimal solution, initialization can have a significant impact on final performance.[9] When theory suggests a useful structure for a codebook, we make use of it in initialization. Here we outline two initialization methods. One is based on point-to-point coding methods and is suitable for weakly correlated sources and side information; the other is based on the binning structure used by practical distributed codes [31], [37] (mirroring the binning structure used to prove the Slepian–Wolf theorem for lossless coding) and is suitable for strongly correlated sources and side information. For both approaches we initialize the entropy codes with the codeword lengths used for fixed-rate coding, and with identity mappings $\{f_r\}$. Consequently, we equate $\hat{i}_{*,r}$ with $i_{*,r}$ in the following discussion on quantizer initialization.

In the point-to-point method, we design a codebook with cells convex with respect to each of $i_{2,1}$ and $i_{3,1}$. We begin by designing a point-to-point VQ for each of $X_{2,1}^n$ and $X_{3,1}^n$. We initialize the network encoders α_2 and α_3 as the corresponding point-to-point encoders. If there were no side information, we could construct the joint decoder by simply taking the cross product of the two point-to-point codebooks. With side information, we need to specify $|\mathcal{Z}^n|$ initial codewords for each $(i_{2,1}, i_{3,1})$ pair. Using the point-to-point encoders, we partition the training set into lists $\Gamma(i_{2,1}, i_{3,1}, z^n)$, where

$$\Gamma(i_{2,1}, i_{3,1}, z^n) = \{(x_{2,1}'^n, x_{3,1}'^n, z'^n) \in \Gamma : \alpha_2(x_{2,1}'^n) = i_{2,1}, \alpha_3(x_{3,1}'^n) = i_{3,1}, z'^n = z^n\}.$$

We then initialize the decoder β_1 by setting each $\beta_{2,1,1}(i_{2,1}, i_{3,1}, z^n)$ and $\beta_{3,1,1}(i_{2,1}, i_{3,1}, z^n)$ to be the centroids (with respect to the appropriate distortion measures) of the list $\Gamma(i_{2,1}, i_{3,1}, z^n)$.

For sources and side informations with high correlations, we use a binning structure. The resulting codebook has cells that are nonconvex for a given $i_{2,1}$ and $i_{3,1}$: noncontiguous regions of one source alphabet can be quantized to the same index, and the decoder relies on the other source and the side information to distinguish the correct region. The set of noncontiguous regions assigned to a particular index is called a *coset*. For the WZ system, in which source $x_{3,1}^n$ does not appear, we can create a binning structure with $2^{r_c} \leq |\mathcal{Z}^n|$ cosets starting with a quantizer that maps each z^n into one of 2^{r_c} different values and a lattice-based codebook for source $x_{2,1}^n$ at rate $R_{2,1}$. We translate the lattice by small amounts (less than or equal to half the distance between adjacent lattice points) in 2^{r_c} different directions; the images of a lattice point under the different translations are allocated to different cosets. Each individual coset consists of a translated copy of the original lattice.[10] For an MA code, the

[9] A variety of annealing techniques have been applied to traditional VQ design (e.g., [46], [47]) in an attempt to address the local optimality problem. These techniques can be generalized to NVQ design. While several authors have conjectured that these techniques yield global optima, this conjecture remains unproven.

[10] An alternative to obtaining the cosets by translation is to encode the source using the original lattice, then partition the training vectors mapped to a particular lattice point into 2^{r_c} sets using their quantized z^n values, and initialize as described in the low correlation method.

base lattice is formed from a cross product of lattices for each of the individual sources.

While our design algorithm can produce optimized VQs with a binning structure, we have not observed it to do so if a binning structure was not used in initialization.

4) Detailed Side Information: The previous discussion assumes $|\mathcal{Z}^n|$ is small. When $|\mathcal{Z}^n|$ is large or $\mathcal{Z}$ is continuous, the number of decoder codewords required prohibits a practical implementation as above. We solve this problem by coarsely quantizing the side information before it is given to the decoder. The quantized side information is denoted by an index $k_Z \in \mathcal{K}_Z = \{1, 2, \ldots, |\mathcal{K}_Z|\}$, and we redefine β_1 so that

$$\beta_1 : \mathcal{I}_{2,1} \times \mathcal{I}_{3,1} \times \mathcal{K}_Z \to \hat{\mathcal{X}}^n_{2,1,1} \times \hat{\mathcal{X}}^n_{3,1,1}$$

instead of

$$\beta_1 : \mathcal{I}_{2,1} \times \mathcal{I}_{3,1} \times \mathcal{Z}^n \to \hat{\mathcal{X}}^n_{2,1,1} \times \hat{\mathcal{X}}^n_{3,1,1}.$$

We allow a different quantization for each pair of received indexes $(\hat{i}_{2,1}, \hat{i}_{3,1})$, and denote by $\delta_Z : \mathcal{I}_{2,1} \times \mathcal{I}_{3,1} \times \mathcal{Z}^n \to \mathcal{K}_Z$ the quantizer encoder that determines k_Z given received index pair $(\hat{i}_{2,1}, \hat{i}_{3,1})$ and side information z^n. The quantizer codewords are denoted $\{\phi_Z(\hat{i}_{2,1}, \hat{i}_{3,1}, k_Z)\}_{k_Z=1}^{K_Z}$, and are initialized by clustering on the set of training vectors

$$\Gamma(i_{2,1}, i_{3,1}) = \{(x'^n_{2,1}, x'^n_{3,1}, z'^n) : \alpha_2(x'^n_{2,1}) = i_{2,1}, \alpha_3(x'^n_{3,1}) = i_{3,1}, z'^n \in \mathcal{Z}^n\}.$$

We create $|\mathcal{K}_Z|$ clusters from this list, number the clusters from 1 to K_Z, and set the decoder codewords $\beta_{2,1,1}(\hat{i}_{2,1}, \hat{i}_{3,1}, k_Z)$ and $\beta_{3,1,1}(\hat{i}_{2,1}, \hat{i}_{3,1}, k_Z)$ as the centroids of the appropriate cluster.

Quantizing the side information is a practical rather than an optimal strategy. However, provided the quantization is of a significantly higher rate than that used for the source messages, we can capture essentially all of the correlation between the source messages and side information. The experimental results of Section V-A suggest that on practical data sets we can keep $|\mathcal{K}_Z|$ (and hence the number of decoder codewords) small, while paying little or no penalty in rate-distortion performance.

B. A General Three-Node Network

In this subsection, we use the example of a general three-node network to discuss the implementation of a Type II encoder that transmits to more than one other node.

Consider the implementation of the design conditions for encoder 2 for the three-node network shown in Fig. 3. Encoder 2 participates in two 2AWZ subsystems: it cooperates with encoder 3 to send information to decoder 1 and with encoder 1 to send information to decoder 3. Since the indexes chosen by encoder 2 affect reproductions at both nodes 1 and 3, as shown in Fig. 6, the distortion terms for both decoders must be evaluated in a single minimization. From the design equations, $\alpha_2^\star$ is given by

$$\alpha_2^\star(x^n_{2,*}) = \arg\min_{i_{2,*} \in \mathcal{I}_{2,*}} \left(\sum_{S \in \mathcal{S}(2)} a_{2,S} \ell_{2,S}(i_{2,*}) + \sum_{(t,S) \in \mathcal{T}(1)} b_{t,S,1} E\left[d(X^n_{t,S}, \beta_{t,S,1}(\hat{I}_{*,1}, Z^n_1)) \middle| X^n_{2,*} = x^n_{2,*}, I_{2,*} = i_{2,*} \right] \right.$$

