Journal of Network and Systems Management (© 2006)
DOI: 10.1007/s10922-005-9019-0

Edge-Cut Bounds on Network Coding Rates

Gerhard Kramer[1,3] and Serap A. Savari[2]

Active networks are network architectures with processors that are capable of executing code carried by the packets passing through them. A critical network management concern is the optimization of such networks and tight bounds on their performance serve as useful design benchmarks. A new bound on communication rates is developed that applies to network coding, which is a promising active network application that has processors transmit packets that are general functions, for example a bit-wise XOR, of selected received packets. The bound generalizes an edge-cut bound on routing rates by progressively removing edges from the network graph and checking whether certain strengthened *d*-separation conditions are satisfied. The bound improves on the cut-set bound and its efficacy is demonstrated by showing that routing is rate-optimal for some commonly cited examples in the networking literature.

KEY WORDS: Network capacity; network coding; active networks; *d*-separation.

1. INTRODUCTION

In recent years there has been considerable interest in technologies known as *active networks* [1] that permit network nodes to execute computations specific to the packets passing through them. The programmability of infrastructure is the key innovation in this approach to network architecture; the added flexibility provides a means to implement novel transmission techniques to improve performance. A small subset of the literature on active networks can be found in [1–6].

The optimization of active networks is a critical network management concern. Network optimization has traditionally studied communication networks in the same framework as other types of networks such as those arising in

[1]Mathematics of Communications Research Department, Bell Labs, Lucent Technologies, New Jersey.
[2]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan.
[3] To whom correspondence should be addressed at Mathematics of Communications Research Department, Bell Labs, Lucent Technologies, 600 Mountain Ave, Murray Hill, New Jersey; Email: gkr@research.bell-labs.com.

transportation and manufacturing problems (see, e.g., [7, p. 1]). A few years ago, a groundbreaking paper [8] pointed out that this paradigm imposes artificial restrictions on the workings of processors in *communication* networks. For example, consider the problem shown in Fig. 1. There are two source-destination pairs (s_1, t_1) and (s_2, t_2), and each of the directed edges has unit capacity. Each source seeks to send a unit-rate information stream to its destination. This is impossible in the traditional *routing* regime where intermediate relays can only forward the information received. However, with *network coding* [8] a processor that receives information can transmit a different *function* of this information on each of its outgoing edges. Fig. 1 illustrates that an appropriate choice of functions makes the desired rates feasible (the "$x + y$" in Fig. 1 is the XOR of the bits x and y). In fact, both destination terminals can decode both messages.

Network coding has become an intensely studied interdisciplinary subfield of information theory since the publication of [8]. Recent work has exploited ideas and techniques from many areas including randomized algorithms, algebraic coding theory, matrix theory, and graph theory. An updated web page [9] lists many publications in the area.

There are several approaches to implementing network codes. One approach designs fixed coding functions for each processor based on a centralized knowledge of the network topology. A second approach (see, e.g., [10–12]) is motivated by issues arising in distributed or dynamic scenarios where centralized control is impractical. This approach has each vertex transmit on its outgoing edges a randomly chosen linear combination of the information from its incoming edges. For decoding, one requires packets to have headers that inform the destinations of which linear combinations were chosen to form the packets. Each header is modified dynamically as the packets flow through, and are combined by, the

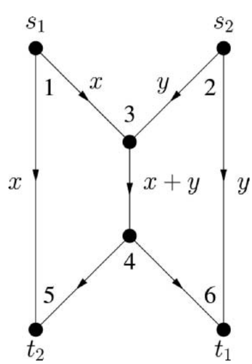

Fig. 1. A two-commodity problem on a directed graph.

network vertices. For example, to keep the header size limited, relaying vertices might wish to decode packets and then re-encode them based on local information. The headers can thus be considered as executable code carried by the packets. Random network coding is therefore a promising active network application since it demonstrates the potential payoff in sending executable code rather than just data.

The headers in the above-mentioned example change the function of network vertices at a fast timescale. An example of an application that reconfigures the network at a slower time scale is when a centralized management system multicasts commands that tune the network, or perhaps even the network codes themselves. In the former case, one would network code the executable code itself. In the latter case, an active packet might contain executable code that installs network codes along certain paths or subgraphs based upon conditions that this active packet meets along its route. One can consider this to be an autonomously installed or temporary network-coded overlay network.

Another consideration in network communication is that the information sent from one processor to another might be lost, e.g., due to congestion, or is corrupted by errors. Active networks offer each processor the opportunity to change its coding functions depending on the error statistics. Since there is a tradeoff between network coding rate and reliability, actively managing network coding rates according to this tradeoff should prove useful.

Yet another application is a new approach to network management for protection from and recovery of link failures (see, e.g., [13, 14]). Here the network is modeled as a finite-state machine where the operation of a processor is affected by management signals that indicate the current link failures and/or directions for recovery behavior. Among the contributions of [13] are bounds on management requirements for several network connection problems.

A further benefit of network coding has been to improve the allocation of physical and medium-access layer resources in wireless ad hoc networks [15, 16]. For example, suppose one is given a collection of end-to-end communication demands and an objective of minimizing power consumption. It was demonstrated that network codes increase the energy efficiency over traditional routing for a particular cross-layer optimization over the physical layer, network layer, and link layer.

1.1 Bounds on Network Coding Rates

The aim of this paper is to develop theoretical bounds on the communication rates that can be attained using network coding. We thereby also determine bounds on the performance of active networks. The value of the theory is, e.g., to help determine how well tuned an active or coded network really is.

Our problem can be considered to be a generalization of the classical problem of bounding the maximal *flow* from one vertex to another in a graph subject to capacity limitations on arcs or edges. Fifty years ago, L. R. Ford, D. R. Fulkerson, and other individuals discovered the celebrated "max-flow min-cut" theorem that states that the maximal flow is the minimum capacity among all edge cuts separating the source and destination vertices [17–20]. A related bound additionally partitions the vertex set into two disjoint sets, and in [21] we developed this latter type of bound for network coding. However, as pointed out in [7, pp. 16–17], sometimes tighter bounds can be found by considering `disconnecting edge sets.` In this paper, we present an information-theoretic counterpart to this latter type of bound. We do this by borrowing from the artificial intelligence literature [22] the concept of *d-separation* in Bayesian networks.

