Linear Network Coding

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

***Abstract*—Consider a communication network in which certain source nodes multicast information to other nodes on the network in the multihop fashion where every node can pass on any of its received data to others. We are interested in how fast *each* node can receive the complete information, or equivalently, what the information rate arriving at *each* node is. Allowing a node to encode its received data before passing it on, the question involves optimization of the multicast mechanisms at the nodes. Among the simplest coding schemes is linear coding, which regards a block of data as a vector over a certain base field and allows a node to apply a linear transformation to a vector before passing it on. We formulate this multicast problem and prove that linear coding suffices to achieve the optimum, which is the max-flow from the source to each receiving node.**

***Index Terms*—Coding, network routing, switching.**

I. Introduction

DEFINE a *communication network* as a pair (G, S), where G is a finite *directed* multigraph and S is the unique node in G without any incoming edges. A directed edge in G is called a *channel* in the communication network (G, S). The special node S is called the *source*, while every other node may serve as a *sink* as we shall explain.

A channel in graph G represents a noiseless communication link on which one unit of information (e.g., a bit) can be transmitted per unit time. The multiplicity of the channels from a node X to another node Y represents the *capacity* of direct transmission from X to Y. In other words, every single channel has unit capacity.

At the source S, a finite amount of information is generated and multicast to other nodes on the network in the multihop fashion where every node can pass on any of its received data to other nodes. At each nonsource node which serves as a sink, the complete information generated at S is recovered. We are naturally interested in how fast each sink node can receive the complete information.

As an example, consider the multicast of two data bits, b_1 and b_2, from the source S in the communication network depicted by Fig. 1(a) to both nodes Y and Z. One solution is to let the channels ST, TY, TW, and WZ carry the bit b_1 and channels SU, UZ, UW, and WY carry the bit b_2. Note that in this scheme, an intermediate node sends out a data bit only if it receives the same bit from another node. For example, the node T receives the bit b_1 and sends a copy on each of the two channels TY and TW. Similarly, the node U receives the bit b_2 and sends a copy into each of the two channels UW and UZ. In our model, we assume that there is no processing delay at the intermediate nodes.

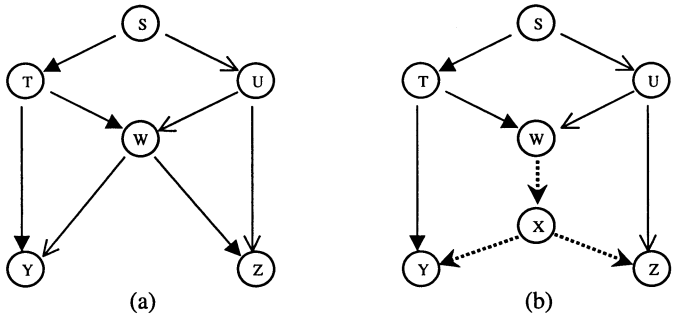

Fig. 1. Two communication networks.

Unlike a conserved physical commodity, information can be replicated or coded. Introduced in [1] (see also [5, Ch. 11]), the notion of network coding refers to coding at the intermediate nodes when information is multicast in a network. Let us now illustrate network coding by considering the communication network depicted by Fig. 1(b). In this network, we want to multicast two bits b_1 and b_2 from the source S to both the nodes Y and Z. A solution is to let the channels ST, TW, TY carry the bit b_1, channels SU, UW, UZ carry the bit b_2, and channels WX, XY, XZ carry the exclusive-OR $b_1 \oplus b_2$. Then, the node Y receives b_1 and $b_1 \oplus b_2$, from which the bit b_2 can be decoded. Similarly, the node Z can decode the bit b_1 from b_2 and $b_1 \oplus b_2$. The coding/decoding scheme is assumed to have been agreed upon beforehand.

It is not difficult to see that the above scheme is the only solution to the problem. In other words, without network coding, it is impossible to multicast two bits per unit time from the source S to both the nodes Y and Z. This shows the advantage of network coding. In fact, replication of data can be regarded as a special case of network coding.

As we have pointed out, the natures of physical commodities and information are very different. Nevertheless, both of them are governed by certain laws of flow. For the network flow of a physical commodity, we have the following.

> The law of commodity flow: The total volume of the outflow from a nonsource node cannot exceed the total volume of the inflow.

Manuscript received November 4, 1999; revised September 12, 2002. The work of S.-Y. R. Li was supported in part by an MTK Computers Grant. The work of R. W. Yeung and N. Cai was supported in part by the RGC Earmarked Grant CUHK4343/98E. The material in this paper was presented in part at the 1999 IEEE Information Theory Workshop, Metsovo, Greece, June/July 1999.

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: bobli@ie.cuhk.edu.hk; whyeung@ie.cuhk.edu.hk).

N. Cai was with Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong. He is now with the Fakultät für Mathematik, Universität Bielefeld, 33501 Bielefeld, Germany (e-mail: cai@mathematik.uni-bielefeld.de).

Communicated by V. Anantharam, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2002.807285

The counterpart for a communication network is as follows.

> The law of information flow: The content of any information flowing out of a set of nonsource nodes can be derived from the accumulated information that has flown into the set of nodes.

After all, information replication and coding do not increase the information content.

The information rate from the source to a sink can potentially become higher and higher when the permitted class of coding schemes is wider and wider. However, the law of information flow limits this information rate to the max-flow (i.e., the maximum commodity flow) from the source to that particular sink for a very wide class of coding schemes. The details are given in [1].

It has been proved in [1] that the information rate from the source to a set of nodes can reach the minimum of the individual max-flow bounds through coding. In the present paper, we shall prove constructively that by linear coding alone, the rate at which a message reaches each node can achieve the individual max-flow bound. (This result is somewhat stronger than the one in [1]. Please refer to the example in Section III.) More explicitly, we treat a block of data as a vector over a certain base field and allow a node to apply a linear transformation to a vector before passing it on. A preliminary version of this paper has appeared in the conference proceedings [3].

The remainder of the paper is organized as follows. In Section II, we introduce the basic notions, in particular, the notion of a *linear-code multicast* (LCM). In Section III, we show that with a "generic" LCM, every node can simultaneously receive information from the source at rate equal to its max-flow bound. In Section IV, we describe the physical implementation of an LCM first when the network is acyclic and then when the network is cyclic. In Section V, we present a greedy algorithm for constructing a generic LCM for an acyclic network. The same algorithm can be applied to a cyclic network by expanding the network into an acyclic network. This results in a "time-varying" LCM, which, however, requires high complexity in implementation. In Section VI, we introduce the time-invariant LCM (TILCM) and present a heuristic for constructing a generic TILCM. Section VII presents concluding remarks.

II. Basic Notions

In this section, we first introduce some graph-theoretic terminology and notations which will be used throughout the paper. Then we will introduce the notion of an LCM, an abstract algebraic description of a linear code on a communication network.

Convention: The generic notation for a nonsource node will be T, U, W, X, Y, or Z. The notation XY will stand for any channel from X to Y.

Definition: Over a communication network a *flow* from the source to a nonsource node T is a collection of channels, to be called the *busy* channels in the flow, such that

1) the subnetwork defined by the busy channels is acyclic, i.e., the busy channels do not form directed cycles;
2) for any node other than S and T, the number of incoming busy channels equals the number of outgoing busy channels;
3) the number of outgoing busy channels from S equals the number of incoming busy channels to T.

In other words, a flow is an *acyclic* collection of channels that abides by the law of commodity flow. The number of outgoing busy channels from S will be called the *volume* of the flow. The node T is called the *sink* of the flow. All the channels on the communication network that are not busy channels of the flow are called the *idle* channels with respect to the flow.

