Network Information Flow

Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, *Senior Member, IEEE*, and
Raymond W. Yeung, *Senior Member, IEEE*

Abstract—We introduce a new class of problems called *network information flow* which is inspired by computer network applications. Consider a point-to-point communication network on which a number of information sources are to be mulitcast to certain sets of destinations. We assume that the information sources are mutually independent. The problem is to characterize the admissible coding rate region. This model subsumes all previously studied models along the same line. In this paper, we study the problem with one information source, and we have obtained a simple characterization of the admissible coding rate region. Our result can be regarded as the Max-flow Min-cut Theorem for network information flow. Contrary to one's intuition, our work reveals that it is in general not optimal to regard the information to be multicast as a "fluid" which can simply be routed or replicated. Rather, by employing coding at the nodes, which we refer to as *network coding*, bandwidth can in general be saved. This finding may have significant impact on future design of switching systems.

Index Terms—Diversity coding, multicast, network coding, switching, multiterminal source coding.

I. INTRODUCTION

LET V be the set of nodes of a point-to-point communication network. Such a network is represented by a directed graph $G = (V, E)$, where E is the set of edges, such that information can be sent noiselessly from node i to node j for all $(i, j) \in E$. An example of this type of networks is the Internet backbone, where with proper data link protocols information can be sent between nodes essentially free of noise.

Let $X_1, \cdots, X_m$ be mutually independent information sources. The information rate (in bits per unit time) of X_i is denoted by h_i, and let $\boldsymbol{h} = [h_1 \cdots h_m]$. Let $a: \{1, \cdots, m\} \to V$ and $b: \{1, \cdots, m\} \to 2^V$ be arbitrary mappings. The source X_i is generated at node $a(i)$, and it is multicast to node j for all $j \in b(i)$. The mappings a, b, and the vector $\boldsymbol{h}$ specify a set of *multicast requirements*.

In this model, the graph G may represent a physical network, while the set of multicast requirements may represent the aggre-

gated traffic pattern the network needs to support. In other situations, the graph G may represent a subnetwork in a physical network, while the set of multicast requirements may pertain to a specific application on this subnetwork, e.g., a video-conference call.

In existing computer networks, each node functions as a switch in the sense that it either relays information from an input link to an output link, or it replicates information received from an input link and sends it to a certain set of output links. From the information-theoretic point of view, there is no reason to restrict the function of a node to that of a switch. Rather, a node can function as an encoder in the sense that it receives information from all the input links, encodes, and sends information to all the output links. From this point of view, a switch is a special case of an encoder. In the sequel, we will refer to coding at a node in a network as *network coding*.

Let R_{ij} be a nonnegative real number associated with the edge (i, j), and let $\boldsymbol{R} = [R_{ij}, (i, j) \in E]$. For a fixed set of multicast requirements, a vector $\boldsymbol{R}$ is *admissible* if and only if there exists a coding scheme satisfying the set of multicast requirements such that the coding rate from node i to node j (i.e., the average number of bits sent from node i to node j per unit time) is less than or equal to R_{ij} for all $(i, j) \in E$. (At this point we leave the details of a coding scheme open because it is extremely difficult to define the most general form of a coding scheme. A class of coding schemes called α-codes will be studied in Section III.) In graph theory, R_{ij} is called the capacity of the edge (i, j). Our goal is to characterize the admissible coding rate region $\mathcal{R}$, i.e., the set of all admissible $\boldsymbol{R}$, for any graph G and multicast requirements a, b, and $\boldsymbol{h}$.

The model we have described includes both multilevel diversity coding (without distortion) [12], [8], [13] and distributed source coding [14] as special cases. As an illustration, let us show how the multilevel diversity coding system in Fig. 1 can be formulated as a special case of our model. In this system, there are two sources, X_1 and X_2. Decoder 1 reconstructs X_1 only, while all other decoders reconstruct both X_1 and X_2. Let r_i be the coding rate of Encoder i, $i = 1, 2, 3$. In our model, the system is represented by the graph G in Fig. 2. In this graph, node 1 represents the source, nodes 2, 3, and 4 represent the inputs of Encoders 1, 2, and 3, respectively, nodes 5, 6, and 7 represent the outputs of Encoders 1, 2, and 3, respectively, while nodes 8, 9, 10, and 11 represent the inputs of Decoders 1, 2, 3, and 4, respectively. The mappings a and b are specified as

$$a(1) = 1 \quad a(2) = 1$$

and

$$b(1) = \{8, 9, 10, 11\} \quad b(2) = \{9, 10, 11\}$$

R. Ahlswede is with Fakultät für Mathematik, Universität Bielefeld, 33501 Bielefeld, Germany (e-mail: hollmann@Mathematik.uni-Bielefeld.de).

N. Cai was with Fakultät für Mathematik, Universität Bielefeld, 33501 Bielefeld, Germany. He is now with the Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong (e-mail: ncai@ie.cuhk.edu.hk).

S.-Y. R. Li and R. W. Yeung are with the Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong (e-mail: syli@ie.cuhk.hk; whyeung@ie.cuhk.hk).

Fig. 1.   A multilevel diversity coding system.

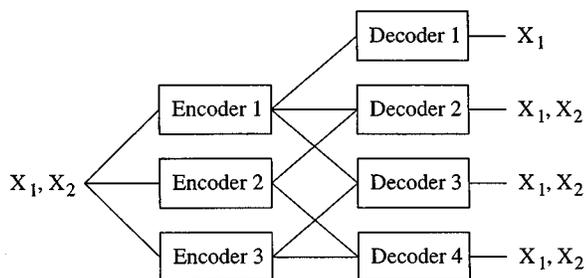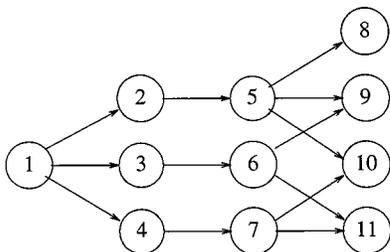Fig. 2.   The graph G representing the coding system in Fig. 1.

Fig. 3.   A single-level diversity coding system.

Fig. 4.   The graph representing the coding system in Fig. 3.

and $\boldsymbol{h} = [h_1 h_2]$ represents the information rates of X_1 and X_2. Now all the edges in G except for $(2,5)$, $(3,6)$, $(4,7)$ correspond to straight connections in Fig. 1, so there is no constraint on the coding rate in these edges. Therefore, in order to determining $\mathcal{R}$, the set of all admissible $\boldsymbol{R}$ for the graph G (with the set of multicast requirements specified by a, b, and $\boldsymbol{h}$), we set $R_{ij} = \infty$ for all edges in G except for $(2,5)$, $(3,6)$, $(4,7)$ to obtain the admissible coding rate region of the problem in Fig. 1.

A major finding in this paper is that, contrary to one's intuition, it is in general not optimal to consider the information to be multicast in a network as a "fluid" which can simply be routed or replicated at the intermediate nodes. Rather, network coding has to be employed to achieve optimality. This fact is illustrated by examples in the next section.

In the rest of the paper, we focus our discussion on problems with $m = 1$, which we collectively refer to as the single-source problem. For problems with $m \geq 2$, we refer to them collectively as the multisource problem. The rest of the paper is organized as follows. In Section II, we propose a Max-flow Min-cut theorem which characterizes the admissible coding rate region of the single-source problem. In Section III, we formally state the main result in this paper. The proof is presented in Sections IV and V. In Section VI, we show that very simple optimal codes do exist for certain networks. In Section VII, we use our results for the single-source problem to solve a special case of the multisource problem which has application in video conferencing. In this section, we also show that the multisource problem is extremely difficult in general. Concluding remarks are in Section VIII.

II. A MAX-FLOW MIN-CUT THEOREM

In this section, we propose a theorem which characterizes the admissible coding rate region for the single-source problem. For this problem, we let $a(1) = s$, and $b(1) = \{t_1, \cdots, t_L\}$. In
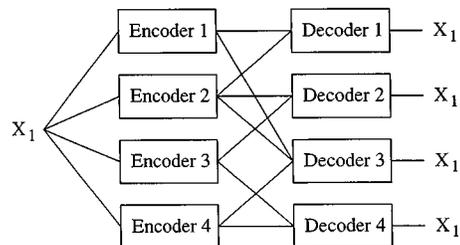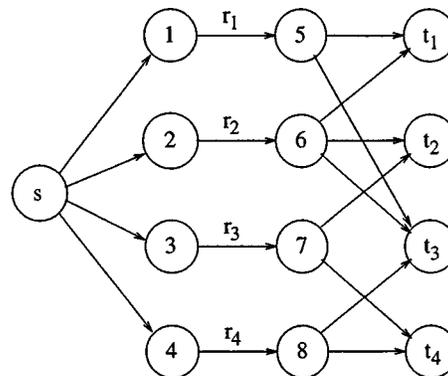
other words, the information source X_1 is generated at node s and is multicast to nodes $t_1, \cdots, t_L$. We will call s the *source* and $t_1, \cdots, t_L$ the *sinks* of the graph G. For a specific L, the problem will be referred to as the one-source L-sink problem.