Bayesian networks are graphs whose vertices represent random variables, and d-separation is a graphical procedure that establishes the conditional statistical independence of certain sets of these random variables. We will here consider special types of Bayesian networks known as *functional dependence graphs* (FDGs) and we use a strengthened version of an extension of *d*-separation called *fd-separation* that appeared in [23, ch. 2].

2. PRELIMINARIES

Consider an undirected, edge-capacitated graph $\mathcal{N} = (\mathcal{V}, \varepsilon)$ with vertex and edge sets

$$\mathcal{V} = \{1, 2, \ldots, V\} \tag{2.1}$$

$$\varepsilon = \{(u_1, v_1), (u_2, v_2), \ldots, (u_E, v_E)\} \tag{2.2}$$

respectively, where $u_e, v_e \in \mathcal{V}$ for $e = 1, 2, \ldots, E$, and where C_e is the capacity of edge e. Consider further a subset $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ of $\mathcal{V}$ called *terminals*, some of which are *sources* and some of which are *sinks*. An *edge cut* is a set *hcal* ε_d of edges that disconnects sources from sinks. (Edge cuts in directed graphs are sometimes called *directed cuts* or *disconnecting edge sets*.) Rather intuitively, the sum of the routing rates of the source-destination pairs that are disconnected by ε_d is upper bounded by the sum of the capacities of the edges in ε_d.

We would like to apply edge-cut bounds to network coding. Such bounds clearly apply to *undirected* graphs and one can prove this by using the techniques of [21]. Unfortunately, a standard example shows that edge-cut bounds do not necessarily apply to *directed* graphs. Consider the network with unit-capacity edges shown in Fig. 1. There are two source-destination pairs $(s_1,\ t_1)$ and $(s_2,\ t_2)$, and we write their respective rates as R_1 and R_2. The set $\varepsilon_d = \{(3, 4)\}$ is an edge cut for both sources so the edge-cut bound states that $R_1 + R_2 \leq 1$ with routing. However, network coding achieves $(R_1, R_2) = (1,\ 1)$ by forming the XOR of the bits x and y on the respective edges (1, 3) and (2, 3), and sending the result down edges

(3, 4), (4, 5), and (4, 6). This example shows that one cannot always rely on edge-cut bounds when using network codes. The main purpose of this paper is to develop an alternative to edge-cut bounds that does apply to network coding. We further use the bound to derive new capacity theorems for network information flow.

2.1 Information Theory

For our analysis, we assume that the reader is familiar with concepts of information theory (see [24, ch. 2]). We write $H(X)$, $H(XY)$, and $H(X|Y)$ for the respective entropy of the random variable X, the joint entropy of the random variables X and Y, and the entropy of X conditioned on Y. We further write $I(X; Y)$ and $I(X; Y|Z)$ for the respective mutual information between X and Y, and the mutual information between X and Y when conditioned on the random variable Z.

We write $P_{XY|Z}(x, y|z)$ for the probability that $X = x$ and $Y = y$ when the event $Z = z$ occurs, assuming that $P_Z(z) > 0$. As usual, for discrete random variables we say that X and Y are statistically independent when conditioned on Z if

$$P_{XY|Z}(x, y|z) = P_{X|Z}(x|z) \cdot P_{Y|Z}(Y|z) \tag{2.3}$$

for all x, y, and z with $P_Z(z) > 0$. Alternatively, we say that $X - Z - Y$ forms a Markov chain. We remark that $X - Z - Y$ forms a Markov chain if and only if

$$I(X; Y|Z) = 0. \tag{2.4}$$

3. NETWORK MODEL

We adopt the model of [21, 25] whose components and rules we list for completeness below (see also [26, section III. A–B]). Most of what follows applies to real networks, perhaps with the exception of the *clocking* described in the first bullet. We remark that this assumption can often be relaxed; its main purpose is to ensure that the network vertices behave in a *causal* fashion. The clocking assumption is further useful to keep track of the bits and symbols being transmitted around the network.

- The network is *clocked*, i.e., a universal clock ticks N times.
- Vertex u transmits a symbol $X_{uv}^{(n)}$, $(u, v) \in \varepsilon$, *after* clock tick $n - 1$ and *before* clock tick n for $n = 1, 2, \ldots, N$.
- Vertex v receives symbols $Y_{uv}^{(n)}$, $(u, v) \in \varepsilon$, *at* clock tick n. Note that there is a small delay between transmission and reception that ensures the network operates in a *causal* fashion. The output $X_{uv}^{(n)}$ is in general a noisy function of the channel input $X_{uv}^{(n)}$, i.e., for all $(u, v) \in \varepsilon$ and all n we have

$$Y_{uv}^{(n)} = f_{uv}\left(X_{uv}^{(n)}, Z_{uv}^{(n)}\right) \tag{3.1}$$

for some function $f_{uv}(\cdot)$, where $Z_{uv}^{(n)}$ is a noise random variable that is statistically independent of all other noise and message random variables. For simplicity we will often model the edge channels as being noise-free, i.e., we will mostly consider channels with $Y_{uv}^{(n)} = X_{uv}^{(n)}$ for all u and v. However, our results do extend to noisy channels. We demonstrate this by an example below.