Proposition 2.1: The busy channels in a flow with volume f can be partitioned into f simple paths from the source to the sink.

The proof of this proposition is omitted.

Notation: For every nonsource node T on a network (G, S), the maximum volume of a flow from the source to T is denoted as $\mathrm{maxflow}_G(T)$, or simply $\mathrm{maxflow}(T)$ when there is no ambiguity.

Definition: A *cut* on a communication network (G, S) between the source and a nonsource node T means a collection $\mathtt{C}$ of nodes which includes S but not T. A channel XY is said to be *in* the cut $\mathtt{C}$ if $X \in \mathtt{C}$ and $Y \notin \mathtt{C}$. The number of channels in a cut is called the *value* of the cut.

The well-known Max-Flow Min-Cut Theorem (see, for example, [2, Ch. 4, Theorem 2.3]) still applies despite the acyclic restriction in the definition of a flow.

Max-Flow Min-Cut Theorem: For every nonsource node T, the minimum value of a cut between the source and a node T is equal to $\mathrm{maxflow}(T)$.

We are now ready to define a linear code multicast.

Notation: Let d be the maximum of $\mathrm{maxflow}(T)$ over all T. Throughout Sections II–V, the symbol Ω will denote a fixed d-dimensional vector space over a sufficiently large base field.

Convention: The information unit is taken as a symbol in the base field. In other words, 1 symbol in the base field can be transmitted on a channel every unit time.

Definition: A *linear-code multicast (LCM)* v on a communication network (G, S) is an assignment of a vector *space* $v(X)$ to every node X and a vector $v(XY)$ to every channel XY such that

1) $v(S) = \Omega$;
2) $v(XY) \in v(X)$ for every channel XY;
3) for any collection $\wp$ of nonsource nodes in the network

$$\langle\{v(T)\colon T \in \wp\}\rangle = \langle\{v(XY)\colon X \notin \wp,\ Y \in \wp\}\rangle.$$

The notation $\langle\bullet\rangle$ is for *linear span*. Condition 3) says that the vector spaces $v(T)$ on all nodes T inside $\wp$ together have the same linear span as the vectors $v(XY)$ on all channels XY to nodes in $\wp$ from outside of $\wp$. Conditions 2) and 3) show that an LCM abides by the law of information flow stated in Section I. The vector $v(XY)$ assigned to every channel XY may be identified with a d-dimensional *column* vector over the base field of Ω by choosing a basis for Ω.

Applying Condition 3) to the collection of a single nonsource node T, the space $v(T)$ is linearly spanned by vectors $v(XT)$

on all incoming channels XT to T. This shows that an LCM on a communication network is completely determined by the vectors it assigns to the channels. Together with Condition 2), we have

4) The vector assigned to an outgoing channel from T must be a linear combination of the vectors assigned to the incoming channels of T.

Condition 4) may be regarded as the "*law of information flow at a node*." However, unlike in network flow theory, the law of information flow being observed for every single node does not necessarily imply it being observed for every *set* of nodes when the network contains a directed cycle. A counterexample will appear in Section IV.

An LCM v specifies a mechanism for data transmission over the network as follows. We encode the information to be transmitted from S as a d-dimensional *row* vector, which we shall call the *information vector*. Under the transmission mechanism prescribed by the LCM v, the data flowing on a channel XY is the matrix product of the information (row) vector with the (column) vector $v(XY)$. In this way, the vector $v(XY)$ acts as the kernel in the linear encoder for the channel XY. As a direct consequence of the definition of an LCM, the vector assigned to an outgoing channel from a node X is a linear combination of the vectors assigned to the incoming channels to X. Consequently, the data sent on an outgoing channel from a node X is a linear combination of the data sent on the incoming channels to X.

Under this mechanism, the amount of information reaching a node T is given by the dimension of the vector space $v(T)$ when the LCM v is used. The physical realization of this mechanism will be discussed in Section IV.

Example 2.2: Consider the multicast of two bits, b_1 and b_2, from S to Y and Z in the communication network in Fig. 1(b). This is achieved with the LCM v specified by

$$v(ST) = v(TW) = v(TY) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$v(SU) = v(UW) = v(UZ) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and

$$v(WX) = v(XY) = v(XZ) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The data sent on a channel is the matrix product of the *row* vector $(b_1 \;\; b_2)$ with the *column* vector assigned to that channel by v. For instance, the data sent on the channel WX is $b_1 + b_2$. Note that, in the special case when the base field of Ω is GF(2), the vector $b_1 + b_2$ reduces to the exclusive-OR $b_1 \oplus b_2$ in an earlier example.

Proposition 2.3: For every LCM v on a network, for all nodes T

$$\dim(v(T)) \leq \text{maxflow}(T).$$

Proof: Fix a nonsource node T and any cut $\mathtt{C}$ between the source and T

$$v(T) \subset \langle v(Z)\colon Z \notin \mathtt{C}\rangle = \langle v(YZ)\colon Y \in \mathtt{C} \text{ and } Z \notin \mathtt{C}\rangle.$$

Hence, $\dim(v(T)) \leq \dim(\langle v(YZ)\colon Y \in \mathtt{C} \text{ and } Z \notin \mathtt{C}\rangle)$, which is at most equal to the value of the cut. In particular, $\dim(v(T))$ is upper-bounded by the minimum value of a cut between S and T, which by the Max-Flow Min-Cut Theorem is equal to $\text{maxflow}(T)$. □

This corollary says that $\text{maxflow}(T)$ is an upper bound on the amount of information received by T when an LCM v is used.

III. Achieving the Max-Flow Bound Through a Generic LCM

In this section, we derive a sufficient condition for an LCM v to achieve the max-flow bound on $\dim(v(T))$ in Proposition 2.3.

Definition: An LCM v on a communication network is said to be *generic* if the following condition holds for any collection of channels X_1Y_1, X_2Y_2, …, X_mY_m for $1 \leq m \leq d$: $(*)v(X_k) \not\subset \langle\{v(X_jY_j)\colon j \neq k\}\rangle$ for $1 \leq k \leq m$ if and only if the vectors $v(X_1Y_1)$, $v(X_2Y_2)$, …, $v(X_mY_m)$ are linearly independent.

If $v(X_1Y_1)$, $v(X_2Y_2)$, …, $v(X_mY_m)$ are linearly independent, then $v(X_k) \not\subset \langle\{v(X_jY_j)\colon j \neq k\}\rangle$ since $v(X_kY_k) \in v(X_k)$. A generic LCM requires that the converse is also true. In this sense, a generic LCM assigns vectors which are as linearly independent as possible to the channels.

Example 3.1: With respect to the communication network in Fig. 1(b), the LCM v in Example 2.2 is a generic LCM. However, the LCM u defined by

$$u(ST) = u(TW) = u(TY) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$u(SU) = u(UW) = u(UZ) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and

$$u(WX) = u(XY) = u(XZ) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

is not generic. This is seen by considering the set of channels $\{ST, WX\}$ where

$$u(S) = u(W) = \left\langle \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle.$$

Then $u(S) \not\subset \langle u(WX)\rangle$ and $u(W) \not\subset \langle u(ST)\rangle$, but $u(ST)$ and $u(WX)$ are not linearly independent. Therefore, u is not generic.

Lemma 3.2: Let v be a generic LCM. Any collection of channels XY_1, XY_2, …, XY_m from a node X with $m \leq \dim(v(X))$ must be assigned linearly independent vectors by v.

Theorem 3.3: If v is a generic LCM on a communication network, then for all nodes T

$$\dim(v(T)) = \text{maxflow}(T).$$

Proof: Consider a node T not equal to S. Let f be the common value of $\text{maxflow}(T)$ and the minimum value of a cut between S and T. The inequality $\dim(v(T)) \leq f$

follows from Proposition 2.3. Thus, we only have to show that $\dim(v(T)) \geq f$.