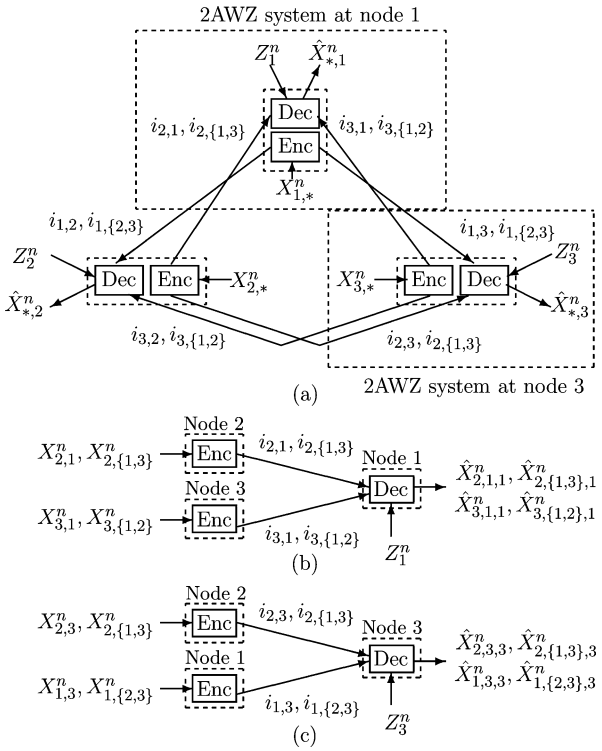


Fig. 6. Optimal encoding at node 2. (a) The estimated performance for a given index set $i_{2,*}$ can be found by summing the performance in two linked 2AWZ subsystems. (b), (c) The two subsystems.

$$+ \sum_{(t,S) \in \mathcal{T}(3)} b_{t,S,3} E\left[d(X^n_{t,S}, \beta_{t,S,3}(\hat{I}_{*,3}, Z^n_3)) \middle| X^n_{2,*} = x^n_{2,*}, I_{2,*} = i_{2,*} \right] \Bigg) \tag{10}$$

where

$$\begin{aligned} \mathcal{S}(2) &= \{\{1,3\}, 1, 3\}, \\ \mathcal{T}(1) &= \{(2, \{1,3\}), (2,1), (3, \{1,2\}), (3,1)\}, \\ \mathcal{T}(3) &= \{(1, \{2,3\}), (1,3), (2, \{1,3\}), (2,3)\}. \\ \hat{I}_{*,1} &= f_1\left((\alpha_{t,S}(X^n_{t,*})_{(t,S) \in \mathcal{T}(1)}, Z^n_1\right) \\ \hat{I}_{*,3} &= f_3\left((\alpha_{t,S}(X^n_{t,*})_{(t,S) \in \mathcal{T}(3)}, Z^n_3\right). \end{aligned}$$

All terms in (10) are of similar form to those for the 2AWZ system, and we can use the same approximation methods to evaluate the conditional expectations. In general, any Type II encoder sending to more than one other node can be designed using the same approach for sending to only one other node; there are just more distortion and rate terms to evaluate.

V. Experimental Results

In this section, we build NVQs for different network systems and present experimental results. We discuss four systems in detail: the 2AWZ and general three-node networks described in Section IV, and the MD and BC networks introduced in Section I. For each of the four systems, we give a brief introduction, a discussion of entropy codeword length selection for entropy-constrained coding, and experimental results. The experiments compare the performance of NVQs to that of independent

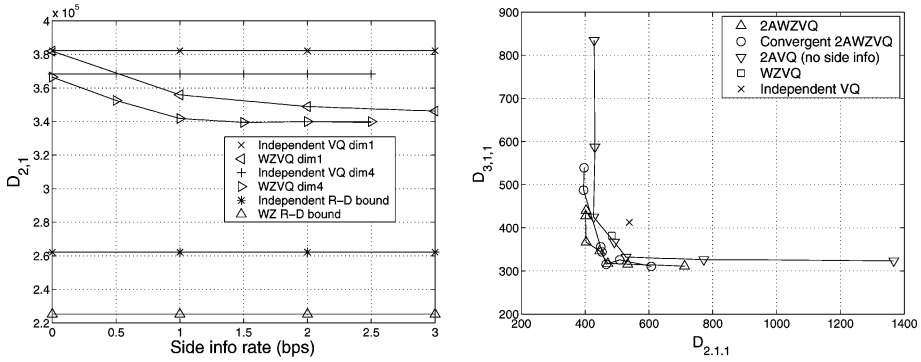


Fig. 7. (a) WZVQ performance as a function of side information rate $\log_2(|\mathcal{K}_Z|)/n$ for jointly Gaussian source and side information ($\rho = 0.375$). (b) Various coding performances for the 2AWZ system on satellite data.

VQs and to available rate-distortion (R-D) bounds. Additionally, we examine the scalability of network codes using a ring network as an example. All experiments use the mean squared error (MSE) distortion measure.

A. The 2AWZ Network

The preceding section treats this system in detail. We therefore skip its introduction.

There are currently no practical entropy codes for the 2AWZ system, nor any provably optimal theoretical codeword lengths. However, there are practical entropy codes for the 2A system [44]. We perform both 2AWZ and 2A experiments, the former at fixed rate only, the latter at both fixed rate and variable rate. Our variable-rate 2A design uses the near-lossless codes from [44]. The nonzero contribution from coding errors is included in the reported distortion.

We conduct four experiments. The first studies the use and impact of side information in a WZ system, removing one of the users of the 2AWZ system for simplicity of result presentation. The second compares network to independent coding performance on the full 2AWZ system. The third compares fixed- and variable-rate coding performance on the 2A system. The fourth investigates the benefit of initializing the decoder to have a binning structure, again using a WZ system for result simplicity.

For the first (WZ) experiment, we generate independent and identically distributed (i.i.d.) jointly Gaussian data with correlation $\rho = 0.375$ between the source and the side information. We quantize the side information as discussed in Section IV. We use fixed-rate codes of rate 1 bit per sample (bps) and vary the vector dimension n and the number $|\mathcal{K}_Z|$ of values used to quantize the side information. For good practical code performance we want $|\mathcal{K}_Z|$ high to make full use of the correlation between the message and the side information. However, we must limit $|\mathcal{K}_Z|$ to limit the number of decoder codewords and hence our design complexity. We compare the performance of WZVQs and independent VQs to two R-D bounds; one is the WZ R-D bound, the other is the point-to-point R-D bound (which uses no side information).

The distortion for the different codes is shown in Fig. 7(a). The R-D bounds are independent of $|\mathcal{K}_Z|$ and are plotted for comparative purposes; they show the theoretically optimal achievable performance for any vector dimension n, and, in the WZ R-D case, arbitrarily high $|\mathcal{K}_Z|$. The results show that the use of side information in the WZ codes improves performance by approximately 0.4 dB and bridges 20% of the gap in distortion between independent coding and the WZ R-D bound at dimension 4 and correlation 0.375. The gain is even higher at dimension 1. For both dimensions it is of comparable size to the difference in the two R-D bounds, suggesting that the WZ codes are making efficient use of the side information. The results also show that almost all of the benefit of side information is captured with a low value of $|\mathcal{K}_Z|$, validating quantization of the side information as a method for reducing design complexity.

For the second experiment, we train and test various fixed-rate codes for the full 2AWZ system. We use satellite weather images for our data set.[11] All codes use vector dimension 4, rate 0.75 bps, and Lagrangian weights $a_{2,1} = a_{3,1} = 0$, $b_{2,1,1}+b_{3,1,1}=1$. We use $|\mathcal{K}_Z|=16$ different values to quantize the side information.