- There are K independent messages W_k, $k = 1, 2, \ldots, K$, in the network. One might think of the W_k as being "commodities.'' For the *multicommodity flow* problem [27, p. 1221], message W_k is associated with a vertex pair (s_k, t_k), $s_k \neq t_k$, and one wishes to transmit NR_k units of data from s_k to t_k simultaneously for all k. The meaning is that s_k is the *source* vertex and t_k is the *sink* or destination vertex. In communications, R_k refers to the *rate* of message k.
- A more general problem is the multimessage multicasting problem, where several destinations decode each message W_k. We write D_k for the number of destinations decoding W_k. More precisely, message W_k is associated with the vertices $(s_k, t_k(1), t_k(2), \ldots, t_k(D_k))$ and one wishes to transmit R_k units of data from s_k to $t_k(i)$, $s_k \neq t_k(i)$, simultaneously for all $k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, D_k$.
- Let $\underline{W}_u$ be the set of messages originating at vertex u. The input $X_{uv}^{(n)}$ is a function of $\underline{W}_u$ and vertex u's past channel outputs

$$\underline{Y}_u^{n-1} = \underline{Y}_u^{(1)}, \underline{Y}_u^{(2)}, \ldots, \underline{Y}_u^{(n-1)}. \tag{3.2}$$

Note that $\underline{Y}_u^{(n)}$ is a vector that includes the nth channel outputs from *all* edges incident to u. Note also that $X_{uv}^{(n)}$ is *any* function of $\underline{W}_u$ and $\underline{Y}_u^{n-1}$, so that we are permitting *joint* channel coding, routing, and/or network coding. We distinguish between routing and network coding in that routing permits message symbols and arriving *packets* (groups of input or output symbols) to be stored, reordered, and collected into other packets. Network coding, however, additionally allows packets to be *combined* to create new packets.

- Suppose W_k is destined for vertex $t_k(i)$. After transmission is completed, vertex $t_k(i)$ puts out its estimate $\hat{W}_k^{(i)}$ of W_k. Note that $\hat{W}_k^{(i)}$ is a function of vertex $t_k(i)$'s messages $\underline{W}_{t_k(i)}$ and its channel outputs $\underline{Y}_{t_k(i)}^N$.
- A rate-tuple $(R_1, R_2, \ldots, R_K)$ is said to be *achievable* if there exist encoders and decoders such that

$$\Pr\left(\bigcup_{k,i} \left\{\hat{W}_k^{(i)} \neq W_k\right\}\right) < \varepsilon \tag{3.3}$$

for any positive ε. The *capacity region* C is the closure of the set of achievable rate-tuples.

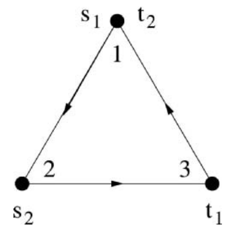

Fig. 2. A two-commodity problem on a directed graph.

4. FUNCTIONAL DEPENDENCE GRAPHS, *d*-SEPARATION, AND CONDITIONAL INDEPENDENCE

We will use the calculus of *d*-separation and *fd*-separation in FDGs. FDGs are graphs where the vertices represent random variables and the edges represent the functional dependencies between the random variables [23, 26]. For instance, suppose we have N_{RV} random variables that are defined by S_{RV} independent (or source) random variables by N_{RV} functions. An FDG $\mathcal{G}$ is a directed graph having $N_{\mathrm{RV}} + S_{\mathrm{RV}}$ vertices representing the random variables and in which edges are drawn from one vertex to another if the random variable of the former vertex is an argument of the function defining the random variable of the latter vertex.

For example, suppose we have the two-commodity problem in a noise-free triangular network depicted in Fig. 2. A corresponding FDG is shown in Fig. 3. In this graph, X_{12}^N is a function of the message W_1 and X_{31}^N (in fact, $X_{12}^{(n)}$ is a function of W_1 and the *past* X_{31}^{n-1} only). The message estimate $\hat{W}_2$ of W_2 at vertex 1 is also a function of W_1 and X_{31}^N. The channel inputs X_{23}^N are a function of W_2 and X_{12}^N, and the estimate $\hat{W}_1$ is a function of X_{23}^N. The $S_{RV} = 2$ vertices representing the independent W_1 and W_2 are distinguished by drawing them with a hollow circle. Note that Fig. 3 is the line graph of Fig. 2 with the addition of vertices representing the messages and their estimates, and edges representing the functional relations of these new vertices to the existing ones.

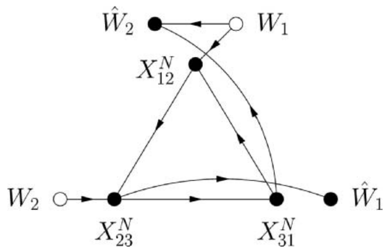

Fig. 3. FDG for the two-commodity problem in Fig. 2.

By d-separation we mean the following reformulation of a definition in [22, p. 117] that is described in [23, 26].

Definition 1. Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be disjoint subsets of the vertices of a FDG $\mathcal{G}$. $\mathcal{Z}$ is said to d-separate $\mathcal{X}$ from $\mathcal{Y}$ if there is no path between a vertex in $\mathcal{X}$ and a vertex in $\mathcal{Y}$ after the following manipulations of the graph have been performed.

1. Consider the subgraph $\mathcal{G}_{\mathcal{XYZ}}$ of $\mathcal{G}$ consisting of the vertices in $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$, as well as the edges and vertices encountered when moving *backward* one or more edges starting from any of the vertices in $\mathcal{X}$ or $\mathcal{Y}$ or $\mathcal{Z}$.
2. In $\mathcal{G}_{\mathcal{XYZ}}$ delete all edges coming *out* of the vertices in $\mathcal{Z}$. Call the resulting graph $\mathcal{G}_{\mathcal{XYZ}}$.
3. Remove the arrows on the remaining edges of $\mathcal{G}_{\mathcal{XYZ}}$ to obtain an undirected graph.

A fundamental result of [22, section 3.3] is that d-separation establishes conditional independence in FDGs having no directed cycles. That is, if $\mathcal{Z}$ d-separates $\mathcal{X}$ from $\mathcal{Y}$ in $\mathcal{G}$ and we collect the random variables of the vertices in $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ in the respective vectors $\underline{X}$, $\underline{Y}$ and $\underline{Z}$, then $\underline{X}$-$\underline{Z}$-$\underline{Y}$ forms a Markov chain.

A simple extension of d-separation is known as *fd*-separation which uses the fact that the FDG represents *functional* relations, and not only Markov relations as in Bayesian networks (see [23, ch. 2]). For *fd*-separation, after the second step above one successively removes all edges coming out of vertices without incoming edges, excepting the source (or message) vertices. One can, in fact, also successively remove all edges on cycles without incoming edges, and we shall refer to this strengthened version of the definition in [23, p. 15] as *fd*-separation. We remark that *fd*-separation applies to FDGs with cycles, as long as all subgraphs of the FDGs are also FDGs (this result follows directly from [23, ch. 2] and will be proved in a future paper).