Let $\dim(C) = \dim(\langle v(X, Y)\colon X \in \mathtt{C}$ and $Y \notin \mathtt{C}\rangle)$ for any cut $\mathtt{C}$ between S and T. We will show that $\dim(v(T)) \geq f$ by contradiction. Assume $\dim(v(T)) < f$ and let $\mathtt{A}$ be the collection of cuts $\mathtt{U}$ between S and T such that $\dim(\mathtt{U}) < f$. Since $\dim(v(T)) < f$ implies $\mathtt{V}\backslash\{T\} \in \mathtt{A}$, where $\mathtt{V}$ is the set of all the nodes in G, $\mathtt{A}$ is nonempty.

By the assumption that v is a generic LCM, the number of edges out of S is at least d, and $\dim(\{S\}) = d \geq f$. Therefore, $\{S\} \notin \mathtt{A}$. Then there must exist a minimal member $\mathtt{U} \in \mathtt{A}$ in the sense that for any $Z \in \mathtt{U}\backslash\{S\} \neq \phi$, $\mathtt{U}\backslash\{Z\} \notin \mathtt{A}$. Clearly, $\mathtt{U} \neq \{S\}$ because $\{S\} \notin \mathtt{A}$.

Let $\mathtt{K}$ be the set of channels in cut $\mathtt{U}$ and $\mathtt{B}$ be the set of boundary nodes of $\mathtt{U}$, i.e., $Z \in \mathtt{B}$ if and only if $Z \in \mathtt{U}$ and there is a channel (Z, Y) such that $Y \notin \mathtt{U}$. Then for all $W \in \mathtt{B}$,

$$v(W) \not\subset \langle v(X, Y)\colon (X, Y) \in \mathtt{K}\rangle$$

which can be seen as follows. The set of channels in cut $\mathtt{U}\backslash\{W\}$ but not in $\mathtt{K}$ is given by $\{(X, W)\colon X \in \mathtt{U}\backslash\{W\}\}$. Since v is an LCM

$$\langle v(X, W)\colon X \in \mathtt{U}\backslash\{W\}\rangle \subset v(W).$$

If $v(W) \subset \langle v(X, Y)\colon (X, Y) \in \mathtt{K}\rangle$, then

$$\langle v(X', Y')\colon X' \in \mathtt{U}\backslash\{W\}, Y' \notin \mathtt{U}\backslash\{W\}\rangle$$

the subspace spanned by the channels in cut $\mathtt{U}\backslash\{W\}$, is contained by $\langle v(X, Y)\colon (X, Y) \in \mathtt{K}\rangle$. This implies

$$\dim(\mathtt{U}\backslash\{W\}) \leq \dim(\mathtt{U}) < f$$

a contradiction.

Therefore, for all $W \in \mathtt{B}$, $v(W) \not\subset \langle v(X, Y)\colon (X, Y) \in \mathtt{K}\rangle$.

For all $(W, Y) \in \mathtt{K}$, since

$$\langle v(X, Z)\colon (X, Z) \in \mathtt{K}\backslash\{(W, Y)\}\rangle \subset \langle v(X, Y)\colon (X, Y) \in \mathtt{K}\rangle, v(W) \not\subset \langle v(X, Y)\colon (X, Y) \in \mathtt{K}\rangle$$

implies

$$v(W) \not\subset \langle v(X, Z)\colon (X, Z) \in \mathtt{K}\backslash\{(W, Y)\}\rangle.$$

Then, by the definition of a generic LCM, $\{v(XY)\colon (X,Y)\in\mathtt{K}\}$ is a collection of vectors such that $\dim(\mathtt{U}) = \min(|\mathtt{K}|, d)$. Finally, by the Max-Flow Min-Cut Theorem, $|\mathtt{K}| \geq f$, and since $d \geq f$, $\dim(\mathtt{U}) \geq f$. This is a contradiction to the assumption that $\mathtt{U} \in A$. The theorem is proved. □

An LCM for which $\dim(v(T)) = \text{maxflow}(T)$ for all T provides a way for broadcasting a message generated at the source S for which every nonsource node T receives the message at rate equal to $\text{maxflow}(T)$. This is illustrated by the next example, which is based upon the assumption that the base field of Ω is an infinite field or a sufficiently large finite field. In this example, we employ a technique which is justified by the following lemma.

Lemma 3.4: Let X, Y and Z be nodes such that $\text{maxflow}(X) = i$, $\text{maxflow}(Y) = j$, and $\text{maxflow}(Z) = k$, where $i \leq j$ and $i > k$. By removing any edge UX in the graph, $\text{maxflow}(X)$ and $\text{maxflow}(Y)$ are reduced by at most 1, and $\text{maxflow}(Z)$ remains unchanged.

Proof: By removing an edge UX, the value of a cut $\mathtt{C}$ between the source S and node X (respectively, node Y) is reduced by 1 if edge UX is in $\mathtt{C}$, otherwise, the value of $\mathtt{C}$ is unchanged. By the Max-Flow Min-Cut Theorem, we see that $\text{maxflow}(X)$ and $\text{maxflow}(Y)$ are reduced by at most 1 when edge UX is removed from the graph. Now consider the value of a cut $\mathtt{C}$ between the source S and node Z. If $\mathtt{C}$ contains node X, then edge UX is not in $\mathtt{C}$, and, therefore, the value of $\mathtt{C}$ remains unchanged upon the removal of edge UX. If $\mathtt{C}$ does not contain node X, then $\mathtt{C}$ is a cut between the source S and node X. By the Max-Flow Min-Cut Theorem, the value of $\mathtt{C}$ is at least i. Then upon the removal of edge UX, the value of $\mathtt{C}$ is lower-bounded by $i - 1 \geq k$. Hence, by the Max-Flow Min-Cut Theorem, $\text{maxflow}(Z)$ remains to be k upon the removal of edge UX. The lemma is proved.

Example 3.5: Consider a communication network for which $\text{maxflow}(T) = 4, 3$, or 1 for nodes T in the network. The source S is to broadcast 12 symbols $a_1, \ldots, a_{12}$ taken from a sufficiently large base field F. (Note that 12 is the least common multiple of 4, 3, and 1.) Define the set

$$\Im_i = \{T\colon \text{maxflow}(T) = i\}, \qquad \text{for } i = 4, 3, 1.$$

For simplicity, we use the second as the time unit. We now describe how $a_1, \ldots, a_{12}$ can be broadcast to the nodes in $\Im_4$, $\Im_3$, $\Im_1$ in 3, 4, and 12 s, respectively, assuming the existence of an LCM on the network for $d = 4, 3, 1$.

a) Let v_1 be an LCM on the network with $d = 4$. Let

$$\alpha_1 = (a_1\ a_2\ a_3\ a_4)$$
$$\alpha_2 = (a_5\ a_6\ a_7\ a_8)$$

and

$$\alpha_3 = (a_9\ a_{10}\ a_{11}\ a_{12}).$$

In the first second, transmit α_1 as the information vector using v_1, in the second second, transmit α_2, and in the third second, transmit α_3. After 3 s, all the nodes in $\Im_4$ can recover α_1, α_2, and α_3. Throughout this example, we assume that all transmissions and computations are instantaneous.