Fig. 7(b) shows a plot of distortions $D_{2,1,1}$ and $D_{3,1,1}$ for the various coding techniques. For any choice of Lagrangian weights, independent code design yields the same code and hence contributes a single point to the graph. For network code design, the Lagrangian weights trade off the importance of the two reproductions and yield different codes. We display the performance of network codes both with (2AWZVQ) and without (2AVQ) the use of side information. For the 2AWZVQ codes, we include results obtained using the convergent as well as the nonconvergent algorithm. We also show the performance achieved by using two separately decoded WZVQs, one for each source. The 2AWZVQs show a gain of at least 1.17 dB in each reproduction over the independent VQs. This gain arises from both the joint decoding of messages and the use of side information; the distortions achieved by the 2AVQs and by the WZVQs suggest that both contributions are significant. The results obtained by the convergent and nonconvergent algorithms are similar on this data set.

The third experiment compares the performance of fixed- versus variable-rate codes for the 2A system. We again use satellite weather images for the data and use vector dimension 4. We set the Lagrangian weights $b_{2,1,1} = b_{3,1,1} = 1$ and $a_{2,1} = a_{3,1} = a$, where a is varied to produce codes targeting different

[11]This data set is described in detail in Appendix II.

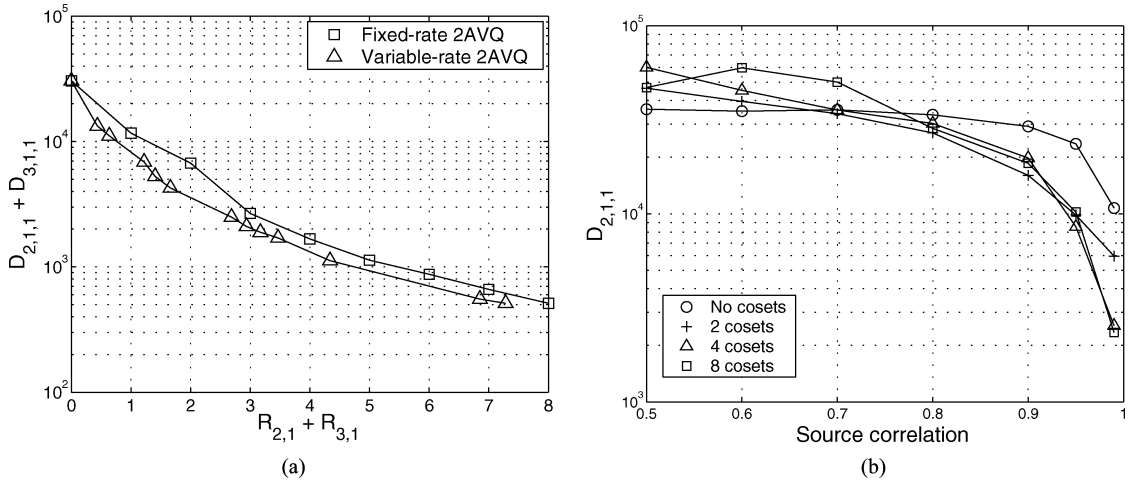


Fig. 8. (a) Comparison of fixed- and variable-rate coding performances for the 2A system. (b) WZ code performance as a function of source and side information correlation and the number of cosets used in decoder initialization.

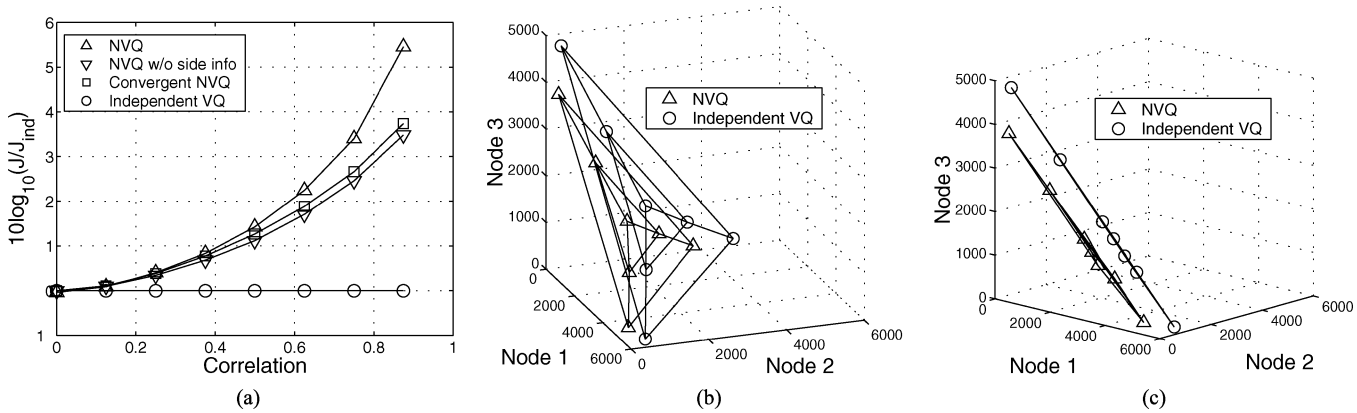


Fig. 9. Efficiency of network source coding versus independent coding. (a) Overall performance as a function of correlation. (b),(c) Weighted sum distortion at each node as a function of the Lagrangian parameters (shown from two different angles).

points on the convex hull of the achievable R-D region. For variable-rate design, we use the codeword lengths and mappings of real near-lossless 2A codes. Fig. 8(a) shows the sum $D_{2,1,1} + D_{3,1,1}$ of the two distortions as a function of the total rate $R_{2,1}+R_{3,1}$ for several fixed- and variable-rate codes. The variable-rate codes consistently outperform their fixed-rate counterparts by 1.2 dB.

The fourth experiment uses WZ codes to investigate the benefits of initializing with a binning structure when the source and side information are highly correlated. The source and side information are i.i.d. jointly Gaussian, with mean 0, variance 1, and correlation ρ. We use fixed-rate codes of dimension 1, $|\mathcal{K}_Z| = 64$ different values to quantize the side information, and initialize the decoder with 2^{r_c} cosets, $r_c \in \{0,1,2,3\}$. Fig. 8 shows the performance obtained by different codes as a function of ρ. For low correlations, such as $\rho = 0.5$, a binning structure significantly hampers performance, and the training optimization removes the binning structure as best it can. The final performance is similar to that of the code with no binning. For high correlation, performance is significantly improved using a binning-structured decoder. The desired number of cosets increases with the correlation.

B. A General Three-Node Network

Section IV includes a detailed introduction of general three-node networks. Optimal entropy codes and coding bounds are currently unavailable for the general three-node network. We perform all of our experiments using fixed-rate codes.

We conduct two experiments for the general three-node network. The first shows the efficiency of NVQs as a function of intersource correlation; the second compares the tradeoff in performance at each node as a function of the Lagrangian weights controlling the optimization. All experiments show fixed-rate coding results at vector dimension 4 and rate 0.5 bps for each of the nine sources. The side information used by the decoder at each node consists of the three sources to be encoded at that node and is quantized to $|\mathcal{K}_Z| = 16$ levels.