Let us first define some notations and terminology which will be used in the rest of the paper. Let $G = (V, E)$ be a graph with source s and sinks $t_1, \cdots, t_L$. The capacity of an edge $(i, j) \in E$ is given by R_{ij}, and let $\boldsymbol{R} = [R_{ij}, (i, j) \in E]$. The subgraph of G from s to t_l, $l = 1, \cdots, L$, refers to the graph $G_l = (V, E_l)$, where

$$E_l = \{(i, j) \in E: (i, j) \text{ is on a directed path from } s \text{ to } t_l\}.$$

$\boldsymbol{F} = [F_{ij}, (i, j) \in E]$ is a *flow* in G from s to t_l if for all $(i, j) \in E$

$$0 \leq F_{ij} \leq R_{ij}$$

such that for all $i \in V$ except for s and t_l

$$\sum_{i':(i',i)\in E} F_{i'i} = \sum_{j:(i,j)\in E} F_{ij}$$

i.e., the total flow into node i is equal to the total flow out of node i. F_{ij} is referred to as the value of $\boldsymbol{F}$ in the edge (i, j). The value of $\boldsymbol{F}$ is defined as

$$\sum_{j:(s,j)\in E} F_{sj} - \sum_{i:(i,s)\in E} F_{is}$$

which is equal to

$$\sum_{i:(i,t_l)\in E} F_{it_l} - \sum_{j:(t_l,j)\in E} F_{t_lj}.$$

F is a *max-flow* from s to t_l in G if F is a flow from s to t_l whose value is greater than or equal to any other flow from s to
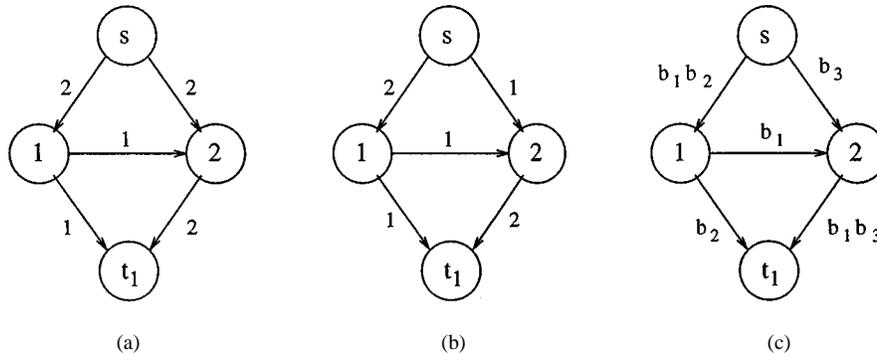
Fig. 5.   A one-source one-sink network.

t_l. Evidently, a max-flow from s to t_l in G is also a max-flow from s to t_l in G_l. For a graph with one source and one sink (for example, the graph G_l), the value of a max-flow from the source to the sink is called the *capacity* of the graph.

We begin our discussion by first reviewing a basic result of diversity coding by considering the single-level diversity system in Fig. 3. In this system, X_1 is the only information source (with rate h_1), and it is reconstructed by all the decoders. Henceforth, we will drop the subscripts of X_1 and h_1 when there is only one information source. Let r_i be the coding rate of encoder i, and let $\boldsymbol{r} = [r_1 \cdots r_4]$. In order for a decoder to reconstruct X, it is necessary that the sum of the coding rates of the encoders accessible by this decoder is at least h. Thus the conditions

$$r_1 + r_2 \geq h \tag{1}$$
$$r_2 + r_3 \geq h \tag{2}$$
$$r_1 + r_2 + r_4 \geq h \tag{3}$$
$$r_3 + r_4 \geq h \tag{4}$$

are necessary for $\boldsymbol{r}$ to be admissible. On the other hand, these conditions are seen to be sufficient by the work of Singleton [10] (also cf. [12]).

We now give a graph-theoretic interpretation of the above result. The graph corresponding to the system in Fig. 3 is given in Fig. 4, where we use s to label the source and t_i to label a sink. Now $R_{15}, R_{26}, R_{37},$ and R_{48} correspond to $r_1, r_2, r_3,$ and r_4, respectively. So the edges $(1, 5), (2, 6), (3, 7),$ and $(4, 8)$ are labeled accordingly. The quantities $r_1, r_2, r_3,$ and r_4 are interpreted as the capacity (in the sense of graph theory) of the corresponding edges. For the other edges in the graph, each one of them corresponds to a straight connection in the system in Fig. 3. Since there is no constraint on the coding rate in these edges, we interpret the capacity of each of them as infinity. To keep the graph simple, we do not label these edges. By considering the subgraph from s to t_1 in Fig. 4, the condition in (1) can be interpreted as the value of the max-flow from s to t_1 being greater than or equal to h, the information rate of the source. Similar interpretations can be made for the conditions in (2)-(4).

Based on the graph-theoretic interpretation of the above diversity coding problem (which is a one-source four-sink problem), we make the following conjecture.

Conjecture 1: Let $G = (V, E)$ be a graph with source s and sinks $t_1, \cdots, t_L$, and the capacity of an edge (i, j) be denoted by R_{ij}. Then $(\boldsymbol{R}, h, G)$ is admissible if and only if the values of

a max-flow from s to $t_l, l = 1, \cdots, L$ are greater than or equal to h, the rate of the information source.

The spirit of our conjecture resembles that of the celebrated Max-flow Min-cut Theorem in graph theory [1]. Before we end this section, we give a few examples to illustrate our conjecture. We first illustrate by the example in Fig. 5 that the conjecture is true for $L = 1$. Fig. 5(a) shows the capacity of each edge. By the Max-flow Min-cut Theorem [1], the value of a max-flow from s to t_1 is 3, so the flow in Fig. 5(b) is a max-flow. In Fig. 5(c), we show how we can send three bits b_1, b_2, b_3 from s to t_1 based on the max-flow in Fig. 5(b). The conjecture is trivially seen to be true for $L = 1$, because when there is only one sink, we only need to treat the raw information bits as physical entities. The bits are routed at the intermediate nodes according to any fixed routing scheme, and they will all eventually arrive at the sink. Since the routing scheme is fixed, the sink knows which bit is coming in from which edge, and the information can be recovered accordingly.

Next we illustrate by the example in Fig. 6 that the conjecture is true for $L = 2$. Fig. 6(a) shows the capacity of each edge. It is easy to check that the value of a max-flow from s to t_1 and to t_2 are 5 and 6, respectively. So the conjecture asserts that we can send 5 bits b_1, b_2, b_3, b_4, b_5 to t_1 and t_2 simultaneously, and Fig. 6(b) shows such a scheme. Note that in this scheme, bits only need to be replicated at the nodes to achieve optimality.

We now show another example in Fig. 7 to illustrate that the conjecture is true for $L = 2$. Fig. 7(a) shows the capacity of each edge. It is easy to check that the value of a max-flow from s to t_l is 2, $l = 1, 2$. So the conjecture asserts that we can send 2 bits b_1, b_2 to t_1 and t_2 simultaneously, and Fig. 7(b) shows such a scheme, where "+" denotes modulo 2 addition. At t_1, b_2 can be recovered from b_1 and $b_1 + b_2$. Similarly, b_1 can be recovered at t_2. Note that when there is more than one sink, we can no longer think of information as a real entity, because information needs to be replicated or transformed at the nodes. In this example, information is coded at the node 3, which is unavoidable. For $L \geq 2$, network coding is in general necessary in an optimal multicast scheme.

Finally, we illustrate by the example in Fig. 8 that the conjecture is true for $L = 3$. Fig. 8(a) shows the capacity of each edge. It is easy to check that the values of a max-flow from s to all the sinks are 2. In Fig. 8(b), we show how we can multicast 2 bits b_1, b_2 to all the sinks.
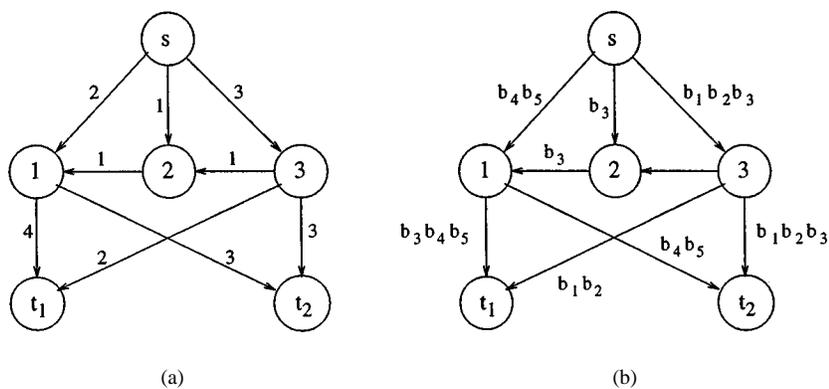
(a)   (b)

Fig. 6. A one-source two-sink network without coding.
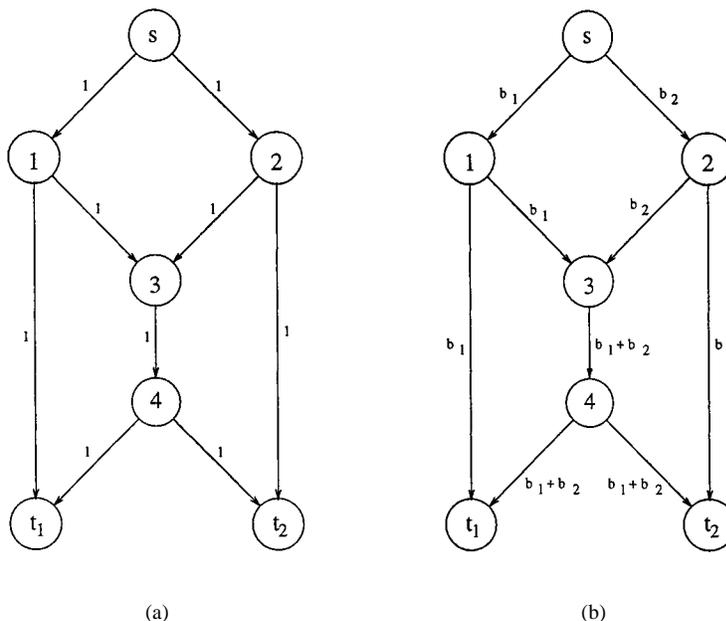

(a)   (b)

Fig. 7. A one-source two-sink network with coding.