b) Let r be a vector in F^4 such that $\langle\{r\}\rangle$ intersects trivially with $v_1(T)$ for all T in $\Im_3$, i.e., $\langle\{r, v_1(T)\}\rangle = F^4$ for all T in $\Im_3$. Such a vector r can be found when F is sufficiently large because there are a finite number of nodes in $\Im_3$. Define $b_i = \alpha_i r$ for $i = 1, 2, 3$. Now remove incoming edges of nodes in $\Im_4$, if necessary, so that $\text{maxflow}(T)$ becomes 3 if T is in $\Im_4$, otherwise, $\text{maxflow}(T)$ remains unchanged. This is possible by virtue of Lemma 3.4. Let v_2 be an LCM on the resulting network with $d = 3$. Let $\beta = (b_1\ b_2\ b_3)$ and transmit β as the information vector using v_2 in the fourth second. Then all the nodes in $\Im_3$ can recover β and hence α_1, α_2, and α_3.

c) Let s_1 and s_2 be two vectors in F^3 such that $\langle\{s_1, s_2\}\rangle$ intersects with $v_2(T)$ trivially for all T in $\Im_1$, i.e., $\langle\{s_1, s_2, v_2(T)\}\rangle = F^3$ for all T in $\Im_1$. Define $\gamma_i = \beta s_i$ for $i = 1, 2$. Now remove incoming edges of nodes in $\Im_4$ and $\Im_3$, if necessary, so that $\text{maxflow}(T)$ becomes 1

if T is in $\Im_4$ or $\Im_3$, otherwise, $\text{maxflow}(T)$ remains unchanged. Again, this is possible by virtue of Lemma 3.4. Now let v_3 be an LCM on the resulting network with $d = 1$. In the fifth and the sixth seconds, transmit γ_1 and γ_2 as the information vectors using v_3. Then all the nodes in $\Im_1$ can recover β.

d) Let t_1 and t_2 be two vectors in F^4 such that $\langle\{t_1, t_2\}\rangle$ intersects with $\langle\{r, v_1(T)\}\rangle$ trivially for all T in $\Im_1$, i.e., $\langle\{t_1, t_2, r, v_1(T)\}\rangle = F^4$ for all T in $\Im_1$. Define $\delta_1 = \alpha_1 t_1$ and $\delta_2 = \alpha_1 t_2$. In the seventh and eighth seconds, transmit δ_1 and δ_2 as the information vectors using v_3. Since all the nodes in $\Im_1$ already knows b_1, upon receiving δ_1 and δ_2, α_1 can then be recovered.

e) Define $\delta_3 = \alpha_2 t_1$ and $\delta_4 = \alpha_2 t_2$. In the nineth and tenth seconds, transmit δ_3 and δ_4 as the information vectors using v_3. Then α_2 can be recovered by all the nodes in $\Im_1$.

f) Define $\delta_5 = \alpha_3 t_1$ and $\delta_6 = \alpha_3 t_2$. In the eleventh and twelveth seconds, transmit δ_5 and δ_6 as the information vectors using v_3. Then α_3 can be recovered by all the nodes in $\Im_1$.

Let us now give a summary of the preceding scheme. In the ith second for $i = 1, 2, 3$, via the generic LCM v_1, each node in $\Im_4$ receives all four dimensions of α_i, each node in $\Im_3$ receives three dimensions of α_i, and each node in $\Im_1$ receives one dimension of α_i. In the fourth second, via the generic LCM v_2, each node in $\Im_3$ receives the vector β, which provides the three missing dimensions of α_1, α_2, and α_3 (one dimension for each) during the first 3 s of multicast by v_1. At the same time, each node in $\Im_1$ receives one dimension of β. Now, in order to recover β, each node in $\Im_1$ needs to receive the two missing dimensions of β during the fourth second. This is achieved by the generic LCM v_3 in the fifth and sixth seconds. So far, each node in $\Im_1$ has received one dimension of α_i for $i = 1, 2, 3$ via v_1 during the first 3 s, and one dimension of α_i for $i = 1, 2, 3$ from β via v_2 and v_3 during the fourth to sixth seconds. Thus, it remains to provide the six missing dimensions of α_1, α_2, and α_3 (two dimensions for each) to each node in $\Im_1$, and this is achieved in the seventh to the twelveth seconds via the generic LCM v_3.

Remark: The scheme in Example 3.5 can readily be generalized to arbitrary sets of max-flow values. The details are omitted here.

In the scheme in Example 3.5, at the end of the 12-s session, each node receives a message of 12 symbols taken from the base field F. Thus, the average information rate arriving at each node over the whole session is 1 symbol/s. The result in [1] asserts that this rate, which is the minimum of the individual max-flow bounds of all the nodes in the network, can be achieved. However, it is seen in our scheme that the nodes in $\Im_4$, $\Im_3$, $\Im_1$ can actually receive the whole message in the first 3, 4, and 12 s, respectively. In this sense, each node in our scheme can receive the message at rate equal to its individual max-flow bound. Thus, our result is somewhat stronger than that in [1]. However, our result does not mean that information can be multicast continually from the source to each node at rate equal to its individual max-flow bound.

If it is not necessary for the nodes in $\Im_1$ to receive the message, i.e., the message is multicast to the nodes in $\Im_4$ and $\Im_3$ only, the session can be terminated after 4 s. Then the average information rate arriving at each node in $\Im_4$ and $\Im_3$ over the truncated session is 3 symbols/s, with the nodes in $\Im_4$ and $\Im_3$ receiving the whole message in the first 3 and 4 s, respectively.

IV. The Transmission Scheme Associated With an LCM

Let v be an LCM on a communication network (G, S), where the vectors $v(SX)$ assigned to outgoing channels SX linearly span a d-dimensional space. As before, the vector $v(XY)$ assigned to a channel XY is identified with a d-dimensional column vector over the base field of Ω by means of the choice of a basis. On the other hand, the total information to be transmitted from the source to the rest of the network is represented by a d-dimensional row vector, called the information vector. Under the transmission scheme prescribed by the LCM v, the data flowing over a channel XY is the matrix product of the information vector with the column vector $v(XY)$. We now consider the physical realization of this transmission scheme associated with an LCM.

Definition: A communication network (G, S) is said to be *acyclic* if the directed multigraph G does not contain a directed cycle.

Lemma 4.1: An LCM v on an acyclic network (G, S) is an assignment of a vector space $v(X)$ to every node X and a vector $v(XY)$ to every channel XY such that

1) $v(S) = \Omega$;
2) $v(XY) \in v(X)$ for every channel XY;
3) for every nonsource node T, the space $v(T)$ is the linear span of vectors $v(XT)$ on all incoming channels XT to T.

In other words, for an acyclic network, the law of information flow being observed for every single node implies it being observed for every *set* of nodes.

Proof: Let $\wp$ be a set of nonsource nodes on an acyclic network. Let an edge XY be called an internal edge of $\wp$ when $X \in \wp$ and $Y \in \wp$. Similarly, let an edge XY be called an incoming edge of $\wp$ when $X \notin \wp$ and $Y \in \wp$. We need to show that, for every internal edge UZ of $\wp$, $v(UZ)$ is generated by

$$\{v(XY)\colon XY \text{ is an incoming edge of } \wp\}.$$

Because the network is acyclic, there exists a node T in $\wp$ without any edge TX, where $X \in \wp$. Thus,

1) every incoming edge of $\wp\backslash\{T\}$ is an incoming edge of $\wp$.

By induction on $|\wp|$, we may assume that for every internal edge UZ of $\wp\backslash\{T\}$

2) $v(UZ)$ is generated by $\{v(XY)\colon XY$ is an incoming edge of $\wp\backslash\{T\}\}$.

Given an internal edge WT of $\wp$, we need to show that $v(WT)$ is generated by $\{v(XY)\colon XY$ is an incoming edge of $\wp\}$. Since $v(WT)$ is generated by

$$\{v(QW)\colon QW \text{ is an edge}\}$$

it suffices to show that $v(QW)$ is generated by $\{v(XY)$: XY is an incoming edge of $\wp\}$.

