INVITED PAPER

Network Coding and Matroid Theory

This paper explores the connection between network coding and matroid theory, a branch of mathematics that generalizes linear algebra and graph theory.

By Randall Dougherty, Chris Freiling, and Kenneth Zeger, *Fellow IEEE*

ABSTRACT | Networks derived from matroids have played a fundamental role in proving theoretical results about the limits of network coding. In this tutorial paper, we review many connections between matroids and network coding theory, with specific emphasis on network solvability, admissible network alphabet sizes, linear coding, and network capacity.

KEYWORDS | Capacity; flow; information theory; matroid; multicast; network coding; polymatroid

I. INTRODUCTION

The field of network coding was essentially born with the foundational paper by Ahlswede *et al.* [2] in 2000. The main idea in network coding is to allow each node of a network to combine together data from its in-edges in order to determine what data to transmit on its out-edges. This concept contrasts with the traditional operations of packet-switched networks, such as the Internet, in which each node must relay data (i.e., using "routing") from selected in-edges to selected out-edges. The paper [2] demonstrated that network coding can in general be strictly superior to network routing in terms of achievable throughput rates.

During the decade following the publication of [2], network coding research has been focused in two general directions: 1) applications of network coding to practical networks, and 2) understanding the theoretical possibilities and limitations of network coding. In this paper, we focus on the *theory* of network coding, specifically on the connection between matroids and network coding.

Matroid theory is a branch of mathematics, founded in 1935 by Whitney [38], that generalizes many concepts in both linear algebra and graph theory and has some close connections with the information-theoretic notion of entropy. At the heart of matroid theory is the abstraction of "independence relation." This is motivated by various notions of independence throughout mathematics, such as linear independence in vector spaces or the acyclic property in graph theory.

Network coding also contains a notion of dependence. For a particular network node n, there are four types of data to deal with:

- data from packets received along the in-edges of n;
- data from messages that originate at n;
- data from packets sent along out-edges of n;
- data from messages demanded by n.

We require that data of the last two types be produced by deterministic functions of the first two types. So we say, for example, that the incoming packets, the originated messages, and any outgoing packet form a *dependent set*.

By connecting this network form of dependence with the matroid definition of dependence, a general method of constructing networks from matroids has been developed. Using this method, several interesting and well-known examples of matroids have been turned into networks that exhibit similar properties. This has increased our knowledge about the limitations of network coding. In addition, many large classes of networks studied in the literature turn out to be constructible from matroids, even though many of them have been discovered with nonmatroid methods.

In what follows, we review the main concepts of network coding and of matroid theory and the connection

Manuscript received November 8, 2009; revised March 21, 2010; accepted November 2, 2010. Date of publication January 31, 2011; date of current version February 18, 2011. This work was supported by the Institute for Defense Analyses and the National Science Foundation.

R. Dougherty is with the Center for Communications Research, San Diego, CA 92121-1969 USA (e-mail: rdough@ccrwest.org).
C. Freiling is with the Department of Mathematics, California State University at San Bernardino, San Bernardino, CA 92407-2397 USA (e-mail: cfreilin@csusb.edu).
K. Zeger is with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093-0407 USA (e-mail: zeger@ucsd.edu).

Digital Object Identifier: 10.1109/JPROC.2010.2095490

between them. Then, we discuss the constructions of several particular networks from matroids and the results obtained from these constructions. Most proofs are omitted in this paper, but references are given to the original sources where complete proofs can be found.

II. NETWORK CODING FUNDAMENTALS

For the purposes of this paper, an *alphabet* $\mathcal{A}$ is a finite set. Informally, we think of a *message* as an arbitrary string of k alphabet symbols and a *packet* is a string of m alphabet symbols. More precisely, a message is a variable with domain $\mathcal{A}^k$ and a packet is a variable with domain $\mathcal{A}^m$. A *network* is based on a finite, directed, acyclic multigraph. A network is also assigned a finite set of messages. Each message originates at a particular node, called the *source node* for that message, and is required by one or more *demand nodes*. Information about the messages is passed from node to node in the form of packets; there is one packet for each edge of the graph. Thus, all edges have the same bandwidth, and the bandwidth between two nodes can only be increased by adding additional edges. For any given network, we may consider different values of k and m, but these must be kept consistent throughout the entire network.

The first interesting example of network coding is the *butterfly* network illustrated in Fig. 1. (In all networks drawn in this paper, unless otherwise indicated, a letter above a node indicates a message whose source is that node, and a letter below a node indicates a message demanded at that node. Letters beside certain edges denote the packets carried by those edges. A node marked with numeral i will be referred to as n_i, and an edge from n_i to n_j may be referred to as $e_{i,j}$.) The butterfly network was introduced in 2000 by Ahlswede *et al.* [2] to demonstrate the utility of network coding. This network has no routing solution. But with $k = m = 1$ and any size alphabet $\mathcal{A} = \{0, 1, \ldots, a-1\}$, there is a simple coding solution. To see this, let node n_1 emit the message x along its two out-edges (it has no other viable choice). Similarly, let node n_2 emit message y along its two out-edges. Node n_3 will emit $z = (x + y) \pmod a$, and, having no sensible alternative, node n_4 will emit z along its two out-edges. Then, node n_5 recovers y from $y = (z - x) \pmod a$, and node n_6 recovers x from $x = (z - y) \pmod a$.

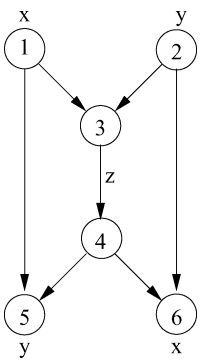

Fig. 1. *The butterfly network.*

More generally, the *inputs* to a network node n are the packets carried on its in-edges, together with any messages generated at n. We will denote this set by $\mathrm{In}(n)$. The *outputs* of n are all of the packets carried on its out-edges, together with any messages demanded at n. We will denote this set by $\mathrm{Out}(n)$. Each output of a node must be a function of its inputs. A *coding solution* for the network is an assignment of such functions (one for each output of each node) which gives the correct results: if edge packets and demand values are computed according to the assigned functions, then the demand values match the messages that were demanded. When the values of k and m need to be emphasized, it will be called a (k, m)-*solution*. Informally, a network coding solution allows each demand node to deduce its demanded messages by having information from the sources propagate through the network in the form of packets. Note that each edge is allowed to be used only once (i.e., at most one packet can travel across each edge).

Two special types of coding solutions are *routing solutions* and *linear solutions*. In a routing solution, all packets must be strings of message symbols, though they are allowed to mix different messages together. In a linear solution, we assume the alphabet $\mathcal{A}$ consists of the elements of a finite ring, and usually, it will be a finite field. Hence, all messages are k-long vectors of ring elements while the packets are m-long vectors. The functions in a linear solution must only use the operations of vector addition and multiplication of a vector by a constant matrix (whose components are ring elements).

The case $k = m$ will be of particular interest. If $k = m = 1$, then a coding solution is said to be a *scalar solution*. If $k = m \geq 2$, it is said to be a *vector solution*. A network is said to be *solvable* if it has a scalar solution over some finite alphabet, *scalar-linearly solvable* if it has a scalar-linear solution, *vector-linearly solvable* if it has a vector-linear solution, etc. (Note that if a network has a (k, k)-solution over an alphabet $\mathcal{A}$, then it has a scalar solution over alphabet $\mathcal{A}^k$, so a network having a vector solution is solvable. "Solvability" refers specifically to the case where the edge capacity matches the message size; without such a restriction, any sufficiently connected network would be solvable simply by using a packet large enough to carry all of the messages.) In fact, when we talk about coding solutions that are not required to have $k = m$, we may emphasize this fact by referring to them as *fractional solutions*.