The performance gain of NVQs over independent VQs is a function of intersource correlation. Fig. 9(a) shows a plot of the Lagrangian cost (2) in decibels as a function of correlation for i.i.d. jointly Gaussian sources. For each sample, the correlation between any two sources has a constant value ρ. Equal weighting is given to each reproduction. As the correlation between sources increases, the performance gain of NVQs over independent VQs increases significantly, exceeding 2 dB for

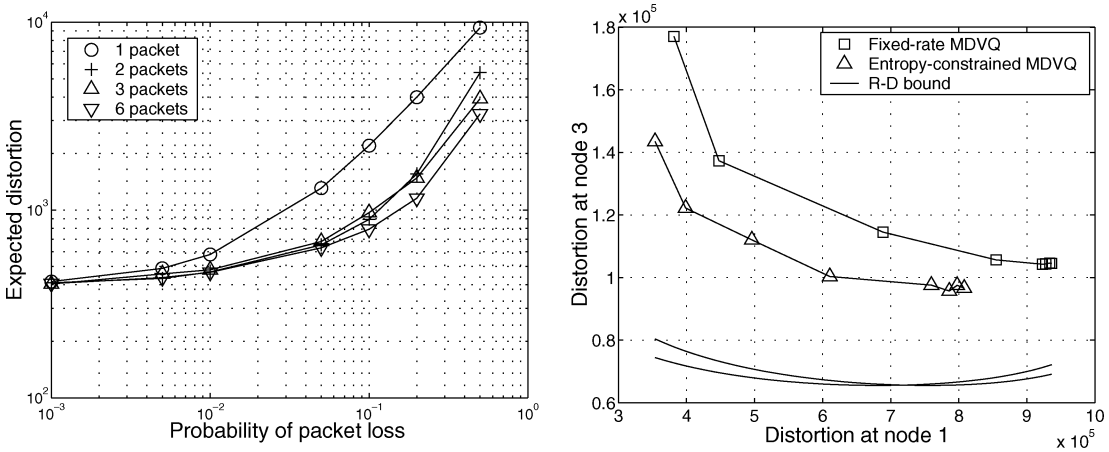


Fig. 10. (a) MDVQ performance on the satellite data as a function of the number of descriptions per vector and the channel failure probability. (b) Fixed-rate and entropy-constrained MDVQ performance on Gaussian data compared to the R-D bound.

$\rho \geq 0.6$. Also, unlike the 2AWZ experiments, for this system and data set the alterations required to ensure convergent design do impair performance.

Fig. 9(b) and (c) shows the weighted sum of the distortions at each of the three nodes using the satellite data set. Varying the Lagrangian weights $\{b_{t,S,r}\}$ traces out the surface shown from two different angles in the figures (the weights $\{a_{t,S}\}$ are extraneous here because the rates are fixed). We constrain the Lagrangian weights to keep all weights corresponding to reproductions at the same node equal. The surface corresponding to the NVQs lies approximately 1 dB closer to the origin than that of independent VQs, indicating an average 1-dB improvement over the set of source reproductions at each node.

C. An MD System

An MD system transmitting descriptions over K unreliable channels can give rise to $2^K - 1$ nontrivial sets of received descriptions. We cast the system into a network model as in Fig. 1(e) by treating the decoder for each of the $2^K - 1$ nontrivial sets as a separate node in our network. We label the encoder with index $M = 2^K$ and each decoder with the integer representation of a binary vector $\boldsymbol{e} = (e_1, \ldots, e_K)$, where $e_k = 1$ if channel k is operational and 0 otherwise. The K channel descriptions now correspond to K network messages; network message k is received by all decoders that have $e_k = 1$. Although some decoders receive two or more descriptions, each outputs only one reproduction since all descriptions are of the same original source. The system contains only one encoder and no side information, so the encoder is of Type I and can be implemented without approximations.

Since the K descriptions must be individually decodable (for each description there is some decoder that receives that and only that description), the optimal coding bound for description k is the entropy $H(I_k)$ of the index used for message k. This bound can be approximated in practice by entropy codes (e.g., arithmetic codes), and we associate with each index i_k the optimal average length $-\log_2 \Pr(I_k = i_k)$.

Given the probability of each $\boldsymbol{e} \in \{0,1\}^K$, we can minimize the expected distortion of our code by setting the weight on each reproduction's distortion to be the probability of receiving exactly the set of descriptions used to make that reproduction. We then adjust the relative sizes of the weights on the description rates to achieve our desired code rates.

We conduct two experiments to demonstrate the performance of MD codes. In the first, we use the satellite weather data set to train and test fixed-rate MDVQs. Each code uses a different number of descriptions to encode each four-dimensional data vector, but all have the same total encoded bit rate of 6 bits per vector. The MDVQs considered use: one 6-bit description, two 3-bit descriptions, three 2-bit descriptions, and six 1-bit descriptions, respectively. For each code, we transmit different descriptions of the same vector on different channels, and we assume that the different channels all have equal failure rates. Fig. 10(a) shows the expected reproduction distortion as a function of channel failure probability and the number of descriptions used. Moving from a single description (as in traditional coding) to two descriptions greatly slows the degradation in performance as a function of channel failure probability. Using more than two descriptions yields even better performance.

The second experiment compares fixed-rate and entropy-constrained two-description, four-dimensional MDVQ performance to the R-D bound using i.i.d. Gaussian data. We choose a rate of 1 bps per description and design codes for different probabilities of channel failure. Each code is characterized by the distortions it achieves at the three decoders. However, for all of the codes designed in this experiment we found the distortion at node 2 to be almost constant, so we plot the distortion at node 3 against that at node 1, as shown in Fig. 10(b). We also plot the R-D bound for the code rate used in the experiment. The bound depends on node 2's distortion, which varied very slightly over the results; the two lines defining the bound correspond to the smallest and largest values of node 2's distortion observed. The results demonstrate the reduction in distortion achieved by variable-rate compared to fixed-rate coding. This reduction varies from 0.5 dB for low

channel failure probability to 1.3 dB for high channel failure probability.

D. A Broadcast Network

The general broadcast network is similar in form to the multiple description network. The sender (node 1) transmits messages to all possible subsets of receivers (nodes $2, \ldots, M$). In the two-receiver BC network there are three sources: the transmitter sends "private" sources $X_{1,2}^n$ and $X_{1,3}^n$ to the receivers at nodes 2 and 3, respectively, and "common" source $X_{1,\{2,3\}}^n$ to both receivers. Using a network approach, node 1 jointly encodes $X_{1,2}^n$, $X_{1,3}^n$, and $X_{1,\{2,3\}}^n$; node 2 jointly decodes $\hat{X}_{1,2,2}^n$ and $\hat{X}_{1,\{2,3\},2}^n$; node 3 jointly decodes $\hat{X}_{1,3,3}^n$ and $\hat{X}_{1,\{2,3\},3}^n$. There is only one encoder and the system uses no side information, so the encoder is of Type I and can be implemented without approximations.

For entropy coding in two-receiver BC systems, we can calculate the optimal achievable rates and we have practical code design schemes to achieve some (but not all) points on the boundary of the achievable rate region. In particular, the lossless two-receiver BC system is a special case of the system addressed in [4]. Using [4, Theorem 4] and following the ideas in [48] to bound the cardinality of an auxiliary random variable, we have the following result.

Lemma 4: Let $(I_{1,2}, I_{1,3}, I_{1,\{2,3\}})$ be drawn i.i.d. according to $P(i_{1,2}, i_{1,3}, i_{1,\{2,3\}})$. The set of achievable rate vectors for lossless source coding in a two-receiver BC system is the set of $(R_{1,2}, R_{1,3}, R_{1,\{2,3\}})$ vectors satisfying

$$\begin{aligned} R_{1,\{2,3\}} &\geq I(I_{1,2}, I_{1,3}, I_{1,\{2,3\}}; W) \\ R_{1,2} &\geq H(I_{1,2}, I_{1,\{2,3\}}|W) \\ R_{1,3} &\geq H(I_{1,3}, I_{1,\{2,3\}}|W) \end{aligned}$$

for some joint probability mass function $p(i_{1,2}, i_{1,3}, i_{1,\{2,3\}}, w)$, where

$$\sum_{w \in \mathcal{W}} p(i_{1,2}, i_{1,3}, i_{1,\{2,3\}}, w) = P(i_{1,2}, i_{1,3}, i_{1,\{2,3\}})$$

and $|\mathcal{W}| \leq |\mathcal{I}_{1,2}| \times |\mathcal{I}_{1,3}| \times |\mathcal{I}_{1,\{2,3\}}| + 3$.