If the edge QW is incoming of $\wp\backslash\{T\}$, then it is incoming to $\wp$ by 1). Otherwise, $v(QW)$ is generated by

$$\{v(XY)\text{: } XY \text{ is an incoming edge of } \wp\backslash\{T\}\}$$

according to 2) and, therefore, is also generated by

$$\{v(XY)\text{: } XY \text{ is an incoming edge of } \wp\}$$

because of 1). □

Lemma 4.2: The nodes on an acyclic communication network can be sequentially indexed such that every channel is from a smaller indexed node to a larger indexed node.

Lemma 4.3: Assume that nodes in a communication network are sequentially indexed as $X_0 = S, X_1, \ldots, X_n$ such that every channel is from a smaller indexed node to a larger indexed node. Then, every LCM on the network can be constructed by the following procedure:

```
{
  for (j = 0; j ≤ n; j++)
  {
    arrange all outgoing channels X_jY from X_j
      in an arbitrary order;
    take one outgoing channel from X_j at a time
    {
      let the channel taken be X_jY;
      assign v(X_jY) to be a vector in the space
        v(X_j);
    }
    v(X_{j+1}) = linear span by vectors v(XX_{j+1}) on
      all incoming channels XX_{j+1} to X_{j+1};
  }
}
```

On an acyclic network, a straightforward realization of the above transmission scheme is as follows. Take one node at a time according to the sequential indexing. For each node, "wait" until data is received from every incoming channel before performing the linear encoding. Then send the appropriate data on each outgoing channel.

This physical realization of an LCM over an acyclic network, however, does not apply to a network that contains a directed cycle. This is illustrated by the following example.

Example 4.4: Let p, q, and r be vectors in Ω, where p and q are linearly independent. Define

$$v(SX) = p, \qquad v(SY) = q$$

and

$$v(WX) = v(XY) = v(YW) = r.$$

This specifies an LCM v on the network illustrated in Fig. 2 if the vector r is a linear combination of p and q. Otherwise, the function v gives an example in which the law of information flow is observed for every single node but not observed for

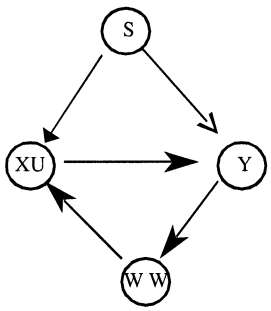

Fig. 2. An LCM on a cyclic network.

every set of nodes. Specifically, the law of information flow is observed for each of the nodes X, Y, and W, but not for the set of nodes $\{X, Y, W\}$.

Now, assume that

$$p = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad q = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \text{and} \quad r = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Then, v is an LCM. Write the information vector as $(b_1 \quad b_2)$, where b_1 and b_2 belong to the base field of Ω. According to the transmission scheme associated with the LCM, all three channels on the directed cycle transmit the same data $b_1 + b_2$. This leads to the logical problem of how any of these cyclic channels acquires the data $b_1 + b_2$ in the first place.

In order to realize the transmission scheme associated with an LCM over a network containing a directed cycle, we shall introduce the parameter of time into the scheme. Instead of transmitting a single data symbol (i.e., an element of the base field of Ω) through each channel, we shall transmit a time-parameterized stream of symbols. In other words, the channel will be time-slotted. Concomitantly, the operation of coding at a node will be time-slotted, as well.

Definition: Given a communication network (G, S) and a positive integer τ, the associated *memoryless* communication network denoted as $(G^{(\tau)}, S)$ is defined as follows. The set of nodes in $G^{(\tau)}$ includes the node S and all the pairs of the type $[X, t]$, where X is a nonsource node in G and t ranges through integers 1 to τ. The channels in the network $(G^{(\tau)}, S)$ belong to one of the three types listed below. For any nonsource nodes X and Y in (G, S)

1) for $t \leq \tau$, the multiplicity of the channel from S to $[X, t]$ is the same as that of the channel SX in the network (G, S);
2) for $t < \tau$, the multiplicity of the channel from $[X, t]$ to $[Y, t+1]$ is the same as that of the channel XY in the network (G, S);
3) for $t < \tau$, the multiplicity of the channel from $[X, t]$ to $[X, \tau]$ is equal to $\mathrm{maxflow}_G(X)$.

Lemma 4.5: The memoryless communication network $(G^{(\tau)}, S)$ is acyclic.

Lemma 4.6: There exists a fixed number ε, independent of τ, such that for all nonsource nodes X in (G, S), the maximum volume of a flow from S to the node $[X, \tau]$ in $(G^{(\tau)}, S)$ is at least $\tau - \varepsilon$ times $\mathrm{maxflow}_G(X)$.

Proof: Given a nonsource node X in G, consider a maximum flow from S to X in the network (G, S). This maximum flow may be partitioned into directed simple paths from S to X. Let ε_X be the maximum length among these directed simple paths. Then, there exists a flow from S to the node $[X, \varepsilon_X]$ in $(G^{(\tau)}, S)$ that is isomorphic to the maximum flow in (G, S). Moreover, time-shifted isomorphs of this flow in $(G^{(\tau)}, S)$ are flows from S to $[X, t]$ for $\varepsilon_X \leq t \leq \tau$. Moreover, the isomorphs are edge-disjoint flows in the network $(G^{(\tau)}, S)$ due to the acyclic nature in the definition of a flow. The union of these flows, together with edges from $[X, t]$ to $[X, \tau]$ for $t < \tau$ in $(G^{(\tau)}, S)$, constitute a flow from S to $[X, \tau]$ with volume $\tau - \varepsilon_X$ times $\mathrm{maxflow}_G(X)$. The proof is completed by choosing ε to be the largest ε_X minus 1 among all X. □

Transmission of data symbols over the network $(G^{(\tau)}, S)$ may be interpreted as "*memoryless*" transmission of data *streams* over the network (G, S) as follows.

1) A symbol sent from S to $[X, t]$ in $(G^{(\tau)}, S)$ corresponds to the symbol sent on the channel SX in (G, S) during the time slot t.

2) A symbol sent from $[X, t]$ to $[Y, t+1]$ in $(G^{(\tau)}, S)$ corresponds to the symbol sent on the channel XY in (G, S) during the time slot $t+1$. This symbol is a linear combination of symbols received by X during the time slot t and is unrelated to symbols received earlier by X.

3) The channels from $[X, t]$ to $[X, \tau]$ for $t < \tau$ signify the accumulation of received information by the node X in (G, S) over time.

Now suppose an LCM has been constructed on the network $(G^{(\tau)}, S)$. Since this is an acyclic network, the LCM can be physically realized in the way mentioned above. The physical realization can then be interpreted as a memoryless transmission of data streams over the original network (G, S).

Transmission, with memory, of data streams over the original network (G, S) is associated with the acyclic network defined below, which is just a slight modification from $(G^{(\tau)}, S)$.

Definition: Given a communication network (G, S) and a positive integer τ, the associated communication network with memory, denoted as $(G^{[\tau]}, S)$, is defined as follows. The set of nodes in $G^{[\tau]}$ includes the node S and all pairs of the type $[X, t]$, where X is a nonsource node in G and t ranges through integers 1 to τ. Channels in the network $(G^{[\tau]}, S)$ belong to one of the three types listed below. For any nonsource nodes X and Y in (G, S)

1) for $t \leq \tau$, the multiplicity of the channel from S to $[X, t]$ is the same as that of the channel SX in the network (G, S);

2) for $t < \tau$, the multiplicity of the channel from $[X, t]$ to $[Y, t+1]$ is the same as that of the channel XY in the network (G, S);

3) for $t < \tau$, the multiplicity of channels from $[X, t]$ to $[X, t+1]$ is equal to t times $\mathrm{maxflow}_G(X)$.