As another example, consider the M-network, illustrated in Fig. 2. (The edges labeled u_1 through u_4 are the out-edges from node n_4.) This network is due to Koetter and was used by Medard *et al.* in [31] as an example of a

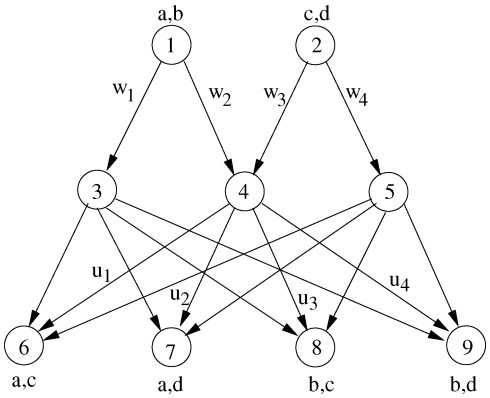

Fig. 2. ***The M-network.***

network that is vector-routing solvable, but not scalar-linear solvable. Their solution with $k = m = 2$ is as follows. Let $\mathcal{A}$ be any alphabet. Let $a = (a_1, a_2)$, $b = (b_1, b_2)$, $d = (d_1, d_2)$, $d = (d_1, d_2)$ denote the pairs of alphabet symbols for each message. Then, let the sources emit the packets

$$w_1 = (a_1, b_2), \quad w_2 = (a_2, b_1)$$
$$w_3 = (c_1, d_2), \quad w_4 = (c_2, d_1)$$

and let node n_4 emit the packets

$$u_1 = (a_2, c_1), \quad u_2 = (a_2, d_2)$$
$$u_3 = (b_1, c_1), \quad u_4 = (b_1, d_2).$$

Nodes n_3, n_4, and n_5 have only one input, so they just copy their input along each outgoing edge. The demands at nodes n_6, n_7, n_8, n_9 are now easily met.

If a network has a (k, m)-solution over an alphabet $\mathcal{A}$, then it has a $(2k, 2m)$-solution over the same alphabet. Indeed, a $(2k, 2m)$-solution can be constructed from two disjoint copies of the (k, m)-solution. Because of this scaling property, we are primarily interested in the ratio k/m. Therefore, when a (k, m)-solution exists over an alphabet $\mathcal{A}$, the number k/m is said to be an *achievable rate* of the network, with respect to $\mathcal{A}$. That is, k/m is the amount of source data that is successfully sent through a network, per network edge bandwidth. An important goal in network coding is to find an achievable coding rate which is as large as possible for a network. The *coding capacity of a network over an alphabet* $\mathcal{A}$ (a related definition appears in [40, p. 339]) is defined to be the supremum (least upper bound) of the ratio k/m over all pairs (k, m) for which the network has a (k, m)-solution over $\mathcal{A}$. If we restrict attention to routing solutions or linear solutions, then the coding capacity is referred to as the *routing capacity* or *linear coding capacity*, respectively. In all cases, if the alphabet $\mathcal{A}$ is not mentioned, the capacity is taken to be the supremum of the capacities over all alphabets $\mathcal{A}$. The coding capacity of a given network is said to be *achievable* if there is some (k, m)-solution for the network for which k/m equals the capacity (in which case the supremum of achievable rates equals the maximum of achievable rates).

A network is *multicast* if there is only one source node and every receiver demands every source message. A network is *multiple unicast* if every source message is demanded by exactly one demand node (but a demand node can demand more than one message). The butterfly network, as presented above, is multiple unicast. But there is a multicast version of this network with a single source node pointing to nodes n_1 and n_2 where nodes n_5 and n_6 both demand messages x and y. One of the most important facts about multicast networks is the following 2003 result of Li *et al.* [28].

Theorem II.1: All solvable multicast networks are scalar-linearly solvable over some finite field.

For more introductory material on network coding, see [18] and [41].

III. POLYMATROIDS AND UPPER BOUNDS ON THE CAPACITY OF A NETWORK

When evaluating a network, we often consider the messages to be independent random variables. This allows us to consider the entropy for any collection of messages and packets, and thus keep track of the information as it flows through the network. Using this perspective, and letting $H(x)$ denote Shannon's entropy function (computed using logarithms to base $|\mathcal{A}|$), the basic requirements of a network coding solution are summarized by the following properties.

- (N1) (source rates): $H(M) = k|M|$ for any collection M of messages.
- (N2) (edge capacities): $H(p) \leq m$ for any packet p.
- (N3) (node input/output functional dependencies): $H(\mathrm{In}(n)) = H(\mathrm{In}(n) \cup \mathrm{Out}(n))$ for any node n.

Shannon [34] showed that when we apply the entropy function $H(x)$ to collections of random variables, certain basic inequalities must be satisfied. These *Shannon-type information inequalities* are important in the study of network coding, as they allow us to deduce upper bounds on the coding capacity of a network. These basic inequalities are also satisfied by the rank function on linear subspaces of a vector space, and there are other examples as well. These basic inequalities are sometimes called the *polymatroidal axioms*, and any function $f(x)$ that satisfies these axioms is called a *polymatroid*.

To be more precise, let $\mathcal{S}$ be a finite set and let f map subsets of $\mathcal{S}$ to nonnegative real numbers. The conditions (P1)–(P3) are called the *polymatroidal axioms* for f.

- (P1) $f(\varnothing) = 0$.
- (P2) If $A \subset B \subset \mathcal{S}$, then $f(A) \leq f(B)$.
- (P3) If $A, B \subset \mathcal{S}$, then $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$.

Alternatively, one can replace conditions (P2) and (P3) with the following combined version.

- (P4) If $A, B, C \subset \mathcal{S}$, then $f(A \cup C) + f(B \cup C) \geq f(C) + f(A \cup B \cup C)$.

When we consider a polymatroid in connection with a network $\mathcal{N}$, the finite set $\mathcal{S}$ will always be the collection of all messages and packets of the network. If such a polymatroid f also satisfies the conditions (N1)–(N3) (when H is replaced by f), then it will be called a (k, m)-*polymatroid assignment* to $\mathcal{N}$. The *polymatroid upper bound on the capacity* of network $\mathcal{N}$ is the supremum of the ratio k/m over all pairs (k, m) for which the network has a (k, m)-polymatroid assignment. Note that if $\mathcal{N}$ has a (k, m)-coding solution, then the derived entropy function on $\mathcal{S}$ is such a (k, m)-polymatroid assignment. So the terminology "polymatroid upper bound on the capacity" of a network is justified. However, a network may have many polymatroid assignments that are not entropy functions, so it is possible for the polymatroid upper bound to be greater than the coding capacity.

The purpose of the polymatroid assignments is to make precise the meaning of "bounds that are derivable from the Shannon inequalities." Indeed, if an upper bound on the coding capacity of $\mathcal{N}$ is derived from Shannon-type information inequalities and uses no information about entropy other than what is contained in the inequalities (P1)–(P4) and the network entropy conditions (N1)–(N3), then it should also be an upper bound for every polymatroid assignment to $\mathcal{N}$. Thus, we may say (somewhat loosely) that the polymatroid upper bound on capacity is *the best upper bound on the network coding capacity obtainable using only Shannon inequalities*. We will see in Section VIII-B that, for some networks, one can use *non-Shannon-type information inequalities* to produce capacity bounds strictly tighter than the polymatroid upper bound.

IV. MATROID FUNDAMENTALS

A *matroid* $\mathcal{M}$ is an ordered pair $(\mathcal{S}, \mathcal{I})$, where $\mathcal{S}$ is a finite set and $\mathcal{I}$ is a set of subsets of $\mathcal{S}$ satisfying the following three conditions.

- (I1) $\varnothing \in \mathcal{I}$.
- (I2) If $I \in \mathcal{I}$ and $J \subset I$, then $J \in \mathcal{I}$.
- (I3) If $I, J \in \mathcal{I}$ and $|J| < |I|$, then there is an element e of $I - J$ such that $J \cup \{e\} \in \mathcal{I}$.