Given the bounded cardinality of auxiliary random variable W, we can find the point in the achievable rate region that minimizes the Lagrangian cost (2) for a fixed quantizer. Some of these points can be well approximated by existing practical codes. The most intuitive point practically achievable is that for which $W = I_{1,\{2,3\}}$. At this point, the rate triple

$$\begin{aligned} R_{1,\{2,3\}} &= H(I_{1,\{2,3\}}) \\ R_{1,2} &= H(I_{1,2}|I_{1,\{2,3\}}) \\ R_{1,3} &= H(I_{1,3}|I_{1,\{2,3\}}) \end{aligned} \tag{11}$$

can be achieved through a sequential coding scheme [5]. That is, first, the encoder encodes the common index $I_{1,\{2,3\}}$ using an entropy code of rate $H(I_{1,\{2,3\}})$ and transmits it to both decoders. Both decoders can reconstruct $I_{1,\{2,3\}}$ without error. The encoder then describes the private indexes $I_{1,2}$ and $I_{1,3}$ to decoders 2 and 3 using conditional entropy codes conditioned on $I_{1,\{2,3\}}$ at rates $H(I_{1,2}|I_{1,\{2,3\}})$ and $H(I_{1,3}|I_{1,\{2,3\}})$, respectively. Since both decoders know $I_{1,\{2,3\}}$, they can reconstruct $I_{1,2}$ and $I_{1,3}$, respectively, without error.

Not all achievable points for the two-receiver case can be well approximated in practice, and for M-receiver $(M > 2)$ lossless BC codes, the optimal lossless coding performance is unknown (see [26, Theorem 1, Theorem 2, Corollary 1] for inner and outer bounds). As a result, we rely on currently available lossless codes and their corresponding entropy bounds, even when these bounds are not optimal. In our experiments, we use sequential techniques and the entropy bounds in (11) for entropy-constrained two-receiver BCVQ design.

We now present variable-rate coding results for the two-receiver BC system. All experiments use coding dimension 4 and are performed on the satellite data set. We assume throughout that the common source is equally important to both receivers and that there is a tradeoff between the two private sources. Thus, we choose the Lagrangian distortion coefficients as $b_{1,\{2,3\},2} = b_{1,\{2,3\},3} = \theta/2$, $b_{1,2,2} = \sigma$, $b_{1,3,3} = 1 - \sigma$, and we use the gradient descent approach of Section II to adjust the rate coefficients $(a_{1,2}, a_{1,3}, a_{1,\{2,3\}})$ so as to achieve our desired rates of 0.5 bps for each of the three sources.

Fig. 11 shows the lower convex hull of the achieved distortions

$$D = ((D_{1,\{2,3\},2} + D_{1,\{2,3\},3})/2, D_{1,2,2}, D_{1,3,3})$$

rendering the three-dimensional distortion space from two different angles. We compare this convex hull to the convex hull of distortions $D = (D_{1,\{2,3\}}, D_{1,2}, D_{1,3})$ achieved using variable-rate independent VQs. Since the rates assigned to each receiver are identical for the BCVQ and for the independent codes, the lower the convex hull of distortions, the better the performance. The convex hull achieved by the BCVQs (solid line) is significantly lower than that achieved by independent VQs (dashed line).

Finally, Fig. 12 compares variable-rate BCVQ with fixed-rate BCVQ. As expected, variable-rate coding consistently outperforms fixed-rate coding.

E. Network Scalability

The complexity of network code design depends on the following factors.

The Number of Codewords: If we assign every codeword a weight equal to the number of encoders that access that codeword, then design complexity is linear in the sum weight of all network codewords. The number of encoders accessing a particular codeword is equal to the node's in-degree. Network design is therefore linear in the number of nodes M when the in-degree of each node is kept constant and exponential in M when in-degree grows linearly with M.

The Size of the Training Set: Code design complexity is linear in the size of the training set, which must be large enough to ensure that each encoder's histogram estimation of the data's joint distribution is accurate. The number of histogram bins required by an encoder transmitting to node r is linear in the number of codewords at node r. Thus, the number of training vectors needed is at most linear in the maximum number of codewords at any node. In addition, the size of each training set vector is linear in the number of network sources. If the in-degree of each node is constant as M increases (so that the number of codewords per node is constant, but the number

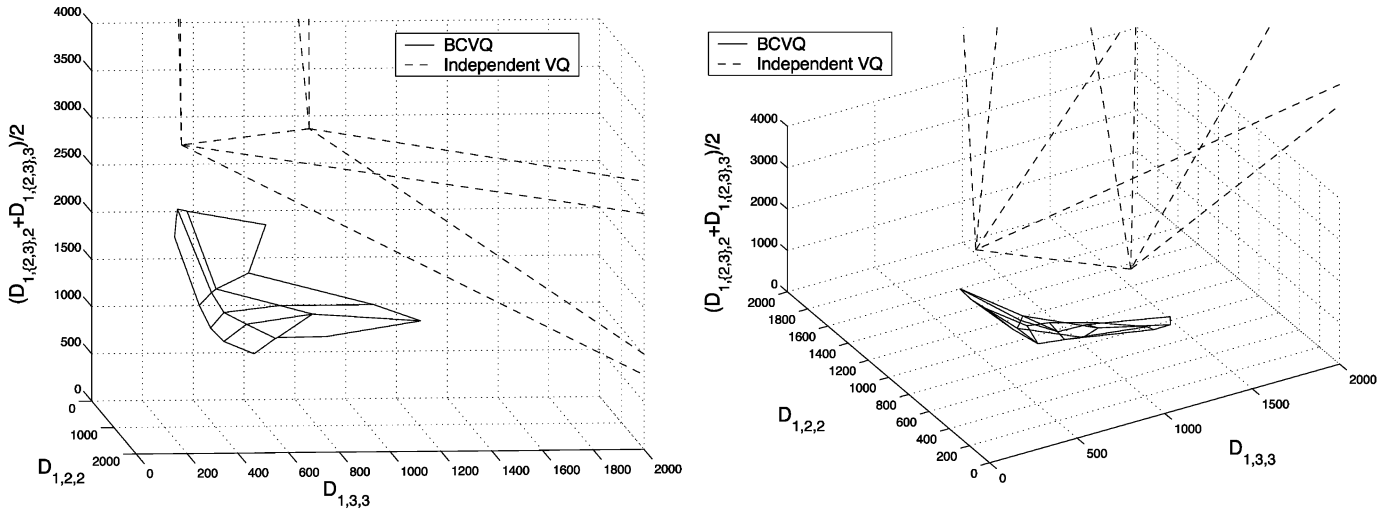


Fig. 11. Variable-rate BCVQ versus variable-rate independent VQ from two different angles.