5. P*d*E BOUND FOR NETWORK CODING

The bound we develop begins with a set of edges ε_{d} like the edge-cut bound. However, in addition to computing the sum of the capacities of these edges, we must perform a series of verification steps. Consider a set $\mathcal{S}_d$ of source indices and an ordering of these indices via a one-to-one mapping $\pi(\cdot)$ from $\{1, 2,\ldots,|\mathcal{S}_d|\}$ to $\mathcal{S}_d$, where $|\mathcal{S}_d|$ is the cardinality of $\mathcal{S}_d$. The reason for introducing this ordering will become clear when we consider some examples below.

We use the notation $X_{\varepsilon_d} = \{X_{uv} : (u, v) \in \varepsilon_d\}$ and similarly for Y_{ε_d} and Z_{ε_d}. The following steps describe our bound for noise-free networks. Let $X^N_{\varepsilon_d}$ be the channel inputs of the edges ε_d, $W_{\mathcal{S}_d}$ be the messages with indices in $\mathcal{S}_d$, $\mathcal{S}_d \subseteq \{1, 2, \ldots, K\}$, and $\mathcal{S}^C_d$ be the complement of $\mathcal{S}_d$ in $\{1,2,\ldots,K\}$.

1) (Initialization) Consider the FDG $\mathcal{G}$ corresponding to the network graph $\mathcal{N}$, i.e., the line graph of $\mathcal{N}$ with the addition of vertices and edges representing the messages and their estimates (see, e.g., Figs. 2 and 3).

- Remove all vertices and edges in $\mathcal{G}$ except those encountered when moving backward one or more edges starting from any of the vertices representing: (1) $X^N_{\varepsilon_d}$, (2) any choice of non-empty subset of $\{\hat{W}^{(i)}_k : i = 1, 2, \ldots, D_k\}$ for all $k \in \mathcal{S}_d$ and (3) all messages W_k, $k = 1, 2, \ldots, K$.
- Further remove the edges coming out of the vertices representing $X^N_{\varepsilon_d}$ and $W_{\mathcal{S}^c_d}$, and successively remove edges coming out of vertices and on cycles that have no incoming edges, excepting source vertices. Call the resulting graph $\mathcal{G}_{\varepsilon_d}$. Set $k = 1$.

2) (Iterations) If $W_{\pi(k)}$ is not disconnected (in an undirected sense) from one of its estimates $\hat{W}^{(i)}_{\pi(k)}$, $i = 1, 2, \ldots D_k$, then stop (one has no bound). If $W_{\pi(k)}$ is disconnected (in an undirected sense) from all of its estimates then:

- Remove the edges coming out of the vertex representing $W_{\pi(k)}$.
- Successively remove edges coming out of vertices and on cycles that have no incoming edges, excepting source vertices. Call the resulting graph $\mathcal{G}_{\varepsilon_d} W^k_\pi$.

3) (Termination and Bound) Increment k. If $k \le K$ go to the previous step. If $k = K + 1$, then we have

$$\sum_{k \in \mathcal{S}_d} R_k \le \sum_{e \in \varepsilon_d} C_e. \tag{5.1}$$

We call this bound a *progressive d-separating edge-set* bound, or PdE bound for short (one might also refer to it as a PdE *algorithm*). The word "progressive" describes the step-by-step removal of edges from $\mathcal{G}$. The term "*d*-separation" describes the use of *fd*-separation in steps 1 and 2 above. We remark that the PdE bound includes as special cases those bounds based on edge cuts that partition $\mathcal{V}$ into two disjoint sets ([21],[24, section 14.10]).

Example 1. Consider the network of Fig. 1. We choose $\varepsilon_d = \{(3, 4)\}$ and $\mathcal{S}_d = \{1, 2\}$, and the resulting graph $\mathcal{G}_{\varepsilon_d}$ is shown in Fig. 4. We choose $\pi(.)$ to be the identity mapping, i.e., we choose the ordering W_1, W_2. For $k = 1$, we must check if W_1 is disconnected from $\hat{W}_1$ in an undirected sense. However, there is an undirected path from W_1 to $\hat{W}_1$ so we must stop without a bound. A similar conclusion to the procedure occurs if we choose the ordering W_2, W_1. Thus, as required, we cannot claim that $R_1 + R_2 \le 1$.

Example 2. Consider the network of Fig. 2 for which $\mathcal{G}$ is the graph in Fig. 3. Suppose that $C_e = 1$ for all e. We choose $\varepsilon_d = \{(2, 3)\}$, $\mathcal{S}_d = \{1, 2\}$, and

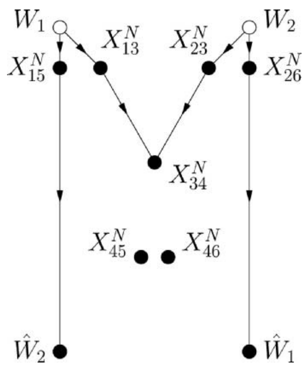

Fig. 4. Modified FDG for the two-commodity problem in Fig. 1.

the resulting graph $\mathcal{G}_{\varepsilon_d}$ is shown in Fig. 5. We next choose $\pi(.)$ to be the identity mapping. For $k = 1$, we must check if W_1 is disconnected from $\hat{W}_1$ which is indeed the case. The next graph $\mathcal{G}_{\varepsilon_{d_{W_\pi^1}}}$ has only one edge and W_2 is disconnected from $\hat{W}_2$. We thus have the desired bound $R_1 + R_2$ (this type of edge-cut bound first appeared in [21]).