The set $\mathcal{S}$ is called the *ground set* and the matroid $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ is called a matroid *on* $\mathcal{S}$. The members of $\mathcal{I}$ are called *independent sets* and any subset of $\mathcal{S}$ not in $\mathcal{I}$ is called a *dependent set*. A maximal independent set of a matroid is called a *base* of the matroid and a minimal dependent set is called a *circuit*.

There are many equivalent definitions of a matroid. One such alternate definition, which is particularly useful for us, uses the notion of a rank function. For any $X \subset \mathcal{S}$, the *rank* of X, denoted $r(X)$, is the size of any maximal independent subset of X. It is easy to show using (I3) that all such subsets are the same size.

The rank function is a function r from subsets of $\mathcal{S}$ to integers satisfying the following three conditions.

- (R1) If $X \subset \mathcal{S}$, then $0 \leq r(X) \leq |X|$.
- (R2) If $X \subset Y \subset \mathcal{S}$, then $r(X) \leq r(Y)$.
- (R3) If $X, Y \subset \mathcal{S}$, then $r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y)$.

Note the correspondence with the polymatroidal axioms (P1)–(P3). Thus, the rank function of a matroid is an example of a polymatroid. We will refer to (I1)–(I3) as the *independence axioms* of a matroid and to (R1)–(R3) as the *rank axioms* of a matroid. The two sets of axioms give two equivalent ways to define a matroid. (One can reconstruct $\mathcal{I}$ from r by letting $\mathcal{I} = \{X \subset \mathcal{S} : r(X) = |X|\}$.)

The primary example of a matroid comes from linear algebra. Let A be an $m \times n$ matrix over a field F. Let $\mathcal{S} = \{1, \ldots, n\}$ and let $X \subset \mathcal{S}$. If the columns indexed by X are linearly independent over F, then we will say that $X \in \mathcal{I}$. The pair $(\mathcal{S}, \mathcal{I})$ forms a matroid, called the *vector matroid* of A.

Another important example of a matroid is obtained from graph theory. If $\mathcal{S}$ is the set of edges of a finite undirected graph, and $\mathcal{I}$ is the collection of all subforests (i.e., cycle-free subgraphs), then $(\mathcal{S}, \mathcal{I})$ is a matroid. The spanning forests and cycles of the graph are, respectively, the bases and circuits in the matroid. The rank of any subgraph determined by a subset of $\mathcal{S}$ is the number of edges in a spanning forest of the subgraph.

A third useful collection of matroids is the family of *uniform matroids* $U_{m,n}$, defined as follows. The ground set of $U_{m,n}$ is the set $\{1, \ldots, n\}$, and a subset of the ground set is independent if and only if it has size at most m.

Two matroids $(\mathcal{S}, \mathcal{I})$ and $(\mathcal{S}', \mathcal{I}')$ are said to be *isomorphic* if there exists a bijection $f : \mathcal{S} \to \mathcal{S}'$ such that $I \in \mathcal{I}$ if and only if $f(I) \in \mathcal{I}'$. If a matroid $\mathcal{M}$ is isomorphic to the vector matroid over a field F, then $\mathcal{M}$ is said to be *representable over* F or *F-representable*. A matroid is *representable* if it is representable over some field.

Matroids of small rank are often depicted geometrically. A rank-3 matroid is represented by a figure in which marked points are ground set elements, with any two distinct points giving a size-2 independent set in the matroid. Three points in the figure represent an independent set if and only if they are not collinear in the figure. (Sometimes the figure will indicate that certain curved "lines" in the figure are to be treated as dependent sets.) For a rank-4 matroid, one uses a 3-D diagram where collinear triples and coplanar quadruples are dependent sets (although

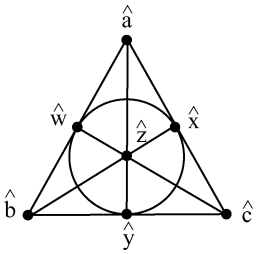

Fig. 3. *Geometric depiction of the Fano matroid.*

usually some hints from the text are needed to tell just which quadruples are supposed to be coplanar).

For example, the matroid depicted in Fig. 3 is called the Fano matroid.[1] The matroid has ground set $\{\hat{a}, \hat{b}, \hat{c}, \hat{w}, \hat{x}, \hat{y}, \hat{z}\}$ and has rank 3. Any three elements of the ground set are dependent if and only if they are collinear in the diagram (where we pretend that points on the drawn circle are also "collinear").

For a detailed introduction to matroid theory, the reader is referred to [32] or [36].

V. NETWORKS FROM MATROIDS

In this section, we describe a method for building networks from matroids. The method involves a number of choices and hence does not produce a unique network.

Let $\mathcal{N}$ be a network with message set M and packet set P. Let $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ be a matroid with rank function r. A *network-matroid mapping* from $\mathcal{N}$ to $\mathcal{M}$ is a function $f : M \cup P \to \mathcal{S}$ such that the following conditions are satisfied.

- (M1) $f|M$ is one-to-one.
- (M2) $f(M) \in \mathcal{I}$.
- (M3) $r(f(\mathrm{In}(n))) = r(f(\mathrm{In}(n) \cup \mathrm{Out}(n)))$, for every node n.

Conditions (M1) and (M2) reflect that the messages are independent while (M3) is a reflection of the network dependencies. When such a mapping exists, we say that the network $\mathcal{N}$ is *matroidal over* $\mathcal{M}$ and we also say that $\mathcal{N}$ is a *matroidal* network. However, the matroid witnessing that a particular network is matroidal need not be unique.

As an example, let us see that the butterfly network (Fig. 1) is matroidal over the uniform matroid $U_{2,3}$. Recall that $U_{2,3}$ has ground set $\{1, 2, 3\}$, and a subset of the ground set is independent if and only if it has size at most 2. To see a network-matroid mapping, let f assign the element 1 to the message x, which originates at node n_1 and also to the two packets emanating from node n_1. Let f assign the element 2 to the message y, and also to the packets emanating from node n_2. The three remaining packets are assigned the element 3. The conditions (M1) and (M2) are easily checked. To see (M3), note that $f(\mathrm{In}(n)) = f(\mathrm{In}(n) \cup \mathrm{Out}(n))$ for nodes $n = n_1, n_2, n_4$. At each of the other nodes, $n = n_3, n_5, n_6$, $f(\mathrm{In}(n) \cup \mathrm{Out}(n)) = \{1, 2, 3\}$, which has rank 2 while $f(\mathrm{In}(n))$ has size two and is therefore independent and therefore also has rank 2.

We now present some basic facts about matroidal networks. First, it follows from (M1) and (M2) that:

$$r(f(S)) = |S| \qquad \text{for all message subsets } S. \tag{1}$$

The following fact about matroidal networks will be used in Section VIII-B.

Lemma V.1 [11, p. 1956]: For any matroidal network, the polymatroid upper bound on the capacity is at least 1.

So, to show that Shannon inequalities are insufficient for computing coding capacity, it suffices to find a matroidal network that has capacity less than 1. This is accomplished using the Vámos network (to be described later).

The following theorem demonstrates that a large class of interesting networks are matroidal.

Theorem V.2 [11, p. 1956]: If a network is scalar-linearly solvable over some finite field, then the network is matroidal over a representable matroid.

It follows from Theorem V.2 and Theorem II.1 that all solvable multicast networks are matroidal.

Theorem V.2 suggests a technique for obtaining a network that has a good chance of not being scalar-linearly solvable: choose a network that is matroidal over a nonrepresentable matroid. The Vámos matroid defined in Section V-D is the smallest example of a nonrepresentable matroid [32, p. 512], providing inspiration to define and study a "Vámos network."

A. Constructing Matroidal Networks

A method was given in [11] for constructing matroidal networks from matroids. Such constructions allow interesting properties of matroids to be transferred to networks. As matroid theory is a field rich in important results, the goal in constructing matroidal networks is to obtain some analogues for networks.