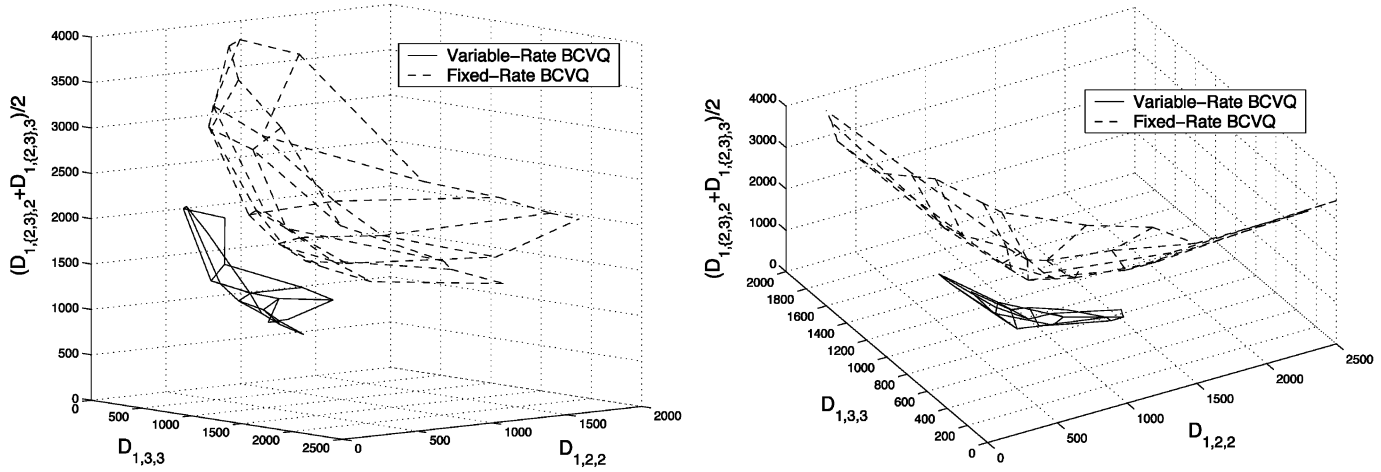


Fig. 12. Variable-rate BCVQ versus fixed-rate BCVQ from two different angles.

of sources is increasing linearly with M), then the required training set size will increase linearly, but if the in-degree of each node increases with M, the required training set size may increase exponentially.

Fixed- Versus Variable-Rate Design: Training complexity is roughly the same for both fixed- and variable-rate design. However, if the network rates must meet specific constraints, then the Lagrangian parameters in variable-rate design must be optimized appropriately. Using a conjugate gradient approach, this increases design complexity approximately by a factor equal to the number of Lagrangian parameters. Symmetry in a network can be exploited to constrain the parameter optimization.

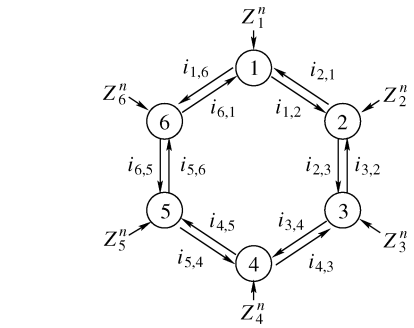
Fig. 13(a) shows a ring network in which each node communicates with its two neighbors. This is an example of a network in which the in-degree of each node remains constant as M increases; the design time therefore increases linearly with M as evidenced by the experimental design times shown in Fig. 13(b). These design times are for fixed-rate quantizer design on a 1-GHz Intel Xeon processor at vector dimension 4, with 4 bits per message and 4-bit side information quantization.

Table II indicates approximate design complexity for several networks by counting the total weight of network codewords. We assume a vector dimension of 4, with 2 bits per network message and 4 bits for side information quantization. Two types of BC network are considered: a limited one in which there is private information for each individual receiver, but only one common information (for all receivers), and a full one in which every subset of receivers will receive a different common information. We see that our design algorithm is not suitable for large MA, fully connected BC, and fully connected general networks. However, almost no real networks will be so connected as to have nodes that transmit a separate message to every possible subset of other nodes. Practical networks would be much more likely to follow a model such as the limited BC or the M-node ring, for which the design complexity scales linearly with M and for which our algorithm is appropriate. The exponential increase in design complexity for MA networks is a concern; suboptimal design techniques would need to be adopted for large M, such as dividing the nodes into fixed-sized groups and jointly decoding each group separately.

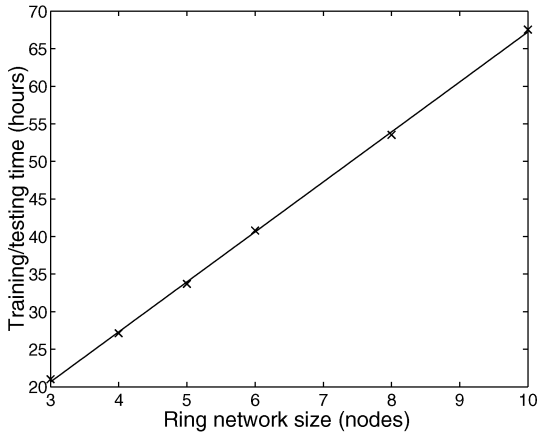
Once a network has been designed, encoding and decoding complexity can be made very low if desired. Approximating the optimal encoders using hierarchical coding allows both encoding and decoding to be done via table lookup [12].

TABLE II
TOTAL WEIGHT OF CODEWORDS FOR VARIOUS SYSTEMS

Network	Codewords per decoder	Total weight of codewords
MA	2^{2M}	$M2^{2M}$
M-receiver limited BC	16	$16M$
M-receiver full BC	2^{2^M}	$M2^{2^M}$
M-node ring	256	$512M$
General M-node	$2^{(M-1)2^{M-1}+4}$	$M(M-1)2^{(M-1)2^{M-1}+4}$



(a)



(b)

Fig. 13. (a) The network messages and side informations of a six-node ring network. (b) Design time for a ring network.

VI. SUMMARY

We present a set of optimality conditions to design locally optimal vector quantizers for general networks. These conditions yield an iterative design algorithm that generalizes existing rate-distortion-optimal VQ design (e.g., for point-to-point, MR, and MD systems) and extends it to WZ, MA, BC, and multinode systems.

Both fixed- and variable-rate VQ design are considered. For some network systems, variable-rate VQ design is complicated either by the fact that the theoretically optimal codeword lengths for the entropy code are unknown, or by the absence of good techniques for designing practical codes to approximate the optimal performance. In these cases, we must optimize relative to the best available bounds on the entropy code's codeword lengths.

We discuss implementation of the algorithm derived from the optimal design conditions. The primary difficulty in implementation is the evaluation of conditional expectations required to design the optimal encoders for a network with joint decoders. We provide approximations to reduce the computational complexity of evaluating the expectations and also to reliably estimate the joint statistics of the training data. Making these approximations removes the guarantee of convergence in iterative code design. In practice, however, we do observe convergence, which suggests that the approximations are reasonable. When required, we show how to ensure convergence at some cost in rate-distortion performance.

The NVQ experiments demonstrate the performance improvements that network-based design yields over independent design. When applied to a satellite weather data set, 2AWZ and three-node network codes both show distortion improvements of more than 1 dB over independent coding. This increase results from the ability of network-designed codes to exploit the redundancy between the different sources in the network; point-to-point design treats every source as independent and thus does not benefit from this type of redundancy. For networks where sources are highly correlated, such as sensor networks, network-based coding can be significantly more efficient.

APPENDIX I
PROOFS OF LEMMAS

We assume in the proofs of Lemmas 1 and 2 that for each alphabet $\hat{\mathcal{X}}_{t,S,r}$ there exists an escape character $\kappa_{t,S,r} \in \hat{\mathcal{X}}_{t,S,r}$ such that $Ed(X_{t,S}, \kappa_{t,S,r}) \leq D_{\max} < \infty$.

Lemma 1: If P is a stationary source, then

$$\mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P) = \overline{\lim_{n\to\infty} \mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),n}(P)}.$$

Proof: The (identical) proofs for the fixed- and variable-rate regions are done in parallel.

It is sufficient to show that $N_1 \geq 1$ and

$$\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \in \mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),N_1}(P)$$

imply that

$$\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \in \mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),N}(P)$$

for all N sufficiently large.