6. NOISY CHANNELS

The above procedure extends to noisy channels by including the $Y_{uv}^{(n)} = f_{uv}(X_{uv}^{(n)}, Z_{uv}^{(n)})$ in the FDGs. One further replaces $X_{\varepsilon_d}^N$ by $\{Y_{\varepsilon_d}^N, Z_{\varepsilon_d^C}^N\}$ in the first step in Section 5,where ε_d^C is the complement of ε_d in ε. The value $\mathrm{C_e}$ in (5.1) is now the capacity of the channel of edge e. For example, consider the FDG in Fig. 6 that is a noisy version of the FDG in Fig. 3. The noise random variables Z_{uv}^N are represented as open circles inside the triangle formed by the cycle

$$X_{12}^N \to Y_{12}^N \to X_{23}^N \to Y_{23}^N \to X_{31}^N \to Y_{31}^N \to X_{12}^N.$$

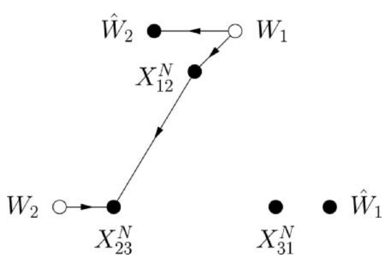

Fig. 5. Modified FDG for the two-commodity problem in Fig. 2.

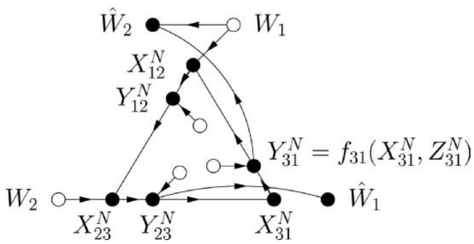

Fig. 6. FDG for the problem in Fig. 2 when the channels are noisy.

The procedure described in Section 5 (with $X_{\varepsilon_d}^N$ replaced by $\{Y_{\varepsilon_d}^N,\ Z_{\varepsilon_d^C}^N\}$) will give graphs like those in Fig. 5.

7. UNDIRECTED GRAPHS

The above procedure extends to undirected graphs with a few extra steps. The main addition is that one replaces every undirected edge $e = (u, v)$ with capacity C_e by a pair of oppositely directed edges labeled by the entropies $C_{uv} := H(X_{uv}^N)/N$ and $C_{vu} := H(X_{vu}^N)/N$. One then requires that $C_{uv} + C_{vu} \leq C_e$. We remark that it is often more convenient to draw only the bidirected version of the undirected graph without formally converting it into a line graph.

Example 3. Consider the network of Fig. 7 that appeared in a paper by Hu [28]. This network served as an example to show that the vertex-partitioning cut-set bound can be loose for three commodities. We construct the bidirected graph shown in Fig. 8, where the edge from vertex u to vertex v represents X_{uv}^N (we have labeled only some of the edges to avoid cluttering the figure with notation). One can construct the FDG line graph directly from this graph.

Suppose that the undirected edges have capacity two. Hu showed that the vertex-partitioning cut-set bound permits the rate triple $(R_1,\ R_2,\ R_3) = (4,\ 2,\ 1)$

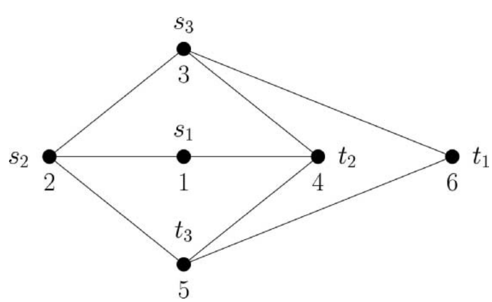

Fig. 7. Hu's three-commodity problem.

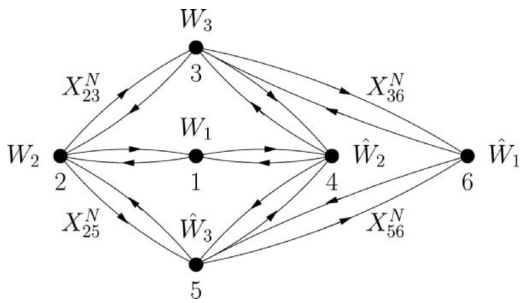

Fig. 8. Bidirected graph for the three-commodity problem in Fig. 7.

but routing requires $R_3 = 0$ when $(R_1, R_2) = (4, 2)$. We wish to determine if the same is true with network coding. We choose $\varepsilon_d = \{(3, 6), (5, 6)\}$ and $\mathcal{S}_d = \{1\}$ from which we obtain $R_1 \leq C_{36} + C_{56} \leq 4$, with equality only if $C_{63} = C_{65} = 0$. Similarly, with $\varepsilon_d = \{(1, 2), (1, 4)\}$ and $\mathcal{S}_d = \{1\}$ we require $C_{21} = C_{41} = 0$ for $R_1 = 4$. Combining these results, we can restrict attention to the graph in Fig. 9.

For Fig. 9, we choose $\varepsilon_d = \{(2, 3), (4, 3), (2, 5), (4, 5)\}$, $\mathcal{S}_d = \{1, 2, 3\}$, and $[\pi(1), \pi(2), \pi(3)] = [3, 1, 2]$. The resulting graph $\mathcal{G}_{\varepsilon_d}$ is shown in Fig. 10. We find that

$$R_1 + R_2 + R_3 \leq C_{23} + C_{43} + C_{25} + C_{45}. \tag{7.1}$$

Next, in Fig. 9 we choose $\varepsilon_d = \{(3, 2), (3, 4), (5, 2), (5, 4)\}$, $\mathcal{S}_d = \{2, 3\}$, and $[\pi(1), \pi(2)] = [2, 3]$. We find that

$$R_2 + R_3 \leq C_{32} + C_{34} + C_{52} + C_{54}. \tag{7.2}$$

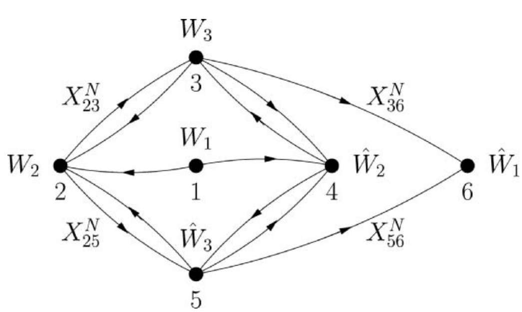

Fig. 9. Modified graph for the three-commodity problem in Fig. 7.

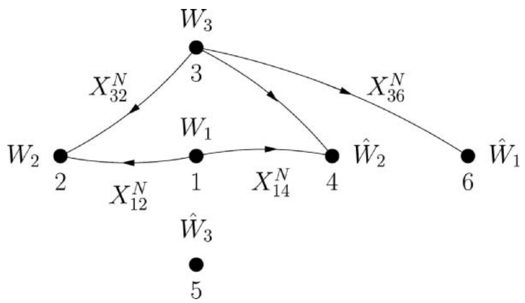

Fig. 10. Modified graph for the three-commodity problem in Fig. 7.