The advantage of network coding can be seen from the examples in Figs. 7 and 8. As an illustration, we will quantify this advantage for the example in Fig. 8 in two ways. First, we investigate the saving in bandwidth when network coding is allowed. For the scheme in Fig. 8(b), a total of 9 bits are sent. If network coding is not allowed, then it is easy to see that at least one more bit has to be sent in order that for t_1, t_2 and t_3 to recover both b_1 and b_2. Thus we see that a very simple network code can save 10% in bandwidth. Second, we investigate the increase in throughput when network coding is allowed. Using the scheme in Fig. 8(b), if 2 bits are sent in each edge, then 4 bits can be multicast to all the sinks. If network coding is not allowed (and 2 bits are sent in each edge), we now show that only 3 bits can be multicast to all the sinks. Let $B = \{b_1, \cdots, b_\kappa\}$ be the set of bits to be multicast to all the sinks. Let the set of bits sent in the edge (s, i) be B_i, where $|B_i| = 2$, $i = 1, 2, 3$. At node i, the received bits are duplicated and sent in the two out-going edges. Thus 2 bits are sent in each edge in the network. Since network coding is not allowed, $B = B_i \cup B_j$ for any $1 \leq i < j \leq 3$. Then we have

$$B_3 \cup (B_1 \cap B_2) = (B_3 \cup B_1) \cap (B_3 \cup B_2) = B.$$

Therefore,

$$\begin{aligned} \kappa &= |B_3 \cup (B_1 \cap B_2)| \\ &\leq |B_3| + |B_1 \cap B_2| \\ &= |B_3| + |B_1| + |B_2| - |B_1 \cup B_2| \\ &= 6 - \kappa \end{aligned}$$

which implies $\kappa \leq 3$. In Fig. 8(c), we show how 3 bits b_1, b_2, and b_3 can be multicast to all the sinks by sending 2 bits in each edge. Therefore, the throughput of the network can be increased by one-third using a very simple network code.

III. MAIN RESULT

In this section, we formally present the main result in this paper. Let $G = (V, E)$ be a directed graph with source s and sinks $t_1, \cdots, t_L$, and R_{ij} be the capacity of an edge (i, j) in E. Since our conjecture concerns only the values of max-flows from the source to the sinks, we assume without loss of generality that there is no edge in E from a node (other than s) to s, because such an edge does not increase the value of a max-flow
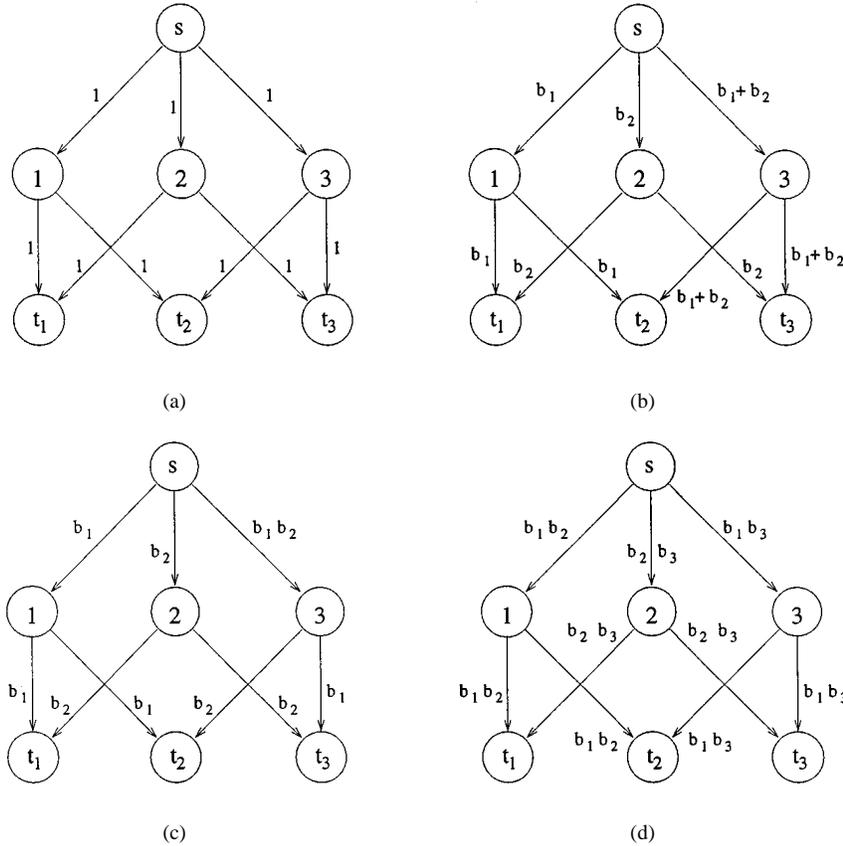
Fig. 8.   A one-source three-sink network.

from s to a sink. Further, we assume $(i, i) \notin E$ for all $i \in V$ for the same reason.

Let us consider a block code of length n. We assume that x, the value assumed by X, is obtained by selecting an index from a set Ω with uniform distribution. The elements in Ω are called messages. For $(i, j) \in E$, node i can send information to node j which depends only on the information previously received by node i. Since the graph G is arbitrary and may contain (directed) cycles, a network code can in general be very complicated. In this paper, we confine our discussion to a class of block codes, called the α-code, which is defined in the next paragraph.

An $(n, (\eta_{ij}, (i, j) \in E), h)$ α-code on a graph G is defined by the following components (the construction of an α-code from these components will be described after their definitions are given):

1) a positive integer K.

2) $u: \{1, \cdots, K\} \to V$, $v: \{1, \cdots, K\} \to V$, such that $(u(k), v(k)) \in E$.

3) $A_k = \{1, \cdots, |A_k|\}\ (|A_k| \geq 1), 1 \leq k \leq K$, such that
$$\prod_{k \in T_{ij}} |A_k| = \eta_{ij}$$
where
$$T_{ij} = \{1 \leq k \leq K: (u(k), v(k)) = (i, j)\}.$$

4) If $u(k) = s$, then $f_k: \Omega \to A_k$, otherwise
$$f_k: \prod_{k' \in Q_k} A_{k'} \to A_k$$

where
$$\Omega = \{1, \cdots, \lceil 2^{nh} \rceil\}$$
and
$$Q_k = \{1 \leq k' < k: v(k') = u(k)\}.$$

5) $g_l: \prod_{k' \in W_l} A_{k'} \to \Omega, 1 \leq l \leq L$, where
$$W_l = \{1 \leq k \leq K: v(k) = t_l\}$$
such that for all $1 \leq l \leq L$, $\tilde{g}_l(x) = x$ for all $x \in \Omega$, where $\tilde{g}_l(x)$ denotes the value of g_l as a function of x.

The $(n, (\eta_{ij}, (i, j) \in E), h)$ α-code is constructed from these components as follows. At the beginning of the coding session, the value of X is available to node s. In the coding session, there are K transactions which take place in chronological order, where each transaction refers to a node sending information to another node. In the kth transaction, node $u(k)$ encodes according to f_k and sends an index in A_k to node $v(k)$. The domain of f_k is the information received by node $u(k)$ so far, and we distinguish two cases. If $u(k) = s$, the domain of f_k is Ω. If $u(k) \neq s$, Q_k gives the indices of all previous transactions for which information was sent to node $u(k)$, so the domain of f_k is $\prod_{k' \in Q_k} A_{k'}$. The set T_{ij} gives the indices of all transactions for which information is sent from node i to node j, so η_{ij} is the number of possible index-tuples that can be sent from node i to node j during the coding session. Finally, W_l gives the indices of all transactions for which information is sent to t_l, and g_l is the decoding function at t_l.

We remark that the α-code is not the most general possible definition of a block code. For example, the order of transactions can depend on the value of X. Also, coding can be done probabilistically. (However, we prove in the Appendix that probabilistic coding does not improve performance.) Instead of a block code, it is also possible to use a variable-length code.

Let $\boldsymbol{R} = [R_{ij}, (i,j) \in E]$. A tuple $(\boldsymbol{R}, h, G)$ is α-*admissible* if for any $\epsilon > 0$ there exists, for sufficiently large n, an $(n, (\eta_{ij}, (i,j) \in E), h - \epsilon)$ α-code on G such that

$$n^{-1} \log_2 \eta_{ij} \leq R_{ij} + \epsilon$$

for all $(i,j) \in E$. (Note that α-admissibility implies admissibility.) Define

$$\mathcal{R}_{h,G} = \{\boldsymbol{R} : (\boldsymbol{R}, h, G) \text{ is } \alpha\text{-admissible}\}.$$

The problem is to characterize $\mathcal{R}_{h,G}$ for any h and G.

For a directed graph $G = (V, E)$ with source s, sinks $t_1, \cdots, t_L$, and the capacity of an edge (i,j) equals R_{ij}, let $\mathcal{R}_{h,G}^*$ be the set consisting of all $\boldsymbol{R}$ such that the values of a max-flow from s to $t_l, l = 1, \cdots, L$ are greater than or equal to h. The following theorem is the main result in this paper.

Theorem 1: $\mathcal{R}_{h,G} = \mathcal{R}_{h,G}^*$.

IV. THE CONVERSE

In this section, we prove that $\mathcal{R}_{h,G} \subset \mathcal{R}_{h,G}^*$, i.e., if for any $\epsilon > 0$ there exists for sufficiently large n an $(n, (\eta_{ij}, (i,j) \in E), h - \epsilon)$ α-code on G such that

$$n^{-1} \log_2 \eta_{ij} \leq R_{ij} + \epsilon$$

for all $(i,j) \in E$, then the values of a max-flow from s to $t_l, l = 1, \cdots, L$ are greater than or equal to h.