Let $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ be a matroid with rank function r. Let $\mathcal{N}$ denote the network to be constructed, with message set M, node set N, and packet set P.

The construction simultaneously constructs the network $\mathcal{N}$, the function $f : M \cup P \to \mathcal{S}$, and an auxiliary function $g : \mathcal{S} \to N$, where for each $x \in \mathcal{S}$, either

i) $g(x)$ is a source node with message m and $f(m) = x$; or
ii) $g(x)$ is a node with in-degree 1 and whose incoming packet p satisfies $f(p) = x$.

The construction is carried out in four stages; each stage can be completed in many ways. We will first

[1] Named after the Italian mathematician Gino Fano (1871–1952), father of the information theorist Robert Fano (1917–).

describe the entire construction and then illustrate the steps with an example.

Step 1) Create network source nodes $n_1, n_2, \ldots, n_{r(\mathcal{S})}$ and corresponding messages $m_1, m_2, \ldots, m_{r(\mathcal{S})}$. Choose any base $B = \{b_1, \ldots, b_{r(\mathcal{S})}\}$ for $\mathcal{M}$ and let $f(m_i) = b_i$ and $g(b_i) = n_i$.

Step 2) (To be repeated until it is no longer possible.) Find a circuit $\{x_0, \ldots, x_j\}$ in $\mathcal{M}$, such that $g(x_1), \ldots, g(x_j)$ have been already defined, but $g(x_0)$ has not yet been defined. Then, we will add the following.

i) a new node y, edges $e_1, \ldots, e_j$, and corresponding packets $p_1, \ldots, p_j$, such that e_i connects $g(x_i)$ to y, and we define $f(p_i) = x_i$.

ii) Another new node n_0 with a single in-edge e_0 and corresponding packet p_0, connecting y to n_0, and we let $f(p_0) = x_0$ and $g(x_0) = n_0$.

Step 3) (To be repeated as many times as desired.) If $\{x_0, \ldots, x_j\}$ is a circuit in $\mathcal{M}$ and $g(x_0)$ is a source node with message m_0, then add to the network a new demand node y, which demands the message m_0 and which has in-edges $e_1, \ldots, e_j$ with corresponding packets $p_1, \ldots, p_j$ where e_i connects $g(x_i)$ to y and where $f(p_i) = x_i$.

Step 4) (To be repeated as many times as desired.) Choose a base $B = \{x_1, \ldots, x_{r(\mathcal{S})}\}$ of $\mathcal{M}$ and create a demand node y that demands all of the network messages, and such that y has in-edges $e_1, \ldots, e_{r(\mathcal{S})}$ with corresponding packets $p_1, \ldots, p_{r(\mathcal{S})}$ where e_i connects $g(x_i)$ to y. Let $f(p_i) = x_i$.

The butterfly network (Fig. 1) was already shown to be matroidal over the uniform matroid $U_{2,3}$. To illustrate the construction procedure, we will show how to construct the butterfly network from $U_{2,3}$ using the steps above. The results of these steps are shown in Figs. 4–6. In order to avoid confusion, we will first rename the elements of the ground set of $U_{2,3}$ so that $\mathcal{S} = \{x, y, z\}$. Recall that a subset of $\mathcal{S}$ is independent if it has size at most 2. Therefore, the bases of this matroid are the sets $\{x, y\}$, $\{x, z\}$, and $\{y, z\}$ and the only circuit is $\{x, y, z\}$.

Step 1) We choose a matroid base $B = \{x, y\}$. We create source nodes n_1, n_2, and network messages m_1 and m_2, and we assign $f(m_1) = x$ and $f(m_2) = y$, and $g(x) = n_1$ and $g(y) = n_2$.

Fig. 4. *After step 1.*

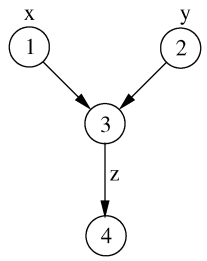

Fig. 5. *After step 2.*

Step 2) The only circuit in the matroid is $\{x, y, z\}$, and $g(x) = n_1$ and $g(y) = n_2$ have already been defined, but $g(z)$ has not yet been defined. For Step 2(i), we add a new node n_3. We also add an edge $e_{1,3}$ from n_1 to n_3 with packet $p_{1,3}$, and an edge $e_{2,3}$ from n_2 to n_3 with packet $p_{2,3}$, and we define $f(p_{1,3}) = x$ and $f(p_{2,3}) = y$. For Step 2(ii), we add another new node n_4 with a single in-edge $e_{3,4}$ with packet $p_{3,4}$ so that $e_{3,4}$ connects n_3 to n_4. Then, we let $f(p_{3,4}) = z$ and $g(z) = n_4$.

Step 3) The only circuit in the matroid is $\{x, y, z\}$ and $g(y) = n_2$ is a source node with message m_2. We add a new demand node n_5, which demands the message m_2 and has in-edges $e_{1,5}$ and $e_{4,5}$ and corresponding packets $p_{1,5}$ and $p_{4,5}$. Edge $e_{1,5}$ connects $g(x) = n_1$ to the new node n_5 and edge $e_{4,5}$ connects $g(z) = n_4$ to n_5. We then let $f(p_{1,5}) = x$ and $f(p_{4,5}) = z$. We repeat this step once more with the same circuit $\{x, y, z\}$, but this time using the source node $g(x) = n_1$ with message m_1. We add a new demand node n_6, which demands the message m_1 and has in-edges $e_{2,6}$ and $e_{4,6}$ with corresponding packets $p_{2,6}$ and $p_{4,6}$, where $f(p_{2,6}) = y$ and $f(p_{4,6}) = z$.

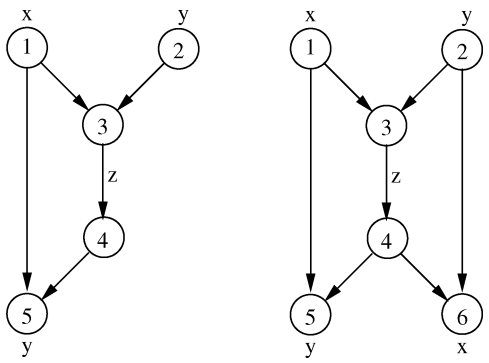

Fig. 6. *After step 3.*

The result is the butterfly network. Also constructed is the network-matroid mapping f showing that this network is matroidal over the uniform matroid $U_{2,3}$. Note that (the optional) Step 4 of the construction was not needed.

It should be noted that the construction procedure described here is not quite general enough to fully handle all matroids. One can reach a situation where there is a circuit for which all g values have already been defined (so Step 2 is not applicable) and none of these values is a source node (so Step 3 cannot be applied), so the dependency given by this circuit will not be reflected in the network. There are extensions or alternative constructions that can fully handle all matroids; for instance, see Section X-C.

We will now describe some additional matroids that were constructed from well-known matroids using this procedure. Most of the construction details will be omitted.

B. The Fano Network

Fig. 3 is a geometric depiction of the well-known Fano matroid [32]. The network shown in Fig. 7, called the *Fano network*, is matroidal over the Fano matroid and can be constructed using the technique described in Section V-A.

The Fano matroid is known to be F-representable over a finite field F if and only if F has characteristic two [32]. Correspondingly, the Fano network was shown in [9] to be solvable if and only if the alphabet size is an integer power of two. It, in fact, has a linear solution over any finite field of characteristic two (by taking $w = a + b$, $x = a + c$, $y = b + c$, and $z = a + b + c$).