Lemma 4.7: The communication network $(G^{[\tau]}, S)$ is acyclic.

Lemma 4.8: Every flow from the source to the node X in the network $(G^{(\tau)}, S)$ corresponds to a flow with the same volume from the source to the node $[X, \tau]$ in the network $(G^{[\tau]}, S)$.

Lemma 4.9: Every LCM v on the network $(G^{(\tau)}, S)$ corresponds to an LCM u on the network $(G^{[\tau]}, S)$ such that for all nodes X in G

$$\dim(u([X, \tau])) = \dim(v(X)).$$

Communication networks with or without memory both have been defined in order to compensate for the lack of a direct physical realization of an LCM on a network that may contain a directed cycle. In Section VI, we shall present another way to make the compensation, which will offer a physical realization by "*time-invariant*" linear coding.

V. Construction of a Generic LCM on an Acyclic Communication Network

We have proved that for a generic LCM, the dimension of the vector space at each node T is equal to the maximum flow between S and T. However, we have not shown how one can construct a generic LCM for a given communication network. In the next theorem, we present a procedure which constructs a generic LCM for any acyclic communication network.

Theorem 5.1: A generic LCM exists on every acyclic communication network, provided that the base field of Ω is an infinite field or a large enough finite field.

Proof: Let the nodes in the acyclic network be sequentially indexed as $X_0 = S, X_1, X_2, \ldots, X_n$ such that every channel is from a smaller indexed node to a larger indexed node. The following procedure constructs an LCM by assigning a vector $v(XY)$ to each channel XY, one channel at a time.

```
{
  for all channels XY
    v(XY) = the zero vector; // initialization
  for (j = 0; j ≤ n; j++)
  {
    arrange all outgoing channels X_jY from X_j
      in an arbitrary order;
    take one outgoing channel from X_j at a time
    {
      let the channel taken be X_jY;
      choose a vector w in the space v(X_j) such
      that w ∉ ⟨v(UZ): UZ ∈ ξ⟩ for any collection
      ξ of at most d − 1 channels with
                    v(X_j) ∉ ⟨v(UZ): UZ ∈ ξ⟩;
      v(X_jY) = w;
    }
      v(X_{j+1}) = the linear span by vectors
      v(XX_{j+1}) on all incoming channels XX_{j+1}
      to X_{j+1};
  }
}
```

The essence of the above procedure is to construct the generic LCM iteratively and make sure that in each step the partially constructed LCM is generic. One point in the above construction procedure of v needs to be clarified. Given a node X_j, there can be only finitely many collections ξ of cardinality at most d such that $v(X_j) \not\subset \langle v(UZ)\colon UZ \in \xi\rangle$. When the base field of Ω is large enough

$$v(X_j) \not\subset \cup_{\xi}\langle v(UZ)\colon UZ \in \xi\rangle$$

where the union is over all such ξ. Hence, the choice of the vector w in the above procedure has been possible.

Let $Z_1Y_1, Z_2Y_2, \ldots, Z_mY_m$ be any collection of channels such that

$$v(Z_k) \not\subset \langle\{v(Z_jY_j)\colon j \neq k\}\rangle$$

for $1 \leq k \leq m$ and $1 \leq m \leq d$. In order to assert that the LCM v is generic, we need to prove the linear independence among the vectors $v(Z_1Y_1), v(Z_2Y_2), \ldots, v(Z_mY_m)$. The proof is by induction on m. Without loss of generality, we may assume that Z_mY_m is the last among the m channels to be assigned a vector in the above construction procedure of v. Since

$$v(Z_m) \not\subset \langle\{v(Z_jY_j)\colon 1 \leq j < m\}\rangle$$

the construction procedure gives

$$v(Z_mY_m) \notin \langle\{v(Z_jY_j)\colon 1 \leq j < m\}\rangle.$$

On the other hand, the induction hypothesis asserts the linear independence among $v(Z_1Y_1), v(Z_2Y_2), \ldots, v(Z_{m-1}Y_{m-1})$. Thus, the vectors $v(Z_1Y_1), v(Z_2Y_2), \ldots, v(Z_mY_m)$ are linearly independent. The theorem is proved. □

Given a communication network (G, S) and a positive integer τ, there exists a generic LCM v on the associated memoryless communication network $(G^{(\tau)}, S)$ by Lemma 4.5 and Theorem 5.1. From Theorem 3.3, for every node X in (G, S), the dimension of $v([X, \tau])$ for the node $[X, \tau]$ in $(G^{(\tau)}, S)$ is equal to the maximum volume of a flow from the source to $[X, \tau]$. This maximum volume, according to Lemma 4.6, is at least $\tau - \varepsilon$ times $\mathrm{maxflow}_G(X)$, where ε is a fixed integer.

In view of Lemmas 4.7–4.9, we have the following similar conclusion about the adapted communication network. For every node X in (G, S), the dimension of the space $v([X, \tau])$ for the corresponding node $[X, \tau]$ in $(G^{[\tau]}, S)$ is equal to the maximum volume of a flow from the source to $[X, \tau]$ in $(G^{[\tau]}, S)$, and this maximum volume is at least $\tau - \varepsilon$ times $\mathrm{maxflow}_G(X)$ for some fixed number ε.

We now translate this conclusion about linear-code multicast over the memoryless network back to the original network (G, S). Let f^* be the minimum of $\mathrm{maxflow}_G(X)$ over all nonsource nodes X in G. Then for a sufficiently large integer K, using the technique in Example 3.5, it is possible to design a broadcast session of length equal to K time units which broadcasts a message of approximately Kf^* symbols from the source. Moreover, the whole message can be recovered at each nonsource node X after approximately $Kf^*/\mathrm{maxflow}_G(X)$ time units.

VI. Time-Invariant LCM and Heuristic Construction

Let F be a finite field. Let $F((z))$ denote the ring of formal power series over F with the variable z. $F((z))$ is a *principal ideal domain*, and every ideal in it is generated by z^n for some n. Algebraic properties of vector spaces over a field can often be generalized to *modules* over a principle ideal domain. In fact, our previous results on LCM and generic LCM on vector spaces can readily be generalized to modules over $F((z))$.

The concept of the dimension of a vector space is generalized to the *rank* of a module. To facilitate our discussion, we shall refer to the rank of a module over $F((z))$ as its *dimension*. The elements of the module will be called *vectors*.

Throughout this section, we assume that Ω is a module over $F((z))$ with a finite but very large dimension (say, larger than the number of channels times the length of transmission stream in the problem instance). Physically, an element $a(z)$ of $F((z))$ is the z-transform of a stream of symbols $a_0, a_1, a_2, \ldots, a_t, \ldots$ that are sent on a channel, one symbol at a time. The formal variable z is interpreted as a unit-time shift.

Definition: A *time-invariant linear-code multicast (TILCM)* v on a communication network (G, S) is an assignment of a module $v(X)$ over $F((z))$ to every node X and a vector $v(XY)$ to every channel XY such that

1) $v(S) = \Omega$;
2) $v(XY) \in v(X)$ for every channel XY;
3) for any collection $\wp$ of nonsource nodes in the network
$$\langle\{v(T)\colon T \in \wp\}\rangle = \langle\{zv(XY)\colon X \notin \wp, Y \in \wp\}\rangle.$$

In particular, the vector assigned to an outgoing channel from a node is z *times* a linear combination of the vectors assigned to incoming channels to the same node. Hence a TILCM is completely determined by the vectors that it assigns to channels. The adoption of z times a linear combination, instead of simply any linear combination, allows a unit-time delay for transmission/coding. This allows a physical realization for the transmission scheme specified by a TILCM. If a certain physical system requires nonuniform duration of delay on different channels, then artificial nodes need to be inserted. For instance, if the delay on a channel is three unit times, then two serial nodes should be added on the channel so that the channel becomes a tandem of three channels.