Consider any $1 \leq l \leq L$ and any $B \subset V$ such that $s \in B$ and $t_l \notin B$. Let

$$E_B = \{(i,j) \in E : i \in B \text{ and } j \notin B\}.$$

Let

$$w_{t_l}(x) = (\tilde{f}_k(x), k \in \cup_{i \in V} T_{it_l})$$

where $x \in \Omega$ and $\tilde{f}_k(x)$ denotes the value of f_k as a function of x. $w_{t_l}(x)$ is all the information known by t_l during the whole coding session when the message is x. Since for an α-code, $\tilde{f}_k(x)$ is a function of the information previously received by node $u(k)$, we see inductively that $w_{t_l}(x)$ is a function of

$$\tilde{f}_k(x), k \in \bigcup_{(i,j) \in E_B} T_{ij}.$$

Since x can be determined at node t_l, we have

$$
\begin{aligned}
H(X) &= H(w_{t_l}(X)) \\
&= H\left(\tilde{f}_k(X), k \in \bigcup_{(i,j) \in E_B} T_{ij}\right) \\
&\leq \sum_{(i,j) \in E_B} \sum_{k \in T_{ij}} H(\tilde{f}_k(X)) \\
&\leq \sum_{(i,j) \in E_B} \sum_{k \in T_{ij}} \log_2 |A_k|
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{(i,j) \in E_B} \log_2 \left(\prod_{k \in T_{ij}} |A_k|\right) \\
&= \sum_{(i,j) \in E_B} \log_2 \eta_{ij}.
\end{aligned}
$$

Thus

$$
\begin{aligned}
h - \epsilon &\leq n^{-1} H(X) \\
&\leq \sum_{(i,j) \in E_B} n^{-1} \log_2 \eta_{ij} \\
&\leq \sum_{(i,j) \in E_B} (R_{ij} + \epsilon) \\
&\leq \sum_{(i,j) \in E_B} R_{ij} + |E|\epsilon.
\end{aligned}
$$

Minimizing the right-hand side over all B, we have

$$h - \epsilon \leq \min_B \sum_{(i,j) \in E_B} R_{ij} + |E|\epsilon.$$

By the Max-flow Min-cut Theorem [1], the first term on the right-hand side is equal to the value of a max-flow from s to t_l. Letting $\epsilon \to 0$, we obtain the desired conclusion.

As a remark, even if we allow an arbitrarily small probability of decoding error in the usual Shannon sense, by modifying our proof by means of a standard application of Fano's inequality [2], it can be seen that it is still necessary for the value of a max-flow from s to $t_l, l = 1, \cdots, L$ to be greater than or equal to h. The details are omitted here.

V. ADMISSIBILITY

In this section, we prove that $\mathcal{R}_{h,G}^* \subset \mathcal{R}_{h,G}$. In Section V-A, we first prove the result when the graph G is acyclic. Then this result will be used to prove the general case in Section V-B.

A. Acyclic Networks

Assume the graph G is acyclic. Let the vertices in G be labeled by $0, 1, \cdots, |V| - 1$ in the following way. The source s has the label 0. The other vertices are labeled in a way such that for $1 \leq j \leq |V| - 1$, $(i,j) \in E$ implies $i < j$. Such a labeling is possible because G is acyclic. We regard $s, t_1, \cdots, t_L$ as aliases of the corresponding vertices.

We will consider an $(n, (\eta_{ij}, (i,j) \in E), h)$ β-code on the graph G defined by

$$f_{sj} : \Omega \to \{1, \cdots, \eta_{sj}\}$$

for all $j \in V$ such that $(s,j) \in E$, where $\Omega = \{1, \cdots, \lceil 2^{nh} \rceil\}$

$$f_{ij} : \prod_{i' : (i',i) \in E} \{1, \cdots, \eta_{i'i}\} \to \{1, \cdots, \eta_{ij}\}$$

for all $(i,j) \in E$ such that $i \neq s$, and

$$g_l : \prod_{i' : (i',t_l) \in E} \{1, \cdots, \eta_{i't_l}\} \to \Omega$$

for all $1 \leq l \leq L$ such that $\tilde{g}_l(x) = x$ for all $x \in \Omega$ (recall that $\tilde{g}_l(x)$ denotes the value of g_l as a function of x). In the above, f_{ij} is the encoding function for the edge (i,j), while g_l is the decoding function for the sink t_l. In the coding session, f_{ij} is applied before $f_{i'j'}$ if $i < i'$, and f_{ij} is applied

before $f_{ij'}$ if $j < j'$. This defines the order in which the encoding functions are applied. Since $i' < i$ if $(i', i) \in E$, all the necessary information is available when encoding at node i is done. If the set $\{i' : (i', i) \in E\}$ is empty, we adopt the convention that f_{ij} is an arbitrary constant taken from the set $\{1, \cdots, \eta_{ij}\}$. An $(n, (\eta_{ij}, (i, j) \in E), H)$ β-code is a special case of an $(n, (\eta_{ij}, (i, j) \in E), h)$ α-code defined in Section III.

Now assume that the vector $\boldsymbol{R}$ is such that, with R_{ij} being the capacity of the edge (i, j) in G, for all $1 \leq l \leq L$, the values of a max-flow from s to t_l is greater than or equal to h. It suffices for us to show that for any $\epsilon > 0$, there exists for sufficiently large n an $(n, (\eta_{ij}, (i, j) \in E), h - \epsilon)$ β-code on G such that

$$n^{-1} \log_2 \eta_{ij} \leq R_{ij} + \epsilon$$

for all $(i, j) \in E$. Instead, we will show the existence of an $(n, (\eta_{ij}, (i, j) \in E), h)$ β-code satisfying the same set of conditions, and this will be done by a random procedure. For the time being, let us replace Ω by $\Omega' = \{1, \cdots, \lceil C2^{nh}\rceil\}$, where C is any constant greater than 1. Thus the domain of f_{sj} is expanded from Ω to Ω' for $(s, j) \in E$.

We now construct the encoding functions as follows. For all $j \in V$ such that $(s, j) \in E$, for all $x \in \Omega'$, $f_{sj}(x)$ is defined to be a value selected independently from the set $\{1, \cdots, \eta_{sj}\}$ with uniform distribution. For all $(i, j) \in E, i \neq s$, and for all

$$z \in \prod_{i' : (i', i) \in E} \{1, \cdots, \eta_{i'i}\}$$

$f_{ij}(z)$ is defined to be a value selected independently from the set $\{1, \cdots, \eta_{ij}\}$ with uniform distribution.

Let $z_s(x) = x$, and for $j \in V, j \neq s$, let $z_j(x) = (\tilde{f}_{ij}(x), (i, j) \in E)$, where $x \in \Omega'$ and $\tilde{f}_{ij}(x)$ denotes the value of f_{ij} as a function of x. $z_j(x)$ is all the information received by node j during the coding session when the message is x. For distinct $x, x' \in \Omega$, x and x' are indistinguishable at the sink t_l if and only if $z_{t_l}(x) = z_{t_l}(x')$. For all $x \in \Omega$, define

$$F(x) = \begin{cases} 1, & \text{if } z_{t_l}(x) = z_{t_l}(x') \\ & \text{for some } 1 \leq l \leq L \text{ and } x' \in \Omega, x' \neq x \\ 0, & \text{otherwise.} \end{cases}$$

$F(x)$ is equal to 1 if and only if x cannot be uniquely determined at at least one of the sinks. Now fix $x \in \Omega$ and $1 \leq l \leq L$. Consider any $x' \in \Omega$ not equal to x and define the sets

$$\begin{aligned} B_0 &= \{i \in V : z_i(x) \neq z_i(x')\} \\ B_1 &= \{i \in V : z_i(x) = z_i(x')\}. \end{aligned}$$

Obviously, $s \in B_0$ and $B_1 = V \backslash B_0$.

Now suppose $z_{t_l}(x) = z_{t_l}(x')$. Then $B_0 = B$ for some $B \subset V$, where $s \in B$ and $t_l \notin B$. For any $(i, j) \in E$

$$P\{\tilde{f}_{ij}(x) = \tilde{f}_{ij}(x') | z_i(x) \neq z_i(x')\} = \eta_{ij}^{-1}.$$

Therefore,

$$\begin{aligned} P\{B_0 = B\} &= P\{B_0 = B, B_0 \supset B\} \\ &= P\{B_0 = B | B_0 \supset B\} P\{B_0 \supset B\} \\ &\leq P\{B_0 = B | B_0 \supset B\} \\ &= \prod_{(i, j) \in E_B} \eta_{ij}^{-1} \end{aligned}$$

where

$$E_B = \{(i, j) \in E : i \in B, j \notin B\}.$$

Let ϵ be any fixed positive real number. For all $(i, j) \in E$, take η_{ij} such that

$$R_{ij} + \zeta \leq n^{-1} \log \eta_{ij} \leq R_{ij} + \epsilon$$

for some $0 < \zeta < \epsilon$. Then

$$\begin{aligned} P\{B_0 = B\} &\leq \prod_{(i, j) \in E_B} 2^{-n(R_{ij} + \zeta)} \\ &\leq 2^{-n\left(\zeta + \sum_{(i, j) \in E_B} R_{ij}\right)} \\ &\leq 2^{-n(h + \zeta)}. \end{aligned}$$

In the second inequality above, we have used $|E_B| \geq 1$, and the last inequality follows from the Max-flow Min-cut Theorem [1]. Note that this upper bound does not depend on B. Since V has $2^{|V|}$ subsets, and B is some subset of V

$$P\{z_{t_l}(x) = z_{t_l}(x')\} \leq 2^{|V|} 2^{-n(h + \zeta)}.$$

Further,

$$\begin{aligned} P\{z_{t_l}(x) = z_{t_l}(x') &\text{ for some } x' \in \Omega', x' \neq x\} \\ &\leq (\Omega' - 1) 2^{|V|} 2^{-n(h + \zeta)} \\ &< C2^{nh} 2^{|V|} 2^{-n(h + \zeta)} \\ &= C2^{|V|} 2^{-n\zeta}. \end{aligned}$$

Therefore,

$$P\{F(x) = 1\} < LC2^{|V|} 2^{-n\zeta}.$$

Let $\delta(n, \zeta) = LC2^{|V|} 2^{-n\zeta}$. Then $\delta(n, \zeta) \to 0$ as $n \to \infty$. Hence, there exists a deterministic code for which the number of messages which can be uniquely determined at all sinks is at least

$$(1 - \delta(n, \zeta))C2^{nh}$$

which is greater than 2^{nh} for sufficiently large n. Let Ω to be any set of 2^{nh} such messages in Ω'. Upon defining

$$g_l(z) = x$$

where $x \in \Omega$ such that $z = z_{t_l}(x), 1 \leq l \leq L$ and

$$z \in \prod_{i' : (i', t_l) \in E} \{1, \cdots, \eta_{i't_l}\},$$

we have obtained a desired $(n, (\eta_{ij}, (i, j) \in E) h)$ β-code. The theorem is proved.