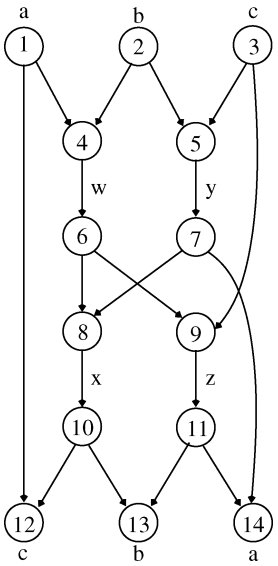

Fig. 7. *The Fano network.*

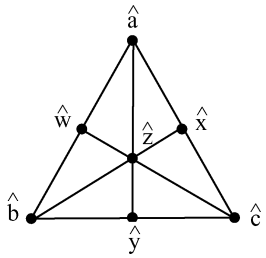

Fig. 8. *Geometric depiction of the non-Fano matroid.*

The network-matroid mapping is shown in Fig. 7 (message a is mapped to matroid element $\hat{a}$, the packet on the edge labeled w is mapped to matroid element $\hat{w}$, and so on). As usual, we omit the label on edges emanating from any node n with only one input. Nodes n_1–n_3 were constructed in Step 1 of the construction, nodes n_4–n_{11} in Step 2, and nodes n_{12}–n_{14} in Step 3; Step 4 was not used.

C. The Non-Fano Network

Fig. 8 is a geometric depiction of the well-known non-Fano matroid [32]. The matroid has ground set $\{\hat{a}, \hat{b}, \hat{c}, \hat{w}, \hat{x}, \hat{y}, \hat{z}\}$ and has rank 3. Any three elements of the ground set are dependent if and only if they are collinear in the diagram. The network shown in Fig. 9, called the *non-Fano network*, is matroidal over the non-Fano matroid and is constructed using the technique described in Section V-A.

The non-Fano matroid is known [32] to be F-representable over a finite field F if and only if F has odd

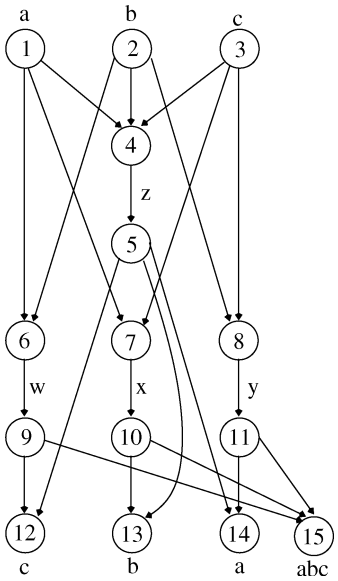

Fig. 9. *The non-Fano network.*

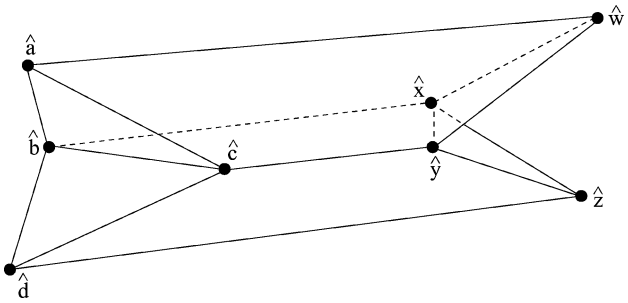

Fig. 10. *A 3-D geometric depiction of the Vámos network.*

characteristic. Correspondingly, the non-Fano network was shown in [9] to be solvable if and only if the alphabet size is odd.[2] It, in fact, has a linear solution over any finite field of odd cardinality, or even any Abelian group of odd cardinality (by taking $w = a + b$, $x = a + c$, $y = b + c$, and $z = a + b + c$).

Nodes n_1–n_3 were constructed in Step 1 of the construction, nodes n_4–n_{11} in Step 2, nodes n_{12}–n_{14} in Step 3, and node n_{15} in Step 4.

D. The Vámos Network

The *Vámos matroid* is an 8-element rank-4 matroid depicted in the 3-D drawing in Fig. 10. The ground set is $\mathcal{S} = \{\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{w}, \hat{x}, \hat{y}, \hat{z}\}$. All subsets of cardinality at least 5 are dependent. The subsets of cardinality 4, which are intended to be coplanar in the diagram and hence dependent in the matroid, are $\{\hat{b}, \hat{c}, \hat{x}, \hat{y}\}$, $\{\hat{a}, \hat{b}, \hat{w}, \hat{x}\}$, $\{\hat{a}, \hat{b}, \hat{w}, \hat{x}\}$, $\{\hat{c}, \hat{d}, \hat{y}, \hat{z}\}$, and $\{\hat{b}, \hat{d}, \hat{x}, \hat{z}\}$. The set $\{\hat{a}, \hat{d}, \hat{w}, \hat{z}\}$ is not considered a coplanar set in Fig. 10 and is independent in the Vámos matroid (though it is impossible to draw this accurately without distortion).

The Vámos matroid is not representable, but every matroid smaller than the Vámos matroid is representable [32, p. 170].

The network shown in Fig. 11 is called the *Vámos network*; it is matroidal over the Vámos matroid[3] and constructed using the technique described in Section V-A. The network has 17 nodes and four message variables. Nodes $n_9, \ldots, n_{13}$ are demand nodes, each demanding one source message, except for n_{11}, which demands two source messages. The network has four hidden source nodes, each generating exactly one of the messages a, b, c, d. As depicted in Fig. 11, source messages are carried on hidden edges from their hidden source to various other network

[2]Actually, a slight variation of the non-Fano network was used in [9]; the variation consisted of removing the demands a and b from node n_{15}. However, the statements here about the solvability of the non-Fano network are true, since it can be shown that the non-Fano network is equivalent to the variant network in the sense of being solvable over the same fields and linearly solvable over the same fields.

[3]It should be emphasized that there are many other networks which are matroidal over the Vámos matroid.

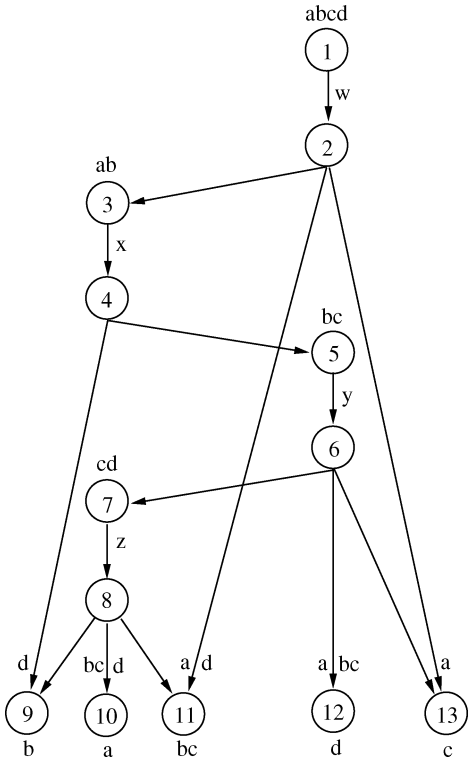

Fig. 11. *The Vámos network. A message variable a, b, c, or d labeled above a node indicates an in-edge (not shown) from the source node (not shown) generating the message.*

nodes (e.g., message c is carried by hidden edges from its hidden source to nodes n_1, n_5, n_7, n_{10}, and n_{12}).

Note: As depicted in Fig. 11, several of the message variables a, b, c, d appear above some of the nodes. This is simply a convenience that makes the depiction easier to draw. When this happens, it is understood that there is an unshown edge from the appropriate source node to the node in question. So, for example, node n_1 actually has four in-edges (not shown), one from each source node (also not shown).

E. A Non-Matroidal Network

A trivial example of an unsolvable network that is not matroidal is shown in Fig. 12. The two messages a and b generated at node n_1 and demanded by node n_2 cannot both be sent over the single link between nodes n_1 and n_2.