Combining (7.1) and (7.2), for $R_1 = 4$ we have

$$R_1 + 2(R_2 + R_3) \leq 8. \tag{7.3}$$

Thus, if $R_1 = 4$ and $R_2 = 2$ we require $R_3 = 0$ with or without network coding.

Example 4. Consider the network of Fig. 11 that appeared in a paper by Okamura and Seymour [29]. This network served as an example to show that the vertex-partitioning cut-set bound is not necessarily tight for routing on a planar graph where one cannot draw the graph so that all sources and sinks are on the boundary of the infinite region (note that s_3 and t_2 are not on the boundary of the infinite region in Fig. 11).

Suppose that the undirected edges have unit capacity. Okamura and Seymour showed that the vertex-partitioning cut-set bound permits the rate-tuple $(R_1, R_2, R_3, R_4) = (1, 1, 1, 1)$ but routing cannot achieve this set of rates [29]. In fact, the best symmetric rate with routing is $R_k = 3/4$ for $k = 1, 2, 3, 4$.

We bound the achievable network coding rates. We choose $\varepsilon_d = \{(2, 1), (3, 1), (4, 1), (2, 5), (3, 5), (4, 5)\}$, $\mathcal{S}_d = \{1, 2, 3, 4\}$, and $\pi(.)$ to be the

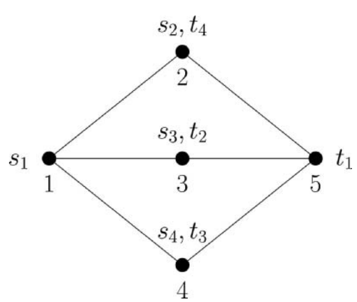

Fig. 11. Okamura and Seymour's four-commodity problem.

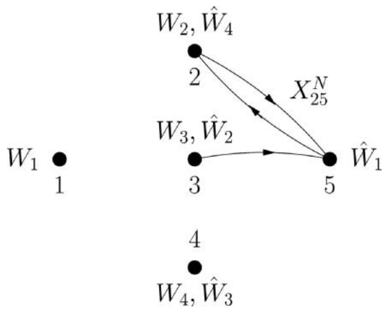

Fig. 12. Modified graph for the four-commodity problem in Fig. 11.

identity mapping. We find that

$$R_1 + R_2 + R_3 + R_4 \leq C_{21} + C_{31} + C_{41} + C_{25} + C_{35} + C_{45}. \tag{7.4}$$

We next choose $\varepsilon_d = \{(2,\ 1), (3,\ 1), (5,\ 3), (5,\ 4)\}$, $\mathcal{S}_d = \{2,\ 3\}$, and $[\pi(1), \pi(2)] = [3, 2]$. The resulting graph $\mathcal{G}_{\varepsilon_d}$ is shown in Fig. 12. We find that

$$R_2 + R_3 \leq C_{21} + C_{31} + C_{53} + C_{54}. \tag{7.5}$$

By symmetry, we similarly obtain

$$R_2 + R_4 \leq C_{21} + C_{41} + C_{52} + C_{53} \tag{7.6}$$

$$R_3 + R_4 \leq C_{31} + C_{41} + C_{52} + C_{54} \tag{7.7}$$

Combining (7.5)–(7.7), we have

$$R_2 + R_3 + R_4 \leq C_{21} + C_{31} + C_{41} + C_{52} + C_{53} + C_{54}. \tag{7.8}$$

We next choose $\varepsilon_d = \{(1,\ 2), (1,\ 3), (4,\ 5), (5,\ 3)\}$, $\mathcal{S}_d = \{1, 2, 4\}$, and $[\pi(1), \pi(2), \pi(3)] = [2,\ 1,\ 4]$. We find that

$$R_1 + R_2 + R_4 \leq C_{12} + C_{13} + C_{45} + C_{53}. \tag{7.9}$$

By symmetry, we similarly have

$$R_1 + R_2 + R_3 \leq C_{13} + C_{14} + C_{25} + C_{54} \tag{7.10}$$

$$R_1 + R_3 + R_4 \leq C_{12} + C_{14} + C_{35} + C_{52} \tag{7.11}$$

Combining (7.9)–(7.11), we have

$$3R_1 + 2(R_2 + R_3 + R_4) \leq 2(C_{12} + C_{13} + C_{14}) + 3. \tag{7.12}$$

Finally, we add the bounds (7.4), (7.8), and (7.12), and obtain

$$R_1 + R_2 + R_3 + R_4 \leq 3. \tag{7.13}$$

Thus, the best equal-rate point is at most three fourth with or without network coding. This result was pointed out to us by the authors of [30, 31] at a meeting on Network Coding in January 2005 [32]. We remark that the PdE bound developed here provides a different and widely applicable method of arriving at this result. For instance, the PdE bound additionally gives the sum-rate bound (7.13). Moreover, this bound can be combined with cut-set bounds to give the entire capacity *region* of the network in Fig. 11.

8. CONCLUDING REMARKS

Upper bounds on network coding rates are currently being developed by other groups [30, 31]. Some of the distinguishing features of our work are (see [33]): the PdE bound applies to general multimessage multicast, we have a formal procedure for generating rate bounds by using FDGs and *d*-separation (which makes a connection to the artificial intelligence literature), the progressive nature of our *fd*-separation bound strengthens an approach based on cutting edges only at the first step, and FDGs let us treat noisy networks as well as noise-free ones.

APPENDIX VALIDITY OF THE PDE BOUND

We prove the validity of the PdE bound described in Sections 5 and 6 for noisy as well as noise-free networks. This section assumes familiarity with advanced concepts in information theory. Recall that we consider the following objects.

- ε_d: a set of edges
- $\mathcal{S}_d$: a set of source indices
- $\pi(.)$: a one-to-one mapping from $\{1, 2, \ldots, |\mathcal{S}_d|\}$ to $\mathcal{S}_d$.
- a nonempty subset of $\{\hat{W}_k^{(i)} : i = 1, 2, \ldots, D_k\}$ for all $k \in \mathcal{S}_d$.