B. Cyclic Networks

For cyclic networks, there is no natural ordering of the nodes which allows coding in a sequential manner as in our discussion on acyclic networks in the last section. In this section, we will

prove our result in full generality which involves the construction of a more elaborate code.

Consider any graph $G = (V, E)$ (acyclic or cyclic) with source s and sinks $t_1, \cdots, t_L$, and the capacity of an edge $(i, j) \in E$ given by R_{ij}. Assume for all $l = 1, \cdots, L$, the value of a max-flow from s to t_l is greater than or equal to h. We will prove that $(\boldsymbol{R}, h, G)$ is α-admissible.

We first construct a time-parametrized graph $G^* = (V^*, E^*)$ from the graph G. The set V^* consists of $\Lambda + 1$ layers of nodes, each of which is a copy of V. Specifically,

$$V^* = \bigcup_{\lambda=0}^{\Lambda} V^{(\lambda)}$$

where

$$V^{(\lambda)} = \{i^{(\lambda)} : i \in V\}.$$

As we will see later, λ is interpreted as the time parameter. The set E^* consists of the following three types of edges:

1) $(s^{(0)}, s^{(\lambda)})$, $1 \leq \lambda \leq \Lambda$;
2) $(t_l^{(\lambda)}, t_l^{(\lambda)})$, $1 \leq \lambda \leq \Lambda - 1$;
3) $(i^{(\lambda)}, j^{(\lambda+1)})$, $(i, j) \in E$, $0 \leq \lambda \leq \Lambda - 1..$

For G^*, let $s^* = s^{(0)}$ be the source, and let $t_l^* = t_l^{(\Lambda)}$ be a sink which corresponds to the sink t_l in G, $1 \leq l \leq L$. Clearly, G^* is acyclic because each edge in G^* ends at a vertex in a layer with a larger index.

Let the capacities of the edges in G be given by $\boldsymbol{R} = [R_{ij}, (i, j) \in E]$. Let $R_{uv}^*, (u, v) \in E^*$ be the capacity of an edge $(u, v) \in E^*$, where

$$R_{uv}^* = \begin{cases} R_{ij}, & \text{if } (u, v) = (i^{(\lambda)}, j^{(\lambda+1)}) \\ & \text{for some } (i, j) \in E \text{ and } 1 \leq \lambda \leq \Lambda - 1 \\ \infty, & \text{otherwise} \end{cases} \quad (5)$$

and let $\boldsymbol{R}^* = [R_{uv}^*, (u, v) \in E^*]$.

Lemma 1: Let s and t be the source and the sink of a graph G, respectively. Then there exists a max-flow $\boldsymbol{F}$ in G expressible as the sum of a number of flows for which each of them consists of a simple path (i.e., a directed path without cycle) from s to t only.

Proof: Let $\boldsymbol{F}$ be a max-flow from s to t in G which does not contain a positive directed cycle (cf. [1, p. 45]). Let P_1 be any positive path from s to t in $\boldsymbol{F}$ (evidently P_1 is simple), and let c_1 be the minimum value of $\boldsymbol{F}$ in an edge along P_1. Let $\tilde{\boldsymbol{F}}^1$ be the flow from s to t along P_1 with value c_1. Subtracting $\tilde{\boldsymbol{F}}^1$ from $\boldsymbol{F}$, $\boldsymbol{F}$ is reduced to $\boldsymbol{F} - \tilde{\boldsymbol{F}}^1$, a flow from s to t which does not contain a positive directed cycle. Apply the same procedure repeatedly until $\boldsymbol{F}$ is reduced to the zero flow. The lemma is proved. $\qquad \square$

Lemma 2: For $l = 1, \cdots, L$, if the value of a max-flow from s to t_l in G is greater than or equal to h, then the value of a max-flow from s^* to t_l^* in G^* is greater than or equal to $(\Lambda -$

$d_l)h$, where d_l is the maximum length of a simple path from s to t_l.

Proof: Let $1 \leq l \leq L$ be fixed. Let $\bar{\boldsymbol{F}}$ be a max-flow from s to t_l in G with value h such that $\bar{\boldsymbol{F}}$ does not contain a positive directed cycle. Using the last lemma, we can write $\bar{\boldsymbol{F}} = \tilde{\boldsymbol{F}}^1 + \cdots + \tilde{\boldsymbol{F}}^w$, where $\tilde{\boldsymbol{F}}^r$, $r = 1, \cdots, w$ contains a positive simple path P_r from s to t only. Specifically

$$\tilde{\boldsymbol{F}}_{ij}^r = \begin{cases} c_r, & \text{if } (i, j) \in P_r \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where

$$c_1 + \cdots + c_w = h. \quad (7)$$

Let q_r be the length of P_r. For an edge $(i, j) \in P_r$, let $a_r(i, j)$ be the distance of node i from s along P_r. Clearly,

$$a_r(i, j) \leq q_r \leq d_l.$$

Now for $0 \leq \lambda \leq \Lambda - d_l - 1$, define

$$\boldsymbol{F}^{(\lambda, r)} = [F_{uv}^{(\lambda, r)}, (u, v) \in E^*]$$

where

$$F_{uv}^{(\lambda, r)} = \begin{cases} c_r, & \text{if } (u, v) = (s^*, s^{(\lambda)}), 1 \leq \lambda \leq \Lambda - d_l \\ c_r, & \text{if } (u, v) = (i^{(\lambda + a_r(i,j))}, j^{(\lambda + a_r(i,j) + 1)}) \\ & \text{where } (i, j) \in P_r \\ c_r, & \text{if } (u, v) = (t_l^{(\lambda + q_r)}, t_l^*) \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Since

$$\lambda + q_r \leq \lambda + d_l \leq \Lambda - 1$$

the third case in (8) and hence $\boldsymbol{F}^{(\lambda, r)}$ is well defined for $0 \leq \lambda \leq \Lambda - d_l - 1$. $\boldsymbol{F}^{(\lambda, r)}$ is a flow from s^* to t_l^* in G^* derived from the flow $\tilde{\boldsymbol{F}}^r$ in G as follows. A flow of c_r is generated at s^* and enters the λth layer of nodes from $s^{(\lambda)}$. Then the flow traverses consecutive layers of nodes by emulating the path P_r in G until it eventually reaches $t_l^{(\lambda + q_r)}$, and it finally leaves G^* at the sink t_l^*. Based on $\boldsymbol{F}^{(\lambda, r)}$, we construct

$$\boldsymbol{F}^{(\lambda)} = \sum_{r=1}^{w} \boldsymbol{F}^{(\lambda, r)}$$

and

$$\boldsymbol{F}^* = \sum_{\lambda=0}^{\Lambda - d_l - 1} \boldsymbol{F}^{(\lambda)}.$$

We will prove that $\boldsymbol{F}^* \leq \boldsymbol{R}^*$ componentwise. Then $\boldsymbol{F}^*$ is a flow from s^* to t_l^* in G^*, and from (7), its value is given by

$$\sum_{\lambda=0}^{\Lambda - d_l - 1} \sum_{r=1}^{w} c_r = \sum_{\lambda=0}^{\Lambda - d_l - 1} h = (\Lambda - d_l)h.$$

This implies that the value of a max-flow from s^* to t_l^* in G^* is at least $(\Lambda - d_l)h$, and the lemma is proved.

Toward proving that $\boldsymbol{F}^* \leq \boldsymbol{R}^*$, we only need to consider $(u, v) \in E^*$ such that $(u, v) = (i^{(\lambda)}, j^{(\lambda+1)})$ for some $(i, j) \in E$ and $0 \leq \lambda \leq \Lambda - 1$, because R_{uv}^* is infinite otherwise (cf. (5)).