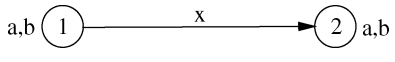

Fig. 12. *An unsolvable nonmatroidal network.*

There are also solvable networks that are not matroidal. One example is the *M*-network shown in Fig. 2. Recall that the *M*-network was presented in [31] as an example of a network with no scalar linear solution, but with a simple vector linear solution. This result can be extended further. The 2-D vector linear solution to the *M*-network given in [31] is a simple routing solution and easily extends to a vector linear solution over any even vector dimension. In [11], it was shown that the *M*-network does not have any (finite field) vector linear solutions of odd vector dimension, generalizing the fact that the *M*-network does not have a scalar-linear solution.

VI. UNACHIEVABILITY OF CAPACITY FOR SOME NETWORKS

When one considers the coding capacity of a network by looking at vectors of symbols over a base alphabet, it turns out that the size of the base alphabet does not affect the final computed capacity [6]. Given long enough vectors, one can simply convert from one alphabet to another at the sources and do the reverse conversion at the sinks, with a loss of efficiency less than any prespecified bound. Of course, these alphabet conversions are highly nonlinear; we will see later that linear coding capacities can depend quite strongly on the size of the base alphabet.

We will now show how the incompatibility of representations between the Fano and non-Fano matroids (one is representable only over fields of characteristic 2, the other only over fields not of characteristic 2) has been used to produce a network which has coding capacity 1 but is not solvable (so there are fractional coding solutions which approach the capacity arbitrarily closely, but no solution that attains it). The argument here comes from [9].

We have already constructed networks corresponding to these two matroids. Since the networks are both matroidal, they have coding capacity at least 1. We now want to see that the incompatibility in the matroids carries over to a corresponding incompatibility in the networks. So we must analyze the solutions to the Fano and non-Fano networks; note that we are not looking at just linear solutions, but all solutions.

Somewhat surprisingly, it turns out to be possible to give complete analyses of the solutions to these two networks, and they are both of the same form: up to permutations of the alphabet at each edge, any such solution can be expressed in terms of an Abelian group. (These permutations are unavoidable; given any solution, one can alter it by performing an alphabet permutation at the beginning of an edge and undoing the permutation at the end of the edge, to get a variant solution.)

The general result is that any solution to either the Fano or the non-Fano network has the following form: for some Abelian group operation $\oplus$ on the alphabet $\mathcal{A}$ and some permutations $\pi_1, \ldots, \pi_6$ of $\mathcal{A}$, the contents of the edges w, x, y, z are given in terms of the messages a, b, c by

$$\begin{aligned} w &= \pi_4(\pi_1(a) \oplus \pi_2(b)) \\ x &= \pi_5(\pi_1(a) \oplus \pi_3(c)) \\ y &= \pi_6(\pi_2(b) \oplus \pi_3(c)) \\ z &= \pi_1(a) \oplus \pi_2(b) \oplus \pi_3(c). \end{aligned}$$

The difference between the two networks shows up as a restriction on the Abelian group $(\mathcal{A}, \oplus)$. For the Fano network, all elements of the group must have order 1 or 2; for the non-Fano network, all elements must have odd order. Hence, the Fano network is solvable only for alphabets whose size is a power of 2, while the non-Fano network is solvable only for odd-sized alphabets.

Now consider a network which is simply a disjoint union of a Fano network and a non-Fano network. This network cannot be solvable, because the alphabet size for a solution would have to be both odd (for the non-Fano part) and a power of 2 (for the Fano part). However, one can get a fractional solution coming arbitrarily close to the capacity by using a large power-of-2 alphabet for the messages, handling the Fano side directly, and handling the non-Fano side via alphabet conversions as described at the beginning of this section.

We note that it is not pathological that a disjoint union of two networks was used to construct a network that cannot achieve its coding capacity. In fact, the Fano and non-Fano networks can easily be connected by adding three new source nodes, each producing one of the messages *a*, *b*, or *c*, and then adding out-edges from these new source nodes to the corresponding sources of the Fano and non-Fano networks. This will create a connected network having the same property of not being able to achieve its coding capacity.

VII. INSUFFICIENCY OF LINEAR NETWORK CODING

In this section, we will modify the example from the preceding section so as to get a network that is solvable but is not linearly solvable, thus demonstrating that linear coding does not suffice in general to attain the full benefits of network coding. The construction here comes from [8].

The network will again be assembled from two parts, and one of the two parts will be the Fano network as before. For the other part, we use the network $\mathcal{N}_2$ shown in Fig. 13. This is a slightly weakened form of the non-Fano network; it has two copies of the non-Fano network cooperating to meet one demand for message *c* at the bottom. It is easy to see that any solution to the non-Fano network can be copied (twice) to give a solution to $\mathcal{N}_2$, so $\mathcal{N}_2$ is solvable over all odd-sized alphabets. Also, the proof that

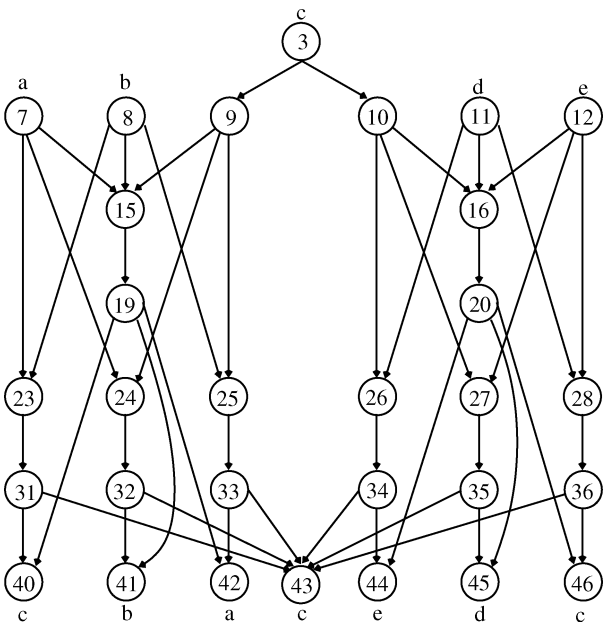

Fig. 13. *The network $\mathcal{N}_2$, a weakened version of the non-Fano network.*

the non-Fano network has no vector linear solution over fields of characteristic 2 can be used to show the same fact for $\mathcal{N}_2$. But it turns out that $\mathcal{N}_2$ does have a nonlinear solution over an alphabet of size 4. (This will be shown in Fig. 14.)

Now, as before, we assemble the two parts to form a combined network $\mathcal{N}_3$, shown in Fig. 14. (This time we explicitly show how one can make it a connected network by having the two parts share the three source messages a, b, c.)

The combined network $\mathcal{N}_3$ cannot have a vector linear solution, because the base field would have to be of characteristic 2 for the Fano part and not of characteristic 2 for the $\mathcal{N}_2$ part. But $\mathcal{N}_3$ does have a solution over the four-element alphabet $\{0, 1, 2, 3\}$. This is shown in the figure; here $\oplus$ is bitwise addition without carry (XOR), $+$ is ordinary addition modulo 4, and t is the operation of interchanging the bits in a two-bit number [so $t(0) = 0$, $t(2) = 1$, $t(2) = 1$, and $t(3) = 3$]. Nodes n_{31}, n_{32}, n_{33} have enough information to reconstruct $2c$ modulo 4, which means we get the lower bit of c; and nodes n_{34}, n_{35}, n_{36} have enough information to reconstruct the lower bit of $t(c)$, which is the upper bit of c, so together they can reconstruct all of c. So $\mathcal{N}_3$ is a network that is (nonlinearly) solvable but has no linear solution over any finite field and any vector dimension.