Example 6.1: On the communication network illustrated in Fig. 2, define the TILCM v by

$$v(SX) = \begin{pmatrix}1\\0\end{pmatrix}, \quad v(SY) = \begin{pmatrix}0\\1\end{pmatrix}, \quad v(XY) = \begin{pmatrix}z\\z^3\end{pmatrix}$$

$$v(YW) = \begin{pmatrix}z^2\\z\end{pmatrix}, \quad \text{and} \quad v(WX) = \begin{pmatrix}z^3\\z^2\end{pmatrix}.$$

One can readily check that v is in fact a TILCM. For example

$$v(XY) = \begin{pmatrix}z\\z^3\end{pmatrix} = z(1-z^3)\begin{pmatrix}1\\0\end{pmatrix} + z\begin{pmatrix}z^3\\z^2\end{pmatrix}$$
$$= z[(1-z^3)v(SX) + v(WX)].$$

Thus, $v(XY)$ is equal to z times the linear combination of $v(SX)$ and $v(WX)$ with coefficients $1-z^3$ and 1, respectively. This specifies an encoding process for the channel XY that does not change with time. It can be seen that the same is true for the encoding process of every other channel in the network. This explains the terminology "*time-invariant*" for an LCM.

Write the information vector as $(a(z)\ \ b(z))$, where

$$a(z)=\sum_{j\geq 0}a_jz^j \quad \text{and} \quad b(z)=\sum_{j\geq 0}b_jz^j$$

belong to $F((z))$. The product of the information (row) vector with the (column) vector assigned to that channel represents the data stream transmitted over a channel

$$\begin{aligned}
&(a(z)\ \ b(z))\bullet v(SX)=a(z)\\
&\quad\rightarrow (a_0,\, a_1,\, a_2,\, a_3,\, a_4,\, a_5,\, \ldots,\, a_t,\, \ldots)\\
&(a(z)\ \ b(z))\bullet v(SY)=b(z)\\
&\quad\rightarrow (b_0,\, b_1,\, b_2,\, b_3,\, b_4,\, b_5,\, \ldots,\, b_t,\, \ldots)\\
&(a(z)\ \ b(z))\bullet v(XY)=za(z)+z^3b(z)\\
&\quad\rightarrow (0,\, a_0,\, a_1,\, a_2+b_0,\, a_3+b_1,\, a_4+b_2,\, \ldots,\\
&\qquad a_{t-1}+b_{t-3},\, \ldots)\\
&(a(z)\ \ b(z))\bullet v(YW)=z^2a(z)+zb(z)\\
&\quad\rightarrow (0,\, b_0,\, a_0+b_1,\, a_1+b_2,\, a_2+b_3,\, a_3+b_4,\, \ldots,\\
&\qquad a_{t-2}+b_{t-1},\, \ldots)\\
&(a(z)\ \ b(z))\bullet v(WX)=z^3a(z)+z^2b(z)\\
&\quad\rightarrow (0,\, 0,\, b_0,\, a_0+b_1,\, a_1+b_2,\, a_2+b_3,\, \ldots,\\
&\qquad a_{t-3}+b_{t-2},\, \ldots).
\end{aligned}$$

Adopt the convention that $a_t=b_t=0$ for all $t<0$. Then the data symbol flowing over the channel XY, for example, at the time slot t is $a_{t-1}+b_{t-3}$ for all $t\geq 0$.

Example 6.2: Another TILCM on the same communication network can be defined by

$$v(SX)=\begin{pmatrix}1\\0\end{pmatrix}, \qquad v(SY)=\begin{pmatrix}0\\1\end{pmatrix}$$

$$v(XY)=\frac{1}{1-z^3}\begin{pmatrix}z\\z^3\end{pmatrix}, \qquad v(YW)=\frac{1}{1-z^3}\begin{pmatrix}z^2\\z\end{pmatrix}$$

and

$$v(WX)=\frac{1}{1-z^3}\begin{pmatrix}z^3\\z^2\end{pmatrix}.$$

The data stream transmitted over the channel XY, for instance, is represented by

$$\begin{aligned}
&\left(\sum_{j\geq 0}a_jz^j \quad \sum_{j\geq 0}b_jz^j\right)\bullet v(XY)\\
&=\sum_{j\geq 0}\left(a_jz^{j+1}+b_jz^{j+3}\right)/\left(1-z^3\right)\\
&=\left[\sum_{j\geq 0}\left(a_jz^{j+1}+b_jz^{j+3}\right)\right]\sum_{i\geq 0}z^{3i}\\
&=\sum_{j\geq 0}\sum_{i\geq 0}\left(a_jz^{3i+j+1}+b_jz^{3i+j+3}\right)\\
&=\sum_{t\geq 1}z^t(a_{t-1}+a_{t-4}+a_{t-7}+\cdots+b_{t-3}+b_{t-6}+b_{t-9}+\cdots).
\end{aligned}$$

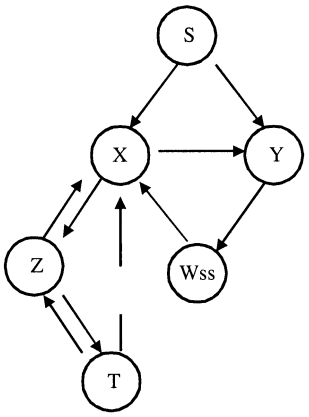

Fig. 3. Redundant channels in a network.

That is, the data symbol

$$a_{t-1}+a_{t-4}+a_{t-7}+\cdots+b_{t-3}+b_{t-6}+b_{t-9}+\cdots$$

is sent on the channel XY at the time slot t. This TILCM v, besides being time invariant in nature, is a "memoryless" one because the following linear equations allows an encoding mechanism that requires no memory:

$$v(XY)=zv(SX)+zv(WX)$$

$$v(YW)=zv(SY)+zv(XY)$$

and

$$v(WX)=zv(YW).$$

There are potentially various ways to define a *generic* TILCM and, as an analog to Theorem 3.3, to establish desirable dimensions of the module assigned to every node. Another desirable characteristic of a TILCM is to be "memoryless." Empirically, it has been relatively easy to construct a TILCM that carries all conceivable desirable properties.

The remainder of this section presents a heuristic construction procedure for a "good" TILCM. The heuristic construction will follow the graph-theoretical *block decomposition* of the network. For the sake of computational efficiency, the procedure will first remove "*redundant*" channels from the network before identifying the "blocks" so that the "blocks" are smaller.

Definition: A channel in a communication network is said to be *irredundant* if it is on a simple path starting at the source. Else, it is said to be *redundant*. Moreover, a communication network is said to be *irredundant* if it contains no redundant channels.

Example 6.3: In the network illustrated in Fig. 3, the channels ZX, TX, and TZ are redundant.

Lemma 6.4: The deletion of a redundant channel from a network results in a subnetwork with the same set of irredundant channels. Consequently, the irredundant channels in a network define an irredundant subnetwork.

Theorem 6.5: Let v be an LCM (respectively, a TILCM) on a network. Then v also defines an LCM (respectively, a TILCM) on the subnetwork that results from the deletion of any redundant channel.