For the last item, recall that $\mathrm{W_k}$ is associated with the vertices $(s_k,\ t_k(1),\ t_k(2), \ldots,\ t_k(D_k))$, so we are considering some subset $\hat{\mathcal{V}}_k$ of the vertices $t_k(i)$, $i = 1, 2, \ldots, D_k$. We write the corresponding subset of estimates as $\hat{W}_k(\hat{\mathcal{V}}_k)$.

We continue by noting that, for reliable communication, Fano's inequality [24, p. 39] requires that

$$\begin{aligned}\sum_{k\in\mathcal{S}_d} R_k &\leq \sum_{k\in\mathcal{S}_d} \tfrac{1}{N} I(W_k; \hat{W}_k(\hat{\mathcal{V}}_k)) \\ &= \sum_{k=1}^{|\mathcal{S}_d|} \tfrac{1}{N} I\big(W_{\pi(k)}; \hat{W}_{\pi(k)}\big(\hat{\mathcal{V}}_{\pi(k)}\big)\big).\end{aligned} \tag{A.1}$$

We define $W_\pi^{k-1} = [W_{\pi(1)}, W_{\pi(2)}, \ldots, W_{\pi(k-1)}]$ and bound

$$\begin{aligned}
&I\big(W_{\pi(k)}; \hat{W}_{\pi(k)}\big(\hat{\mathcal{V}}_{\pi(k)}\big)\big) \\
&\overset{(a)}{\le} I\big(W_{\pi(k)}; \hat{W}_{\pi(k)}\big(\hat{\mathcal{V}}_{\pi(k)}\big) Y_{\varepsilon_d}^N Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C} W_\pi^{k-1}\big) \\
&\overset{(b)}{=} I\big(W_{\pi(k)}; \hat{W}_{\pi(k)}\big(\hat{\mathcal{V}}_{\pi(k)}\big) Y_{\varepsilon_d}^N | Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C} W_\pi^{k-1}\big) \\
&\overset{(c)}{=} I\big(W_{\pi(k)}; Y_{\varepsilon_d}^N | Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C} W_\pi^{k-1}\big)
\end{aligned} \tag{A.2}$$

where (a) follows because $I(A; B) \le I(A; BC)$, (b) follows because the messages and noise are statistically independent, and (c) follows by the chain rule for mutual information and because success in step 2) in Section 5 implies that

$$I\big(W_{\pi(k)}; \hat{W}_{\pi(k)}\big(\hat{\mathcal{V}}_{\pi(k)}\big) | Y_{\varepsilon_d}^N Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C} W_\pi^{k-1}\big) = 0 \tag{A.3}$$

via *fd*-separation. Inserting (A.2) into (A.1) and applying the chain rule for mutual information, we find that

$$\sum_{k \in \mathcal{S}_d} R_k \le \frac{1}{N} I\big(W_{\mathcal{S}_d}; Y_{\varepsilon_d}^N | Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C}\big). \tag{A.4}$$

We continue by upper bounding the mutual information expression in (A.4) by

$$\begin{aligned}
&I\big(W_{\mathcal{S}_d}; Y_{\varepsilon_d}^N | Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C}\big) \\
&\overset{(a)}{=} \sum_{n=1}^N I\big(W_{\mathcal{S}_d}; Y_{\varepsilon_d}^{(n)} | Y_{\varepsilon_d}^{n-1} Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C}\big) \\
&\le \sum_{n=1}^N I\big(W_{\mathcal{S}_d} X_{\varepsilon_d}^{(n)}; Y_{\varepsilon_d}^{(n)} | Y_{\varepsilon_d}^{n-1} Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C}\big) \\
&\overset{(b)}{=} \sum_{n=1}^N \Big[H\big(Y_{\varepsilon_d}^{(n)} | Y_{\varepsilon_d}^{n-1} Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C}\big) - H\big(Y_{\varepsilon_d}^{(n)} | X_{\varepsilon_d}^{(n)}\big)\Big] \\
&\overset{(c)}{=} \sum_{n=1}^N \Big[H\big(Y_{\varepsilon_d}^{(n)} | Y_{\varepsilon_d}^{n-1} Z_{\varepsilon_d^C}^N W_{\mathcal{S}_d^C}\big) - \sum_{e \in \varepsilon_d} H\big(Y_e^{(n)} | X_e^{(n)}\big)\Big] \\
&\text{(d)} \le \sum_{n=1}^N \sum_{e \in \varepsilon_d} \big[H\big(Y_e^{(n)} - H\big(Y_e^{(n)} | X_e^{(n)}\big)\big] \\
&\le \sum_{e \in \varepsilon_d} \sum_{n=1}^N \max_{P_{X_e^{(n)}}} I\big(X_e^{(n)}; Y_e^{(n)}\big) \\
&\overset{(e)}{=} \sum_{e \in \varepsilon_d} N \cdot C_e
\end{aligned} \tag{A.5}$$

where (a) follows by the chain rule for mutual information, (b) and (c) follow by (3.1), (d) follows because conditioning cannot increase entropy, and (e) follows because it is known that (see [24], Ch. 8)

$$C_e = \max_{P_{X_e^{(n)}}} I\left(X_e^{(n)}; Y_e^{(n)}\right). \tag{A.6}$$

Inserting (A.5) into (A.4) gives (5.1).

ACKNOWLEDGMENTS

The authors wish to thank C. Chekuri for stimulating discussions and S. Bush for helpful comments and suggestions. The work of G. Kramer was partially supported by the Board of Trustees of the University of Illinois Subaward No. 04-217 under NSF Grant No. CCR-0325673. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the University or its Project Director or of the National Science Foundation. The work of S. A. Savari was supported by NSF Grant No. CCF-0430201.