The coding capacity of the network $\mathcal{N}_3$ is 1. (We have already seen a solution attaining capacity 1; the fact that there is a unique path from the source node n_1 for message a to node n_{39}, which demands a, can be used to show that the capacity is at most 1. Actually, all that is needed is that there is some single edge that all paths from n_1 to n_{39} must pass through.) Given the results of the preceding section, it is natural to ask whether the network

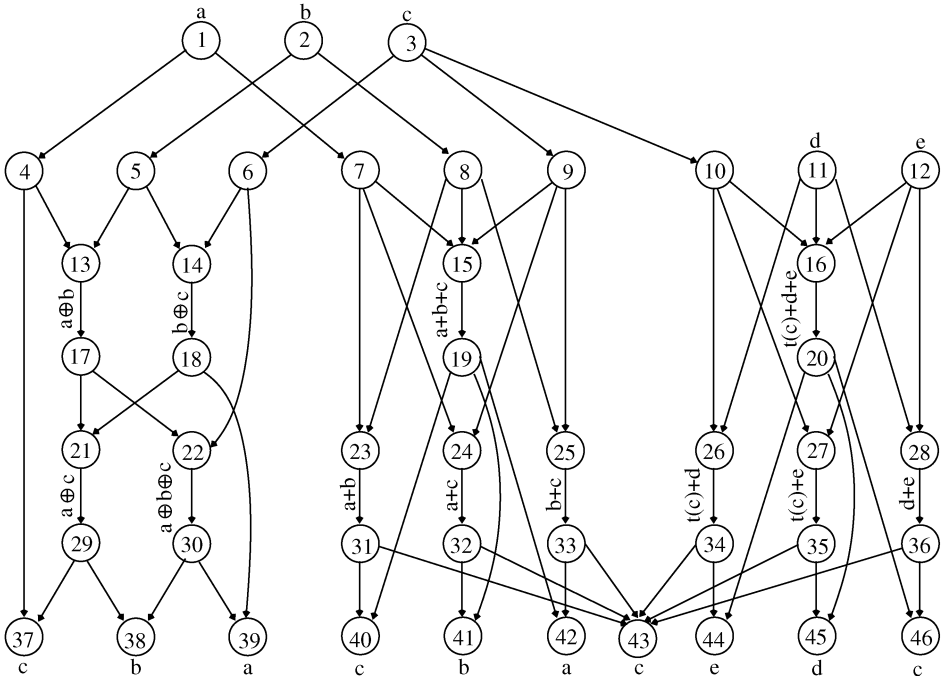

Fig. 14. *The network $\mathcal{N}_3$ demonstrating the insufficiency of linear coding.*

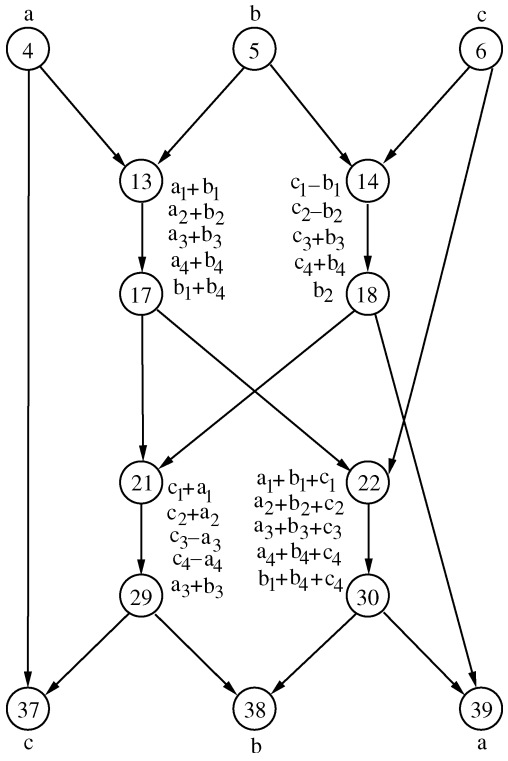

Fig. 15. *A (4,5) fractional linear solution over any field alphabet for the Fano network.*

$\mathcal{N}_3$ can "almost" be solved linearly (i.e., whether there are fractional linear solutions approaching capacity 1 arbitrarily closely). This turns out not to be the case; in fact, we can compute the linear capacity of $\mathcal{N}_3$ and its parts precisely.

The linear capacities of these networks depend on the size of the base alphabet (field). For the Fano network, we have already seen a (scalar) linear solution over any field of characteristic 2; and the capacity is at most 1 by the argument of the preceding paragraph, so the characteristic-2 linear capacity of the Fano network is 1. On the other hand, some rather involved linear algebra can be used to show that, for a field which is not of characteristic 2, the linear capacity of the Fano network is at most 4/5. This value can actually be attained by the solution shown in Fig. 15. Here message a consists of four components a_1, a_2, a_3, a_4 (and similarly for b and c) from the base field, and each edge can carry five such components.

The linear coding capacity of the network $\mathcal{N}_2$ is 1 over any odd-characteristic finite field; this can be attained by using two copies of the scalar linear solution for the non-Fano network. But over fields of characteristic 2, the linear capacity of $\mathcal{N}_2$ works out to be exactly 10/11. (Again one can give an explicit linear solution using 10-component messages and 11-component edge values.) It now follows that the linear coding capacity of the combined network $\mathcal{N}_3$ is 10/11 over fields of characteristic 2 and 4/5 over all other fields, so in no case does one have a vector linear solution, which closely approaches the coding capacity of 1. In fact, the coding capacity of $\mathcal{N}_3$ is exactly 10% greater than the maximum linear coding capacity over any finite field.

The fact that network $\mathcal{N}_3$ is solvable but not even asymptotically linearly solvable (i.e., its linear capacity is bounded by a number strictly less than 1) allows one to deduce that even more general coding methods such as convolutional coding or filter-bank linear coding (see [22] for the definitions of these) cannot be used to give a linear solution for network $\mathcal{N}_3$. One also cannot reach the full coding capacity by "time sharing" the network between linear solutions over different fields. As shown in [8], the negative results still hold even if the vector spaces over fields here are generalized to modules over rings.

VIII. BOUNDS ON CAPACITIES OF SOME NETWORKS

In general, the routing capacity of an arbitrary network can in principle be determined using a linear programming approach [6],[4] although the computational complexity can be prohibitive for even relatively small networks. It thus appears to be generally nontrivial to efficiently determine the routing capacity. In addition, there are presently no known techniques for computing the coding capacity or the linear coding capacity of an arbitrary network.[5] In fact, the linear coding capacity of a network depends, in general, on the finite field alphabet used [8], whereas the routing capacity and coding capacity do not depend on the alphabet size [6].

Here we present a few more capacity bounds for the example networks we have already seen. Most of these computations come from [8] and [11]; extensions to computations of rate regions (where messages can have different sizes) will be presented in [13].

A. The Fano and Non-Fano Networks

For the Fano network, we have already seen the coding capacity (1) and the linear coding capacity (1 for characteristic-2 fields, 4/5 for other fields). The routing capacity is at most 2/3 because each part of all of the messages a, b, c must pass through at least one of the edges labeled x and z in Fig. 7. It is not hard to see that the value 2/3 can be attained: just route messages a and c on the direct paths to their destinations while sending half of message b via edges w and z and the other half via edges y and x. So the routing capacity is exactly 2/3.

The non-Fano network can be treated by similar methods. The routing capacity of this network is 1/3: it is trivial to attain this value by simply sending all messages anywhere they can go, and one cannot do better because all three messages a, b, c must pass through the edge labeled z in Fig. 9 in order to reach their respective

[4]This is analogous to the algorithm for achieving multicommodity flow capacity given in [17].

[5]An exception is for multicast networks, where it is known that the coding capacity equals the linear coding capacity and is computable [28].

destinations n_{14}, n_{13}, n_{12}. The coding capacity is at most 1 (again because of the bottleneck at z, which still applies to each message separately even in the coding case), and this is attainable using linear coding over any odd-characteristic field, as noted in Section V-C. So the coding capacity and the odd-characteristic linear capacity are both equal to 1. This leaves the linear capacity for characteristic-2 fields, which turns out to be 5/6. Again the upper bound is proved by rather messy linear algebra, while the lower bound is proved by giving an explicit (5,6) fractional linear solution.