From the definition of $\mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),N_1}(P)$, for any

$$\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \in \mathcal{J}^{(\mathrm{fr}|\mathrm{vr}),N_1}(P)$$

and any $\epsilon > 0$, there exists a quantizer $Q^{N_1} \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N_1}$ such that

$$\frac{1}{N_1}\left(\boldsymbol{R}(P,Q^{N_1}), \boldsymbol{D}(P,Q^{N_1})\right) \leq \left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) + (\epsilon,\ldots,\epsilon).$$

For any $N > N_1$, let k_1 and k_2 be the unique pair of integers such that $N = k_1 N_1 + k_2$, $k_1 \geq 0$, $0 \leq k_2 < N_1$. We construct a quantizer $Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}$ of dimension N that acts in the following way. Any input $x_{*,*}^N$ is broken into k_1 subvectors of dimension N_1, with a final subvector of dimension k_2. Each of the first k_1 subvectors is coded with the codewords of Q^{N_1}. For the final subvector, each symbol is coded using the set of escape characters $\kappa_{*,*}$. For any stationary P, quantizer Q^N has rates and distortions satisfying

$$\frac{1}{N}\boldsymbol{R}(P,Q^N) \leq \frac{k_1}{N}\boldsymbol{R}(P,Q^{N_1}) \leq (R_{t,S})_{(t,S)\in\mathcal{S}} + (\epsilon,\ldots,\epsilon)$$
$$\frac{1}{N}\boldsymbol{D}(P,Q^N) \leq \frac{1}{N}\left(k_1\boldsymbol{D}(P,Q^{N_1}) + k_2\left(D_{\max},\ldots,D_{\max}\right)\right) \leq (D_{t,S,r})_{(t,S,r)\in\mathcal{T}} + (2\epsilon,\ldots,2\epsilon)$$

for N large enough since k_2 is bounded and the escape distortion $D_{\max} < \infty$. Since ϵ is arbitrary, the desired result follows. □

Lemma 2: If P is a stationary source, then $\mathcal{J}^{\mathrm{fr}}(P)$ and $\mathcal{J}^{\mathrm{vr}}(P)$ are convex sets.

Proof: For any

$$\left((R'_{t,S})_{(t,S)\in\mathcal{S}}, (D'_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \in \mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)$$

and

$$\left((R''_{t,S})_{(t,S)\in\mathcal{S}}, (D''_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \in \mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)$$

and any $\lambda \in [0,1]$, let

$$\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) = \lambda\left((R'_{t,S})_{(t,S)\in\mathcal{S}}, (D'_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) + (1-\lambda)\left((R''_{t,S})_{(t,S)\in\mathcal{S}}, (D''_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right).$$

We need to show that $\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right)$ lies in $\mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)$.

Combining the definition of $\mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)$ with Lemma 1 implies that for any $\epsilon > 0$, there exists an integer N_0 such that for any $N \geq N_0$, $N_1 = \lfloor \lambda N \rfloor$, and $N_2 = N - N_1$, there exist quantizers $Q^{N_1} \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N_1}$ and $Q^{N_2} \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N_2}$ such that

$$\frac{1}{N_1}\left(\boldsymbol{R}(P,Q^{N_1}), \boldsymbol{D}(P,Q^{N_1})\right) \leq \left((R'_{t,S})_{(t,S)\in\mathcal{S}}, (D'_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) + (\epsilon,\ldots,\epsilon)$$
$$\frac{1}{N_2}\left(\boldsymbol{R}(P,Q^{N_2}), \boldsymbol{D}(P,Q^{N_2})\right) \leq \left((R''_{t,S})_{(t,S)\in\mathcal{S}}, (D''_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) + (\epsilon,\ldots,\epsilon).$$

For any $N \geq N_0$, design an N-dimensional quantizer $Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}$ by concatenating codewords from Q^{N_1} and Q^{N_2}. For stationary P, the rates and distortions of Q^N satisfy

$$\frac{1}{N}\boldsymbol{R}(P,Q^N) = \frac{1}{N}\left(\boldsymbol{R}(P,Q^{N_1}) + \boldsymbol{R}(P,Q^{N_2})\right) \leq (R_{t,S})_{(t,S)\in\mathcal{S}} + \frac{1}{N}(R''_{t,S})_{(t,S)\in\mathcal{S}} + (\epsilon,\ldots,\epsilon)$$
$$\frac{1}{N}\boldsymbol{D}(P,Q^N) = \frac{1}{N}\left(\boldsymbol{D}(P,Q^{N_1}) + \boldsymbol{D}(P,Q^{N_2})\right) \leq (D_{t,S,r})_{(t,S,r)\in\mathcal{T}} + \frac{1}{N}(D''_{t,S,r})_{(t,S,r)\in\mathcal{T}} + (\epsilon,\ldots,\epsilon)$$

where $D''_{t,S,r} \leq D_{\max} < \infty$. Thus, there exists a sequence of codes $\{Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}\}$ such that

$$\frac{1}{N}\left(\boldsymbol{R}(P,Q^N), \boldsymbol{D}(P,Q^N)\right) \leq \left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) + 2(\epsilon,\ldots,\epsilon)$$

for N large enough and arbitrary ϵ, giving the desired result. □

Lemma 3: For any source P,

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) = \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b})$$

where

$$j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b}) = \inf_{Q^n\in\mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),n}} \sum_{(t,S)\in\mathcal{S}} \frac{1}{n}\left[a_{t,S}R_{t,S}(P,Q^n) + \sum_{r\in S} b_{t,S,r}D_{t,S,r}(P,Q^n)\right].$$

Proof: The proof is in two parts; the first shows that

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \geq \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b})$$

and the second that

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \leq \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b}).$$

From the definition of $j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b})$, for any $\epsilon > 0$, there exists $\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \in \mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)$ such that

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \geq \sum_{(t,S)\in\mathcal{S}}\left[a_{t,S}R_{t,S} + \sum_{r\in S} b_{t,S,r}D_{t,S,r}\right] - \epsilon.$$

Further, by the definition of $\mathcal{J}^{(\mathrm{fr}|\mathrm{vr})}(P)$, there exist $N \geq 1$ and $Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}$ such that

$$\left((R_{t,S})_{(t,S)\in\mathcal{S}}, (D_{t,S,r})_{(t,S,r)\in\mathcal{T}}\right) \geq \frac{1}{N}\left(\left(R_{t,S}(P,Q^N)\right)_{(t,S)\in\mathcal{S}}, (D_{t,S,r}(P,Q^N))_{(t,S,r)\in\mathcal{T}}\right) - (\epsilon,\ldots,\epsilon).$$

Thus, there exist a dimension N and a quantizer $Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}$ such that

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \geq \frac{1}{N}\sum_{(t,S)\in\mathcal{S}}\left[a_{t,S}R_{t,S}(P,Q^N) + \sum_{r\in S} b_{t,S,r}D_{t,S,r}(P,Q^N)\right] - \delta$$

where $\delta \to 0$ as $\epsilon \to 0$. Since ϵ is arbitrary, and the above property holds for a particular $N \geq 1$ and $Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}$

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \geq \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b}).$$

On the other hand, for any $\epsilon > 0$, there exist a dimension $N \geq 1$ and a code $Q^N \in \mathcal{Q}^{(\mathrm{fr}|\mathrm{vr}),N}$ such that

$$\frac{1}{N}\sum_{(t,S)\in\mathcal{S}}\left[a_{t,S}R_{t,S}(P,Q^N) + \sum_{r\in S} b_{t,S,r}D_{t,S,r}(P,Q^N)\right] \leq \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b}) + \epsilon.$$

Thus,

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \leq \inf_n j^{(\mathrm{fr}|\mathrm{vr}),n}(P,\boldsymbol{a},\boldsymbol{b}) + \epsilon$$