For notational convenience, we will adopt the convention that $\boldsymbol{F}^{(\lambda, r)} = 0$ for $\lambda < 0$. Now for $0 \le \lambda \le \Lambda - 1$ and $(i, j) \in E$

$$
\begin{aligned}
F^*_{i^{(\lambda)} j^{(\lambda+1)}} &= \sum_{\mu=0}^{\Lambda - d_l - 1} F^{(\mu)}_{i^{(\lambda)} j^{(\lambda+1)}} \\
&= \sum_{\mu=0}^{\Lambda - d_l - 1} \sum_{r=1}^{w} F^{(\mu, r)}_{i^{(\lambda)} j^{(\lambda+1)}} \\
&= \sum_{r=1}^{w} F^{(\lambda - a_r(i,j), r)}_{i^{(\lambda)} j^{(\lambda+1)}} \\
&\le \sum_{r=1}^{w} \tilde{F}^r_{ij} \\
&= F_{ij} \\
&\le R_{ij} \\
&= R^*_{i^{(\lambda)} j^{(\lambda+1)}}.
\end{aligned}
$$

In the above, the third equality follows from (8), and the first inequality is justified as follows. First, the inequality is justified for $\lambda < a_r(i,j)$ since $\boldsymbol{F}^{(\mu, r)} = 0$ for $\mu < 0$. For $\lambda \ge a_r(i,j)$, we distinguish two cases. From (8) and (6), if $(i, j) \in P_r$, we have

$$
F^{(\lambda - a_r(i,j), r)}_{i^{(\lambda)} j^{(\lambda+1)}} = \tilde{F}^r_{ij} = c_r.
$$

If $(i, j) \notin P_r$, we have

$$
F^{(\lambda - a_r(i,j), r)}_{i^{(\lambda)} j^{(\lambda+1)}} = \tilde{F}^r_{ij} = 0.
$$

Thus the inequality is justified for all cases. Hence we conclude that $\boldsymbol{F}^* \le \boldsymbol{R}^*$, and the lemma is proved. $\square$

From Lemma 2 and the result in Section V-A, we see that $(\boldsymbol{R}^*, (\Lambda - d)h, G^*)$ is α-admissible, where $d = \max_{1 \le l \le L} d_l$. Thus for every $\epsilon > 0$, there exists for sufficiently large ν a $(\nu, (\eta^*_{uv}, (u, v) \in E^*, (\Lambda - d)h)$ β-code on G^* such that

$$
\nu^{-1} \log_2 \eta^*_{uv} \le R^*_{uv} + \epsilon
$$

for all $(u, v) \in E^*$. For this β-code on G^*, let us use f^*_{uv} to denote the encoding function for an edge $(u, v) \in E^*$, and use g^*_l to denote the decoding function at the sink t^*_l, $1 \le l \le L$. Without loss of generality, we assume that for $1 \le \lambda \le \Lambda - 1$

$$
f^*_{s^* s^{(\lambda)}}(x) = x
$$

for all x in

$$
\Omega = \{1, \cdots, 2^{\nu(\Lambda - d)h}\}
$$

and

$$
f^*_{t_l^{(\lambda)} t_l^*}(y) = y
$$

for all y in

$$
\prod_{k:(k, t_l^{(\lambda)}) \in E} \{1, \cdots, \eta^*_{k^{(\lambda-1)} t_l^{(\lambda)}}\}, \qquad 1 \le l \le L.
$$

Note that if the β-code does not satisfy these assumptions, it can readily be converted into one.

Let Λ be a positive integer, and let $n = \Lambda \nu$. Using the β-code on G^*, we now construct an $(n, (\eta_{ij}, (i, j) \in E), (\Lambda - d)h/\Lambda)$ γ-code on G, where

$$
\eta_{ij} = \prod_{\lambda=0}^{\Lambda-1} \eta^*_{i^{(\lambda)} j^{(\lambda+1)}},
$$

which is defined by the following components:

1) for $(i, j) \in E$ such that $i \ne s$, a constant $f^{(1)}_{ij}$ taken from the set $\{1, \cdots, \eta^*_{i^{(0)} j^{(1)}}\}$;
2) for $1 \le \lambda \le \Lambda$

$$
f^{(\lambda)}_{sj} : \Omega \to \{1, \cdots, \eta^*_{s^{(\lambda-1)} j^{(\lambda)}}\}
$$

   for all $j \in V$ such that $(s, j) \in E$, where

$$
\Omega = \{1, \cdots, 2^{\nu(\Lambda-d)h}\}
$$

   and for $2 \le \lambda \le \Lambda$

$$
f^{(\lambda)}_{ij} : \prod_{k:(k,i) \in E} \{1, \cdots, \eta^*_{k^{(\lambda-2)} i^{(\lambda-1)}}\}
$$
$$
\to \{1, \cdots, \eta^*_{i^{(\lambda-1)} j^{(\lambda)}}\}
$$

   for all $(i, j) \in E$ such that $i \ne s$ (if the set $\{k : (k, i) \in E\}$ is empty, we adopt the convention that $f^{(\lambda)}_{ij}$ is a constant taken from $\{1, \cdots, \eta^*_{i^{(\lambda-1)} j^{(\lambda)}}\}$);
3) for $1 \le l \le L$

$$
g_l : \prod_{\lambda=0}^{\Lambda-1} \prod_{i:(i, t_l) \in E} \{1, \cdots, \eta^*_{i^{(\lambda)} t_l^{(\lambda+1)}}\} \to \Omega
$$

   such that $\tilde{g}_l(x) = x$ for all $x \in \Omega$ (recall that $\tilde{g}_l(x)$ denotes the value of g_l as a function of x);

where

1) for $(i, j) \in E$ such that $i \ne s$, $f^{(1)}_{ij} = f^*_{i^{(0)} j^{(1)}}$ ($f^*_{i^{(0)} j^{(1)}}$ is an arbitrary constant in $\{1, \cdots, \eta^*_{i^{(0)} j^{(1)}}\}$ since $\{u \in V^* : (u, i^{(0)}) \in E^*\}$ is empty);
2) for $1 \le \lambda \le \Lambda$, for all $x \in \Omega$, $f^{(\lambda)}_{sj}(x) = f^*_{s^{(\lambda-1)} j^{(\lambda)}}(x)$, and for $2 \le \lambda \le \Lambda$ and all $(i, j) \in E$ such that $i \ne s$, $f^{(\lambda)}_{ij}(y) = f^*_{i^{(\lambda-1)} j^{(\lambda)}}(y)$ for all y in

$$
\prod_{k:(k,i) \in E} \{1, \cdots, \eta^*_{k^{(\lambda-2)} i^{(\lambda-1)}}\};
$$

3) for $1 \le l \le L$, $g_l(z) = g^*_l(z)$ for all z in

$$
\prod_{\lambda=0}^{\Lambda-1} \prod_{i:(i, t_l) \in E} \{1, \cdots, \eta^*_{i^{(\lambda)} t_l^{(\lambda+1)}}\}.
$$

The coding process of the γ-code consists of $\Lambda + 1$ phases:

1) In Phase 1, for all $(i, j) \in E$ such that $i \ne s$, node i sends $f^{(1)}_{ij}$ to node j, and for all $j \in V$ such that $(s, j) \in E$, node s send $f^{(1)}_{sj}(x)$ to node j.
2) In Phase λ, $2 \le \lambda \le \Lambda$, for all $(i, j) \in E$, node i sends $\tilde{f}^{(\lambda)}_{ij}(x)$ to node j, where $\tilde{f}^{(\lambda)}_{ij}(x)$ denotes the value of $f^{(\lambda)}_{ij}$ as a function of x, and it depends only on $\tilde{f}^{(\lambda-1)}_{ki}(x)$

for all $k \in V$ such that $(k, i) \in E$, i.e., the information received by node i during Phase $\lambda - 1$.

3) In Phase $\Lambda + 1$, for $1 \leq l \leq L$, the sink t_l uses g_l to decode x.

From the definitions, we see that an $(n, (\eta_{ij}, (i, j) \in E), (\Lambda - d)h/\Lambda)$ γ-code on G is a special case of an $(n, (\eta_{ij}, (i, j) \in E), (\Lambda - d)h/\Lambda)$ α-code on G. For the γ-code we have constructed

$$
\begin{aligned}
n^{-1} \log_2 \eta_{ij} &= (\Lambda \nu)^{-1} \log_2 \prod_{\lambda=0}^{\Lambda-1} \eta^*_{i(\lambda)j(\lambda+1)} \\
&\leq \Lambda^{-1} \sum_{\lambda=0}^{\Lambda-1} (R^*_{i(\lambda)j(\lambda+1)} + \epsilon) \\
&= \Lambda^{-1} \sum_{\lambda=0}^{\Lambda-1} (R^*_{ij} + \epsilon) \\
&= R_{ij} + \epsilon
\end{aligned}
$$

for all $(i, j) \in E$. Finally, for any $\epsilon > 0$, by taking a sufficiently large Λ, we have

$$
\frac{(\Lambda - d)h}{\Lambda} > h - \epsilon.
$$

Hence, we conclude that $(\boldsymbol{R}, h, G)$ is α-admissible.

VI. AN EXAMPLE

Despite the complexity of our proof of Theorem 1 in the last two sections, we will show in this section that very simple optimal codes do exist for certain cyclic networks. Therefore, there is much room for further research on how to design simple optimal codes for (single-source) network information flow. The code we construct in this section can be regarded as a kind of convolutional code, which possesses many desirable properties of a practical code.

Consider the graph $G = (V, E)$ in Fig. 9, where

$$
V = \{s, v_0, v_1, v_2, u_0, u_1, u_2, t_0, t_1, t_2\}
$$

and E contains the following types of edges for $l = 0, 1, 2$:

1) (s, v_l);
2) (v_l, u_l);
3) $(v_l, t_{l\ominus 1})$ and $(v_l, t_{l\ominus 2})$;
4) $(u_l, u_{l\ominus 1})$;
5) $(u_l, t_{l\ominus 1})$;

where $\oplus$ denotes modulo 3 addition. Here s is the source and t_0, t_1, t_2 are the sinks. In the graph G, the edges (u_0, u_1), (u_1, u_2), and (u_2, u_0) form a cycle.