REFERENCES

1. S. F. Bush and A. B. Kulkarni, *Active Networks and Active Network Management,* Kluwer Academic, New York, 2001.
2. S. F. Bush, Active Virtual Network Management Prediction: Complexity as a Framework for Prediction, Optimization, and Assurance, Available at http://arxiv.org/PS_cache/cs/pdf/0203/0203014.pdf
3. D. S. Alexander, W. A. Arbaugh, M. W. Hicks, P. Kakkar, A. D. Keromytis, J. T. Moore, C. A. Gunter, S. M. Nettles, and J. M. Smith, The SwitchWare Active Network Architecture, *IEEE Network*, Vol. 12, No. 3, pp. 27–36, 1998.
4. K. Calvert, ed. Active Networks Framework, Available at http://www.cc.gatech.edu/projects/canes/papers/arch-1-0.ps.gz
5. M. Hicks, P. Kakkar, T. Moore, C. Gunter, and S. Nettles, PLAN: A Programmable Language for Active Networks, *ACM SIGPLAN Notices*, Vol. 34, No. 1, pp. 86–93, 1999.
6. D. Wetherall, J. Guttag, and D. Tennenhouse, ANTS: Network Services Without the Red Tape, *Computer*, Vol. 32, No. 4, pp. 42–48, 1999.
7. L. R. Ford Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.
8. R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, Network information flow, *IEEE Transactions on Information Theory*, Vol. 46, No. 4, pp. 1204–1216, 2000.
9. R. Koetter, Available at http://tesla.csl.uiuc.edu/koetter/NWC/
10. P. A. Chou, Y. Wu, and K. Jain, Practical network coding, In: *Proceedings of the 41st Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2003.
11. T. Ho, M. Médard, J. Shi, D. R. Karger, and M. Effros, On randomized network coding, In: *Proceedings of the 41st Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2003.

12. T. Ho, B. Leong, M. Médard, R. Koetter, Y.-H. Chang, and M. Effros, On the utility of network coding in dynamic environments, In: *Proceedings of the International Workshop on Wireless Ad-hoc Networks (IWWAN)*, Oulu, Finland, 2004.
13. T. Ho, M. Médard, and R. Koetter, An Information-Theoretic View of Network Management, *IEEE Transactions on Information Theory*, Vol. 51, No. 4, pp. 1295–1312, 2005.
14. R. Koetter and M. Médard, An Algebraic Approach to Network Coding, *IEEE/ACM Transactions on Networking*, Vol. 11, No. 5, pp. 782–795, 2003.
15. Y. Wu. P. A. Chou, and S.-Y. Kung, Minimum-energy multicast in mobile ad hoc networks using network coding, In *Proceedings of the Information Theory Workshop 2004*, San Antonio, TX, 2004.
16. Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, Network Planning in Wireless Ad Hoc Networks: A Cross-Layer Approach, *IEEE Journal on Selected Areas of Communications*. Vol. 23, no. 1, pp. 136–150, 2005.
17. G. B. Dantzig and D. R. Fulkerson, On the max-flow, min-cut theorem of networks, In: H. W. Kuhn (ed.), *Linear Inequalities*, Annals of Mathematical Studies, Number 38, Princeton University Press, Princeton, NJ, pp. 215–221, 1956.
18. P. Elias, A. Feinstein, and C. E. Shannon, A Note on the Maximum Flow through a Network, *IRE Transactions on Information Theory*, Vol. 2, No. 4, pp. 117–119, 1956.
19. L. R. Ford and D. R. Fulkerson, Maximal Flow through a Network, *Canadian Journal of Mathematics*, Vol. 8, No. 3, pp. 399–404, 1956.
20. J. T. Robacker, On Network Theory, The RAND Corporation, Research Memorandum RM-1498, May 26, 1955.
21. G. Kramer and S. A. Savari, On networks of two-way channels, In: A. Ashikhmin and A. Barg (eds.), *Algebraic Coding Theory and Information Theory, DIMACS Workshop, Dec. 15-18, 2003, Rutgers University*, *Vol. 68: DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society, Providence, Rhode Island, pp. 133–143, 2005.
22. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo. CA, 1988.
23. G. Kramer, *Directed Information for Channels with Feedback*, Hartung-Gorre Verlag, Konstanz, 1998, ETH Series in Information Processing, Vol. 11.
24. T. M. Cover and J. A. Thomas, *Elements of information Theory*, Wiley, New York, 1991.
25. S. P. Borade, Network information flow: Limits and achievability, In: *Proceedings of the IEEE International Symposium on Information Theory*, Lausanne, Switzerland, p. 139, June 30-July 5, 2002.
26. G. Kramer, Capacity Results for the Discrete Memoryless Network, *IEEE Transactions on Information Theory*, Vol. 49, No. 1, pp. 4–21, 2003.
27. A. Schrijver, *Combinatorial Optimization*, Springer-Verlag, New York, 2003.
28. T. C. Hu, Multi-commodity Network Flows, *Operations Research,* Vol. 11, No. 3, pp. 344–360, 1963.
29. H. Okamura and P. D. Seymour, Multicommodity Flows in Planar Graphs, *Journal of Combinatorial Theory, Series B,* Vol. 31, No. 1, pp. 75–81, 1981.
30. N. J. A. Harvey, R. D. Kleinberg, and A. R. Lehman. On the Capacity of Information Networks, *IEEE Transactions on Information Theory*, in press.
31. K. Jain, P. A. Chou, V. V. Vazirani, R. Yeung, and G. Yuval, On the Capacity of Multiple Unicast Sessions in Undirected Graphs, *IEEE Transactions on Information Theory*, in press.
32. DIMACS Working Group on Network Coding, DIMACS Center, Rutgers University, Piscataway, NJ, Jan. 26–28, 2005, Available at http://dimacs.rutgers.edu/Workshops/NetworkCodingWG.
33. G. Kramer and S. A. Savari. Progressive *d*-separating edge set bounds on network coding rates, In: *Proceedings of the 2005 IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005.

Edge-Cut Bounds on Network Coding Rates

Gerhard Kramer is a Member of Technical Staff in the Mathematics of Communications Research Department, Bell Laboratories, Lucent Technologies, Murray Hill, NJ. He is currently serving as a Publications Editor for the IEEE Transactions on Information Theory and has organized several workshops on coding and information theory.

Serap A. Savari is an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. Her research interests include information theory, data compression, network coding, computing and communication systems. She is currently serving as the Associate Editor for Source Coding for the IEEE Transactions on Information Theory and has been on the program committees of several information theory and data compression conferences.