TABLE III
DATA SOURCE ASSIGNMENTS FOR THE EXPERIMENTS

Satellite Name	Frequency Band	WZ	2AWZ	2A	Three-Node	MD	BC
GMS-5	Visible				$X_{1,\{2,3\}}$		
GMS-5	Infrared 1				$X_{1,2}$		
GMS-5	Infrared 2				$X_{1,3}$		
GOES-8	Visible	Z_1	Z_1	Z_1	$X_{2,\{1,3\}}$	X	$X_{1,\{2,3\}}$
GOES-8	Infrared 2	$X_{2,1}$	$X_{2,1}$	$X_{2,1}$	$X_{2,3}$		$X_{1,2}$
GOES-8	Infrared 5		$X_{3,1}$		$X_{2,1}$		$X_{1,3}$
GOES-10	Visible				$X_{3,\{1,2\}}$		
GOES-10	Infrared 2				$X_{3,1}$		
GOES-10	Infrared 5				$X_{3,2}$		

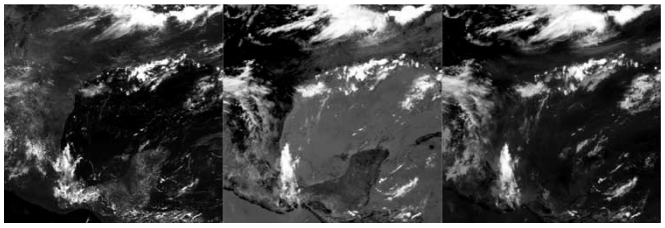


Fig. 14. Sample images from the GOES-8 weather satellite. From left to right: visible spectrum, infrared 2, infrared 5.

since

$$j^{(\mathrm{fr}|\mathrm{vr})}(P,\boldsymbol{a},\boldsymbol{b}) \leq \frac{1}{N}\sum_{(t,S)\in\mathcal{S}}\left[a_{t,S}R_{t,S}(P,Q^N)+\sum_{r\in S}b_{t,S,r}D_{t,S,r}(P,Q^N)\right].$$

Once again, since ϵ is arbitrary the desired result follows. □

APPENDIX II
SATELLITE WEATHER IMAGE DATA SET

The satellite weather data set, obtained courtesy of NASA and the University of Hawaii, contains images from three geosynchronous weather satellites. Each satellite records 8-bit greyscale images in frequency bands ranging from infrared to the visible spectrum. For each satellite, we use images from three bands. Each image is cropped to 512×512 pixels. Table III shows the assignment of satellite images to data sources for the WZ, 2AWZ, 2A, three-node, MD, and BC system experiments. Fig. 14 shows sample images. The training and testing sets are nonoverlapping and consist of eight and four images per source, respectively.

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[2] V. Koshelev, "Hierarchical coding of discrete sources," *Probl. Pered. Inform.*, vol. 17, no. 3, pp. 20–33, 1981.

[3] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. 37, pp. 269–275, Mar. 1991.

[4] R. M. Gray and A. D. Wyner, "Source coding for a simple network," *Bell Syst. Tech. J.*, vol. 53, no. 9, pp. 1681–1721, Nov. 1974.

[5] Q. Zhao and M. Effros, "Broadcast system source codes: A new paradigm for data compression," in *Conf. Rec. 33rd Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 1999, pp. 337–341.

[6] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.

[7] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 1–10, Jan. 1976.

[8] A. D. Wyner, "The rate-distortion function for source coding with side information at the decoder-II: General sources," *Inform. Contr.*, vol. 38, pp. 60–80, 1978.

[9] J. K. Wolf, A. D. Wyner, and J. Ziv, "Source coding for multiple descriptions," *Bell Syst. Tech. J.*, vol. 59, pp. 1417–1426, Oct. 1980.

[10] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, pp. 446–472, Dec. 1980.

[11] A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 851–857, Nov. 1982.

[12] H. Jafarkhani and N. Farvardin, "Channel-matched hierarchical table-lookup vector quantization," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1121–1125, May 2000.

[13] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[14] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.

[15] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 562–574, Oct. 1980.

[16] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Processing*, vol. 39, pp. 2500–2507, Nov. 1991.

[17] H. Brunk and N. Farvardin, "Fixed-rate successively refinable scalar quantizers," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Apr. 1996, pp. 250–259.

[18] H. Jafarkhani, H. Brunk, and N. Farvardin, "Entropy-constrained successively refinable scalar quantization," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1997, pp. 337–346.

[19] M. Effros, "Practical multi-resolution source coding: TSVQ revisited," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 53–62.

[20] M. Effros and D. Dugatkin, "Multiresolution vector quantization," *IEEE Trans. Inform. Theory*, submitted for publication.

[21] V. A. Vaishampayan, "Vector quantizer design for diversity systems," in *Proc. 25th Annu. Conf. Information Sciences and Systems*, Mar. 1991, pp. 564–569.

[22] ——, "Design of multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 39, pp. 821–834, May 1993.

[23] V. A. Vaishampayan and J. Domaszewicz, "Design of entropy-constrained multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 40, pp. 245–250, Jan. 1994.

[24] M. Effros and L. Schulman, "Rapid near-optimal VQ design with a deterministic data net," in *Proc. IEEE Int. Symp. Inform. Theory*, Chicago, IL, June 2004, p. 298.

[25] M. Fleming and M. Effros, "Generalized multiple description vector quantization," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 3–12.

[26] Q. Zhao and M. Effros, "Lossless and lossy broadcast system source codes: Theoretical limits, optimal design, and empirical performance," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 63–72.

[27] M. Effros, "Network source coding," in *Proc. 2000 Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2000.

[28] M. Fleming and M. Effros, "Network vector quantization," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2001, pp. 13–22.

[29] T. J. Flynn and R. M. Gray, "Encoding of correlated observations," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 773–787, Nov. 1987.

[30] A. D. Wyner, "Recent results in Shannon theory," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 2–10, Jan. 1974.

[31] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1250–76, June 2002.

[32] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academics, 1992.

[33] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Necessary conditions for the optimality of variable-rate residual vector quantizers," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1903–14, Nov. 1995.

[34] H. S. Witsenhausen, "The zero-error side information problem and chromatic numbers," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 592–593, Sept. 1976.

[35] N. Alon and A. Orlitsky, "Source coding and graph entropies," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1329–1339, Sept. 1996.

[36] J. Körner and A. Orlitsky, "Zero-error information theory," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2207–2228, Oct. 1998.

[37] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 158–167.

[38] Y. Yan and T. Berger, "On instantaneous codes for zero-error coding of two correlated sources," in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, June 2000, p. 344.

[39] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose, "Ony zero-error coding of correlated sources," *IEEE Trans. Information Theory*, vol. 49, pp. 2856–2873, Nov. 2003.

[40] Q. Zhao and M. Effros, "Lossless and near-lossless source coding for multiple access networks," *IEEE Trans. Inform. Theory*, vol. 49, pp. 112–128, Jan. 2003.

[41] ——, "Optimal code design for lossless and near-lossless source coding in multiple access networks," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2001, pp. 263–272.

[42] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[43] T. M. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 226–228, Mar. 1975.

[44] Q. Zhao and M. Effros, "Low complexity code design for lossless and near lossless side information source codes," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2003, pp. 3–12.

[45] K. Popat and R. W. Picard, "Cluster-based probability model and its application to image and texture processing," *IEEE Trans. Image Processing*, vol. 6, pp. 268–284, Feb. 1997.

[46] K. Zeger and A. Gersho, "Globally optimal vector quantization design by stochastic relaxation," *IEEE Trans. Signal Processing*, vol. 40, pp. 310–22, Feb. 1992.

[47] K. Rose, E. Gurewitz, and G. C. Fox, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1249–57, July 1992.

[48] R. Ahlswede and J. Körner, "Source coding with side information and a converse for degraded broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 629–637, Nov. 1975.