In this example, we let the information rate h be 3. Without loss of generality, we assume that the information source S generates three symbols $x_1(k), x_2(k), x_3(k)$ at time $k = 1, 2, \cdots$, where $x_l(k)$ are elements of some finite field GF (q). For the purpose of our discussion, we can regard the sequence of symbols $\{x_l(k)\}$ as deterministic. Consider the rate tuple $\boldsymbol{R} = \boldsymbol{1}$, the vector whose components are all equal to 1. Then the value of a max-flow from s to t_l is 3, $l = 0, 1, 2$, and Theorem 1 asserts that $\boldsymbol{R}$ is admissible.
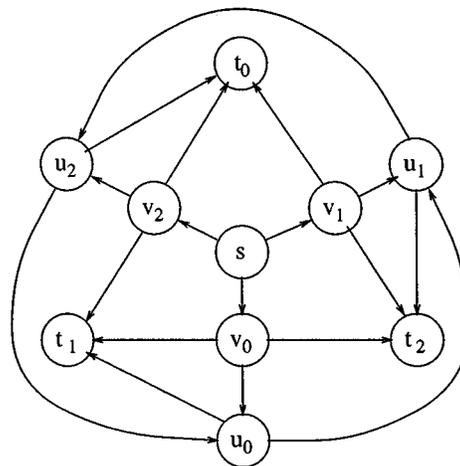

Fig. 9. An example of a cyclic network.

We now show that $\boldsymbol{R}$ is admissible by presenting a coding scheme which can multicast $\{x_0(k), x_1(k), x_2(k)\}$ from the source to all the sinks. To simplify notation, we adopt the convention that $x_l(k) = 0$ for $k \leq 0$. At time $k \geq 1$, information transactions occur in the following order:

T1. s sends $x_l(k)$ to v_l, $l = 0, 1, 2$
T2. v_l sends $x_l(k)$ to u_l, $t_{l\ominus 1}$, and $t_{l\ominus 2}$, $l = 0, 1, 2$
T3. u_0 sends $x_0(k) + x_1(k-1) + x_2(k-1)$ to u_1
T4. u_1 sends $x_0(k) + x_1(k-1) + x_2(k-1)$ to t_2
T5. u_1 sends $x_0(k) + x_1(k) + x_2(k-1)$ to u_2
T6. u_2 sends $x_0(k) + x_1(k) + x_2(k-1)$ to t_0
T7. u_2 sends $x_0(k) + x_1(k) + x_2(k)$ to u_0
T8. u_0 sends $x_0(k) + x_1(k) + x_2(k)$ to t_1
T9. t_2 decodes $x_2(k-1)$
T10. t_0 decodes $x_0(k)$
T11. t_1 decodes $x_1(k)$

where "+" denotes addition in GF (q). Note that the coding rate in each edge is equal to 1, since exactly one symbol is sent in each edge in one unit time.

We now show that these information transactions can actually be performed. Let us start at $k = 1$. At $k = 1$, T1 and T2 can obviously be performed. T3 and T4 can be performed because $x_1(0) = x_2(0) = 0$. T5 and T6 can be performed since $x_0(1)$ and $x_1(1)$ have been sent to u_1 from u_0 and v_1, respectively. T7 and T8 can be performed since $x_0(1) + x_1(1)$ and $x_2(1)$ have been sent to u_2 from u_1 and v_2, respectively. T9 can obviously be performed since $x_2(0) = 0$. T10 can be performed since $x_1(1), x_2(1)$, and $x_0(1) + x_1(1)$ have been sent to t_0 from v_1, v_2, and u_2, respectively. Finally, T11 can be performed since $x_0(1), x_2(1)$, and $x_0(1) + x_1(1) + x_2(1)$ have been sent to t_1 from v_0, v_2, and u_0, respectively.

Now assume that T1-T11 can be performed up to time $k - 1$ for some $k \geq 2$, and we will show that they can be performed at time k. T1 and T2 can obviously be performed. Just before T3 is performed, $x_0(k)$ has been sent to u_0 from v_0 at time k, and $x_0(k-1)$ and $x_0(k-1) + x_1(k-1) + x_2(k-1)$ have been sent to u_0 from v_0 and u_2, respectively, at time $k - 1$. Therefore, at time k, u_0 can determine $x_0(k) + x_1(k-1) + x_2(k-1)$, and T3

and T4 can be performed accordingly. By similar arguments, we see that T5–T8 can be performed. Just before T9 is performed, $x_0(k)$ and $x_0(k) + x_1(k-1) + x_2(k-1)$ have been sent to t_2 from v_0 and u_1, respectively, at time k, and $x_1(k-1)$ has been sent to t_2 from v_1 at time $k-1$. Therefore, at time k, T9 can be performed. By similar arguments, we see that T10 and T11 can be performed.

At time k, $x_{l\ominus1}(k)$ and $x_{l\ominus2}(k)$ are sent to t_l from $v_{l\ominus1}$ and $v_{l\ominus2}$, respectively, $l = 0, 1, 2$. With T9–T11, at time k, t_0 and t_1 can recover $x_0(k'), x_1(k')$ and $x_2(k')$ for all $k' \leq k$, while t_2 can recover $x_0(k')$ and $x_1(k')$ for all $k' \leq k$, and $x_2(k')$ for all $k' \leq k-1$. Note the unit time delay for t_2 to recover $x_2(k)$. Thus our coding scheme can multicast $\{x_0(k), x_1(k), x_2(k)\}$ to all the sinks, and hence $\boldsymbol{R = 1}$ is admissible.

It is also possible to design convolutional codes for an acyclic network. Compared with the block code we used in proving Theorem 2, it seems that a convolution code has the advantage that the code can be very simple, and both the memory at each node and the end-to-end decoding delay can be very small.

VII. MULTIPLE SOURCES

In the classical information theory for point-to-point communication, if two information sources are independent, optimality can be achieved (asymptotically) by coding the sources separately. This coding method is referred to as coding by superposition [12]. If this coding method is always optimal for multisource network information flow problems, then in order to solve the problem, we only need to solve the subproblems for the individual information sources separately, where each of these subproblems is a single-source problem. However, as we will see shortly, the multisource problem is not a trivial extension of the single-source problem, and it is extremely difficult in general.

Let us consider the multilevel diversity coding system in Fig. 1. Assume that $h_1 = h_2 = 1$. Since the sources X_1 and X_2 are independent, if coding by superposition is always optimal, then for any admissible coding rate triple (r_1, r_2, r_3), for $i = 1, 2, 3$, we can write

$$r_i = r_i^1 + r_i^2$$

where r_i^1 and r_i^2 are the subrates associated with sources X_1 and X_2, respectively. Since X_1 is multicast to all the decoders, from the discussion in Section II, we have the following constraints for $r_i^1, i = 1, 2, 3$:

$$r_1^1 \geq 1$$
$$r_1^1 + r_2^1 \geq 1$$
$$r_2^1 + r_3^1 \geq 1$$
$$r_1^1 + r_3^1 \geq 1.$$

Similarly, since X_2 is multicast to Decoders 2, 3, and 4, we have the following constraints for $r_i^2, i = 1, 2, 3$:

$$r_1^2 + r_2^2 \geq 1$$
$$r_2^2 + r_3^2 \geq 1$$
$$r_1^2 + r_3^2 \geq 1.$$

However, it was shown in [12] that the rate triple $(1, 1, 1)$ is admissible, but it cannot be decomposed into two sets of subrates

as prescribed above. Therefore, coding by superposition is not optimal in general, even when the two information sources are generated at the same node.

In [5],[1] it was found that coding by superposition is optimal for 86 out of all 100 configurations of multilevel diversity coding systems with three encoders. In [8] and [13] it was shown that coding by superposition is optimal for all symmetrical multilevel diversity coding systems. However, how to characterize multilevel diversity coding systems for which coding by superposition is always optimal is still an open problem.

Although the multisource problem in general is extremely difficult, there exist special cases which can be readily solved by the results for the single-source problem. Consider a network information flow problem with m information sources. Let $a(i) = s_i$, and suppose $b(i) = \{t_1, \cdots, t_L\}$, $1 \leq i \leq m$. Here, each information source X_i is multicast from s_i to sinks $t_1, \cdots, t_L$. It turns out that this problem can be reduced to a single-source problem by adding the following components to the graph G:

1) a node s;

2) edges (s, s_i), $1 \leq i \leq m$.

Set R_{s, s_i} to h_i, the information rate of X_i, $1 \leq i \leq m$, and call this augmented graph G'. Then we can regard all information sources $X_1, \cdots, X_m$ as one information source X (with rate $\sum_{i=1}^{m} h_i$) generated at node s, where X_i is sent to node s_i via edge (s, s_i) at rate h_i. Then the problem can be regarded as a one-source L-sink problem on the graph G' with source s and sinks $t_1, \cdots, t_L$.

In video-conferencing, the information generated by each participant is multicast to all other participants on the network. This is a special case of the situation described in the last paragraph.

VIII. DISCUSSION

In this paper, we have proposed a new class of problems called network information flow which is inspired by computer network applications. This class of problems consolidates all previous work along this line [12], [8], [14] into a new direction in multiterminal source coding.

In the past, most results in multiterminal source coding are generalizations of either the Slepian–Wolf problem [9] or the multiple descriptions problem [3]. The class of problems we have proposed are generalizations of neither of these problems. Further, they distinguish themselves from most classical multiterminal source coding problems in the following ways:

1) there is no rate-distortion consideration;
2) the sources are mutually independent;
3) the network configuration, described by a graph, is arbitrary;
4) the reconstruction requirements are arbitrary.

Our formulation covers a large class of problems instead of one particular problem. For most classical multiterminal source

---

[1]The reader can contact Raymond Yeung for a copy of this reference.

coding problems, the problem degenerates if there is no rate-distortion consideration and the sources are mutually independent. For our class of problems, neither of these assumptions is made. Yet they are highly nontrivial problems.

In this paper, we have characterized the admissible coding rate region of the single-source problem. Our result can be regarded as the Max-flow Min-cut Theorem for network information flow. We point out that our discussion is based on a class of block codes called α-codes. Therefore, it is possible, though not likely, that our result can be enhanced by considering more general coding schemes. Nevertheless, we prove in the Appendix that probabilistic coding does not improve performance.