Proof: We shall prove the case of LCM, the case of TILCM being similar. Denote the deleted redundant channel as XW. Given a set $\wp$ of nonsource nodes with $X \notin \wp$ and $W \in \wp$, we need to show

$$\langle v(YZ)\colon Y \notin \wp,\ Z \in \wp \text{ and } YZ \neq XW\rangle = \langle v(YZ)\colon Y \notin \wp \text{ and } Z \in \wp\rangle.$$

Due to the redundancy of XW, any simple path from S to X must go through the node W. Let $\wp_W$ denote the set consisting of the node W plus all those nodes Y such that any path from S to Y must go through W. (If there is no simple path from S to a particular node, then that node fully qualifies as a node in $\wp_W$.) Then, every channel from outside of $\wp_W$ into $\wp_W$ must be toward the node W. Therefore,

$$\begin{aligned}
&\langle v(YZ)\colon Y \notin \wp \cup \wp_W \text{ and } Z \in \wp \cup \wp_W\rangle\\
&\quad = \langle v(YZ)\colon Y \notin \wp \cup \wp_W \text{ and } Z \in \wp\rangle\\
&\quad \subset \langle v(YZ)\colon Y \notin \wp,\ Z \in \wp \text{ and } YZ \neq XW\rangle\\
&\quad \subset \langle v(YZ)\colon Y \notin \wp \text{ and } Z \in \wp\rangle\\
&\quad = \langle v(Z)\colon Z \in \wp\rangle\\
&\quad \subset \langle v(Z)\colon Z \in \wp \cup \wp_W\rangle\\
&\quad = \langle v(YZ)\colon Y \notin \wp \cup \wp_W \text{ and } Z \in \wp \cup \wp_W\rangle
\end{aligned}$$

This implies the desired equality. □

Corollary 6.6: Let v be an LCM (respectively, a TILCM) on a network. Then v defines an LCM (respectively, a TILCM) on the subnetwork formed by irredundant channels.

In a directed multigraph, an equivalence relationship (which by definition is *reflexive, symmetric* and *transitive*) among nodes can be defined as follows. Two nodes are equivalent if there exists a directed path leading from one node to the other and *vice versa*. An equivalence class under this relationship is called a *block* in the graph. The source node by itself always forms a block. When every block "*contracts*" into a single node, the resulting graph is acyclic. In other words, the blocks can be sequentially indexed so that every interblock channel is from a smaller indexed block to a larger indexed block.

For the construction of a "good" TILCM, smaller sizes of blocks tend to facilitate the computation. The extreme favorable case of the block decomposition of a network is when the network is acyclic, which implies that every block consists of a single node. The opposite extreme is when all nonsource nodes form a single block exemplified by the network illustrated in Fig. 3.

The removal of redundant channels sometimes serves for the purpose of breaking up a block into pieces. For the network illustrated in Fig. 3, the removal of the three redundant channels breaks the block $\{T, W, X, Y, Z\}$ into the three blocks $\{T\}$, $\{W, X, Y\}$, and $\{Z\}$.

In the construction of a "good" LCM on an *acyclic* network, the procedure inside the proof of Theorem 5.1 takes one node at a time according to the acyclic ordering of nodes and assigns vectors to outgoing channels from the taken node. For a general network, we can start with the trivial TILCM v on the network consisting of just the source and then expand it to a "good" TILCM v that covers one more block at a time.

The sequential choices of blocks are according to the acyclic order in the block decomposition of the network. Thus, the expansion of the "good" TILCM v at each step involves only *incoming* channels to nodes in the new block. A heuristic algorithm for assigning vectors $v(XY)$ to such channels XY is for $v(XY)$ to be z times an *arbitrary convenient* linear combination of vectors assigned to incoming channels to X. In this way, a system of linear equations of the form of $Ax = b$ is set up, where A is a square matrix with the dimension equal to the total number of channels in the network and x is the unknown column vector whose entries are $v(XY)$ for all channels XY. The elements of A and b are polynomials in z. In particular, the elements of A are either ± 1, 0, or a polynomial in z containing the factor z. Therefore, the determinant of A is a formal power series with the constant term (the zeroth power of z) being ± 1, and, hence is invertible in $F((z))$. According to Cramer's rule, a unique solution exists. (This is consistent with the physical intuition because the whole network is completely determined once the encoding process for each channel is specified.) If this unique solution does not happen to satisfy the requirement for being a "good" TILCM, then the heuristic algorithm calls for adjustments on the coefficients of the linear equations on the trial-and-error basis.

After a "good" TILCM is constructed on the subnetwork formed by irredundant channels in a given network, we may simply assign the zero vectors to all redundant channels.

Example 6.7: After the removal of redundant channels, the network depicted by Fig. 3 consists of four blocks in the order of $\{S\}$, $\{W, X, Y\}$, $\{Z\}$, and $\{T\}$. The subnetwork consisting of the first two blocks is the same as the network in Fig. 2. When we expand the trivial TILCM on the network consisting of just the source to cover the block $\{W, X, Y\}$, a heuristic trial would be

$$v(SX) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad v(SY) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

together with the following linear equations:

$$v(XY) = zv(SX) + zv(WX)$$

$$v(YW) = zv(SY) + zv(XY)$$

and

$$v(WX) = zv(YW).$$

The result is the memoryless TILCM v in the preceding example. This TILCM can be further expanded to cover the block $\{Z\}$ and then the block $\{T\}$.

VII. Concluding Remarks

In this paper, we have presented an explicit construction of a code for multicast in a network that achieves the max-flow bound on the information transmission rate. An important aspect of our code is its linearity, which makes encoding and decoding easy to implement in practice. Our greedy algorithm for code construction works for all networks, but the code

so constructed is not necessarily the simplest possible. In fact, there often exist optimal LCMs much simpler than the ones constructed according to the presented greedy algorithm. Therefore, there is much room for further research in this direction.

The code we have constructed for a cyclic network is time varying, which makes it less appealing in practice. To our knowledge, there has not been a proof for the existence of an optimal time-invariant code for a cyclic network. However, examples of such a code have been given in the present paper and also in [1]. Therefore, proving the existence of such codes, in particular, constructing such codes in simple linear form, is a challenging problem for future research.

Further research problems in network coding include code construction when two or more sources are simultaneously multicast in the network. This is the so-called multisource network coding problem in [1] which is yet unexplored (see also [5, Ch. 15]).

When networking coding is implemented in computer or satellite networks, synchronization is a problem that needs to be addressed. Since an encoder at an intermediate node may take more than one incoming data stream as input, it is necessary to acquire synchronization among these data streams. This is not a problem for nonreal-time applications (e.g., file transfer), but it can be a serious problem for real-time applications (e.g., voice and video transmission). On the other hand, certain types of networks, for example switching networks, are inherently fully synchronized. These networks are excellent candidates for network coding, and, in fact, the possible use of linear network codes in switching networks has been investigated [6].

ACKNOWLEDGMENT

The authors would like to thank Venkat Anantharam for pointing out an error in the original proof of Theorem 3.3 and for his very careful reading of the manuscript. They also thank the anonymous reviewers for their useful comments.

REFERENCES

[1] R. Alshwede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow: Single source," *IEEE Trans. Inform. Theory*, submitted for publication.

[2] E. L. Lawler, *Combinatorial Optimization: Network and Matroid*. Fort Worth, TX: Saunder College Pub., 1976.

[3] S.-Y. R. Li and R. W. Yeung, "Network multicast flow via linear coding," in *Proc. Int. Symp. Operational Research and Its Applications (ISORA'98)*, Kunming, China, Aug. 1998, pp. 197–211.

[4] D. J. A. Welsh, *Matroid Theory*. New York: Academic , 1976.

[5] R. W. Yeung, *A First Course in Information Theory*. Norwell, MA/New York: Kluwer/Plenum, 2002.

[6] F. R. Kschischang, private communication.