In analog telephony, when a point-to-point call is established, there is a physical connection between the two parties. When a conference call is established, there is a physical connection among all the parties involved. In computer communication (which is digital), we used to think that for multicasting, there must be a logical connection among all the parties involved such that raw information bits are sent to the destinations via such a connection. The notion of a logical connection in computer communication is analogous to the notion of a physical connection in analog telephony. As a result, multicasting in a computer network is traditionally being thought of as replicating bits at the nodes, so that each sink eventually receive a copy of all the bits. The most important contribution of the current paper is to show that the traditional technique for multicasting in a computer network in general is not optimal. Rather, we should think of information as being "diffused" through the network from the source to the sinks by means of network coding. This is a new concept in multicasting in a point-to-point network which may have significant impact on future design of switching systems.

In classical information theory for point-to-point communication, we can think of information as a "fluid" or some kind of physical entity. For network information flow with one source, this analogy continues to hold when there is one sink, because information flow conserves at all the intermediate nodes in an optimal scheme. However, the analogy fails for multicasting because information needs to be replicated or coded at the nodes.

The problem becomes more complicated when there are more than one source. In the classical information theory for point-to-point communication, if two sources are independent, optimality can be achieved (asymptotically) by coding the sources separately. However, it has been shown by a simple example in [12] that for simultaneous multicast of two sources, it may be necessary to code the sources jointly in order to achieve optimality. A special case of the multisource multisink problem which finds application in satellite communication has been studied in [14]. In this work, they obtained inner and outer bounds on the admissible coding rate region.

For future research, the multisource multisink problem is a challenging problem. For the single-source problem, there are still many unresolved issues which are worth further investigation. In proving our result for acyclic graphs, we have used a random block code. Recently, Li and Yeung [4] have devised a systematic procedure to construct linear codes for acyclic networks. Along another line, the example in Section V shows that convolutional codes are good alternatives to block codes. It seems that convolutional codes have the advantage that the code can be very simple, and the memory at each node and the

end-to-end decoding delay can be very small. These are all desirable features for practical codes.

Finally, by imposing the constraint that network coding is not allowed, i.e., each node functions as a switch in existing computer networks, we can ask whether a rate tuple $\boldsymbol{R}$ is admissible. Also, we can ask under what condition can optimality be achieved without network coding. These are interesting problems for further research.

Recently, there has been a lot of interest in factor graph [7], a graphical model which subsumes Markov random field, Bayesian network, and Tanner graph. In particular, the problem of representing codes in graphs [11], [6] has received much attention. The codes we construct for a given network in this paper can be regarded as a special type of codes in a graph.

APPENDIX
PROBABILISTIC CODING DOES NOT IMPROVE PERFORMANCE

For an α-code, the kth transaction of the coding process is specified by a mapping f_k. Suppose instead of the mapping f_k, the kth transaction is specified by a transition matrix from the domain of f_k to the range of f_k. Also, instead of the mapping g_l, decoding at sink t_l is specified by a transition matrix from the domain of g_l to the range of g_l, $1 \le l \le L$. Then the code becomes a probabilistic code, and we refer to such a code as a probabilistic α-code. With a slight abuse of notation, we continue to use $\tilde{f}_k$ to denote the code in the kth transaction (where $\tilde{f}_k$ is a random variable), and we use w_{t_l} to denote $(\tilde{f}_k, k \in \cup_{i \in V} T_{it_l})$.

In general, one can use probabilistic coding schemes instead of deterministic coding schemes. By using probabilistic schemes, it may be possible to multicast information from s to t_l, $1 \le l \le L$ at a rate higher than that permitted by deterministic schemes. Before showing that this is impossible, however, we first discuss a subtlety of probabilistic coding.

For a probabilistic α-code on a graph G, it seems intuitively correct that for any $1 \le l \le L$ and any $B \subset V$ such that $s \in B$ and $t_l \notin B$, the information source X, $(\tilde{f}_k, k \in \cup_{(i,j) \in E_B} T_{ij})$, and w_{t_l} form a Markov chain because all the information sent from s to t_l has to go through the set of nodes $\{j \in E : (i,j) \in E_B\}$. If this is the case, then by the Data Processing Theorem [2], we have

$$H(\tilde{f}_k, k \in \cup_{(i,j) \in E_B} T_{ij}) \ge I(X; \tilde{f}_k, k \in \cup_{(i,j) \in E_B} T_{ij})$$
$$\ge I(X; w_{t_l})$$
$$= H(X)$$

where the last equality holds because X can be recovered at sink t_l. However, we show next by an example that the Markov chain asserted is not valid in general.

Consider the graph G in Fig. 10 with three nodes $s, 1$, and t. Let $X = (X_1, X_2)$ be uniformly distributed on $\mathrm{GF}(2)^2$, and Z be independent of X and uniformly distributed on $\mathrm{GF}(2)$. Consider the following probabilistic α-code with five transactions:

$$u(1) = s, \quad v(1) = 1, \quad \tilde{f}_1 = X_1$$
$$u(2) = 1, \quad v(2) = t, \quad \tilde{f}_2 = X_1 + Z$$
$$u(3) = t, \quad v(3) = s, \quad \tilde{f}_3 = X_1 + Z$$
$$u(4) = s, \quad v(4) = 1, \quad \tilde{f}_4 = (X_1, X_2 + Z)$$
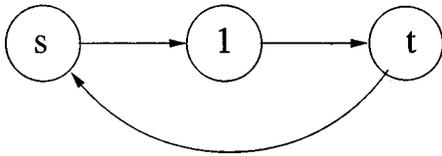$$u(5) = 1, \quad v(5) = t, \quad \tilde{f}_5 = (X_1, X_2 + Z).$$

Fig. 10.   A three-node network.

Note that the fourth transaction is possible because upon knowing X_1 and $X_1 + Z$, Z can be determined. For $B = \{s\}$, $E_B = \{(s, 1)\}$. Then

$$(\tilde{f}_k, k \in \cup_{(i,j)\in E_B} T_{ij})$$
$$= (\tilde{f}_k, k \in T_{s1}) = (\tilde{f}_1, \tilde{f}_4) = (X_1, X_2 + Z)$$

and

$$w_t = (X_1 + Z, X_2 + Z, X_1).$$

Now from w_t, both X_1 and X_2 can be recovered. However, from $(\tilde{f}_k, k \in \cup_{(i,j)\in E_B} T_{ij})$, it is impossible to recover X_2. Therefore, the Markov chain asserted in the last paragraph is invalid.

We now show that the use of probabilistic coding cannot reduce coding rates. Consider any probabilistic coding scheme, and let Θ be the random parameter (assumed to be real) of the scheme with distribution function $F_\Theta(\theta)$. Without loss of generality, we assume that Θ is independent of X. This assumption can be justified by showing that if Θ is not independent of X, then we can construct an equivalent probabilistic coding scheme whose random parameter Θ' is independent of X. Define an independent random vector $\Theta' = [\Theta_x, x \in \mathcal{X}]$, where $\mathcal{X}$ denotes the alphabet set of X; $\Theta_x, x \in \mathcal{X}$ are mutually independent; and Θ_x has marginal distribution function $F_{\Theta|X}(\theta|x)$. We then use Θ_x as the random parameter of the coding scheme when the message is x. This coding scheme, which uses Θ' as the random parameter, is equivalent to the original scheme using Θ as the random parameter.

Let $\boldsymbol{r}(x, \theta) = [r_{ij}(x, \theta), (i, j) \in E]$ be the coding rate tuple incurred when the message is x and the random parameter takes the value θ. (Here the coding scheme can be variable-length, so $\boldsymbol{r}$ may depend on x.) Since Θ and X are independent, the average coding rate tuple of this coding scheme is given by

$$\boldsymbol{r} = \int \sum_x p(x)\boldsymbol{r}(x, \theta)\, dF_\Theta(\theta).$$

Now observe that for a fixed θ, the coding scheme becomes deterministic. Therefore, the probabilistic coding scheme is actually a mixture of deterministic coding schemes. By time-sharing these deterministic coding schemes according to $F_\Theta(\theta)$ (use approximation if necessary), we obtain a deterministic coding scheme. Hence, any coding rate tuple achievable by a probabilistic coding scheme can be achieved asymptotically by a sequence of deterministic coding schemes.

REFERENCES

[1] B. Bollobas, *Graph Theory, An Introductory Course*.   New York: Springer-Verlag, 1979.
[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*.   New York: Wiley, 1991.
[3] A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Theory*, vol. IT–28, pp. 851–857, Nov. 1982.
[4] S.-Y. R. Li and R. W. Yeung, "Linear network coding," *IEEE Trans. Inform. Theory*, submitted for publication.
[5] K. P. Hau, "Multilevel diversity coding with independent data streams," M.Phil. thesis, The Chinese Univ. Hong Kong, June 1995.
[6] R. Koetter and A. Vardy, "Factor graphs, trellis formations, and generalized state realizations," presented at the Institute for Mathematics and Its Applications, Aug. 6, 1999, available at http://www.ima.umn.edu/talks/workshops/aug12-13.99/koetter.html.
[7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, submitted for publication.
[8] J. R. Roche, R. W. Yeung, and K. P. Hau, "Symmetrical multilevel diversity coding," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1059–1064, May 1997.
[9] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.
[10] R. C. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inform. Theory*, vol. IT-10, pp. 116–118, Apr. 1964.
[11] N. Wiberg, H.-A. Loileger, and R. Koetter, "Codes and iterative decoding on general graphs," *Euro. Trans. Telecommun.*, vol. 6, pp. 513–526, Sept. 1995.
[12] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Trans. Inform. Theory*, vol. 41, pp. 412–422, Mar. 1995.
[13] R. W. Yeung and Z. Zhang, "On symmetrical multilevel diversity coding," *IEEE Trans. Inform. Theory*, vol. 45, pp. 609–621, Mar. 1999.
[14] ——, "Distributed source coding for satellite communications," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1111–1120, May 1999.