B. The Vámos Network

For the Vámos network, we first note that the routing capacity is relatively easy to compute. Referring back to Fig. 11, we see that, in any routing solution, messages a and d must both pass in their entirety through both of edges x and y in order to reach their corresponding demand nodes. Also, all parts of message b must pass through either edge x or edge y to get to demand node n_9. Hence, edges x and y must together have the capacity to carry five messages; this means that the routing capacity of the network is at most 2/5. It turns out that the routing capacity is in fact equal to 2/5; this value is attainable by splitting message b into two halves b_1 and b_2 and letting edges w, x, y, and z carry combinations (a, b_1, d), (a, b_1, d), (a, b_2, d), and (a, b_2, c), respectively.

Computing the coding capacity of the Vámos network is much more difficult; in fact, it is an open question at the moment, though we do have partial results. First, it is easy to see that the coding capacity is at most 1, because all of the information in message a must pass through the edge $e_{8,10}$ in order to get to demand node n_{10}. Using standard methods based on the Shannon information inequalities, we cannot get a better bound than this; since the Vámos network is matroidal, Lemma V.1 shows that the polymatroid upper bound on the coding capacity of the Vámos network is at least 1. [The argument given informally here using $e_{8,10}$ and the bottleneck arguments for the other networks are abbreviations for derivations using the Shannon inequalities. For instance, the argument for the Vámos network could be given more formally as follows: $H(a) \leq H(e_{8,10})$ because

$$\begin{aligned} & H(e_{8,10}) + H(b,c,d) \\ & \quad \geq H(e_{8,10}, b, c, d) && \text{[from Shannon]} \\ & \quad = H(e_{8,10}, b, c, d, a) && \text{[from (N3)]} \\ & \quad \geq H(a, b, c, d) && \text{[from Shannon]} \\ & \quad = 4k && \text{[from (N1)]} \\ & \quad = H(a) + H(b,c,d), && \text{[from (N1)]} \end{aligned}$$

so $k = H(a) \leq H(e_{8,10}) \leq m$ by (N1) and (N2), so $k/m \leq 1$.]

But Shannon inequalities do not tell the whole story. In 1998, Zhang and Yeung [42] gave the first example of an information inequality on four random variables which is not of Shannon type; that is, the inequality is true for any four jointly distributed discrete random variables, but it does not follow directly from the Shannon inequalities. The Zhang–Yeung inequality can be written in the following form:

$$I(A;B) \leq 2I(A;B|C) + I(A;C|B) + I(B;C|A) + I(A;B|D) + I(C;D).$$

[Here $I(A;B)$ and $I(A;B|C)$ are shorter ways of writing the entropy combinations $H(A) + H(B) - H(A,B)$ and $H(A,C) + H(B,C) - H(C) - H(A,B,C)$, respectively.]

Using the Zhang–Yeung inequality along with the Shannon and network inequalities, one can improve the upper bound on the coding capacity of the Vámos network to 10/11. In particular, since the coding capacity is strictly less than 1, the Vámos network is not solvable. This was the first application of a non-Shannon-type information inequality to network coding.

Since 1988, many additional non-Shannon-type information inequalities have been discovered (see, for instance, [10], [30], [39], and [41]). In fact, in 2007 Matúš showed [30] that the list of non-Shannon-type information inequalities on four variables is essentially infinite—no finite list of them implies all of the others.

It turns out that most of the inequalities found so far do not further improve the Vámos network coding capacity bound. But the inequality

$$2I(A;B) \leq 4I(A;B|C) + 3I(A;C|B) + I(B;C|A) + 2I(A;B|D) + 2I(C;D)$$

from [14] does give a slight further improvement: the coding capacity of the Vámos network is at most 19/21.

One more inequality that can be used in these computations is the Ingleton inequality [21]. This inequality is rather opaque in its usual form, but becomes much clearer when written as in [19]

$$I(A;B) \leq I(A;B|C) + I(A;B|D) + I(C;D).$$

Unlike the preceding inequalities, the Ingleton inequality is *not* an information inequality—there exist jointly distributed random variables that violate the Ingleton inequality. However, the Ingleton inequality does hold for the special case of random variables that vary uniformly and independently over specified subspaces of a given finite

vector space. These are precisely the kind of random variables that come up when one is considering *linear* solutions to a network coding problem. Hence, the Ingleton inequality can be used to give an upper bound on the *linear* capacity of the Vámos network; this bound turns out to be 5/6. This bound is sharp; one can give an explicit linear (5,6)-solution to the Vámos network (which works over any field).

So the linear capacity of the Vámos network is 5/6, while the coding capacity is somewhere between 5/6 and 19/21 (inclusive). Furthermore, the known non-Shannon-type inequalities provide numerous bounding inequalities for the rate region of the Vámos network; in fact, since the closed entropy region for four random variables is not polytopal [30], it is possible that the Vámos network rate region is not a polytope. However, no solution to the Vamos network has yet been found outside the linear rate region, so it is possible that the coding capacity is just 5/6 and the coding rate region matches the linear rate region (which, for the Vámos network, is known to be a simple polytope not depending on the field).

The Vámos network is not the only network for which non-Shannon-type information inequalities provide capacity bounds. Chan and Grant [7] prove a quite general result, showing that, given *any* non-Shannon-type information inequality, one can construct a network for which the given inequality provides an improvement to the polymatroid bounds for the rate-capacity region of the network.

IX. NETWORK SOLVABILITY AND POLYNOMIAL SYSTEMS

The problem of determining whether a given network is solvable is quite difficult. Even if the alphabet size is known in advance and small, a brute-force search of the possible network codes is almost always computationally infeasible. One cannot assume that the alphabet will be small; Lehman and Lehman [27] give examples of solvable networks for which the minimum alphabet size required for a solution is extremely large (doubly exponential in the size of the network).

In fact, since no upper bound is known on the size of the alphabet that would be required for a given network, it is possible that there is no algorithm at all for determining network solvability; the problem could be undecidable. This could be demonstrated by converting instances of another known undecidable problem to networks whose solvability would correspond to a positive answer to the other problem. One candidate for such a problem is Rhodes' problem on identities in finite groups, although this is not yet known to be undecidable [3].

Even if one restricts to vector linear solutions, no algorithm is known for determining whether a given network is solvable (and again such an algorithm might not exist). But the scalar linear case is more tractable. Koetter and Médard [25] show that, from a given network, one can produce a system of polynomial equations such that scalar linear solutions to the network correspond exactly to solutions to the polynomial system. This reduces the scalar linear network solvability problem to the polynomial system satisfiability problem, which is known to be solvable using the method of Gröbner bases [4].

However, even this is a complicated and time-consuming algorithm; it is natural to ask whether there is an alternative, simpler way to determine whether a network is scalar-linearly solvable. This question is answered in [12], where a construction is given that is basically a converse to the Koetter–Médard construction: given a polynomial system, it produces (constructively, without much increase in size) a network such that, for any finite field F, the network is solvable over F if and only if the polynomial system has a solution over F. Thus, in terms of computational complexity, the network scalar-linear solvability problem and the polynomial system satisfiability problem are equally difficult.

This is connected to matroid theory in two ways. First, it turns out that the network constructed as above is matroidal. Second, the motivation for the construction came from a 1936 construction of MacLane [29] in the early history of matroid theory.

MacLane's construction is motivated by some simple geometric constructions in the plane for doing arithmetic. One can add two lengths by simply concatenating segments; and constructions involving similar triangles allow us to multiply numbers. So, given a unit segment and some other starting lengths, one can do constructions involving intersections of lines and drawing parallel lines to draw a segment, which is a polynomial in the given starting lengths. The parallel lines can be eliminated by moving to the *projective plane*, which includes "points at infinity" and a "line at infinity" so that any two lines intersect at a point. Then, one replaces the field of real numbers with a finite field to get finite projective planes. In these finite projective planes, one gets low-rank matroids just as in the geometric depictions described in Section IV.

The result of MacLane's construction is as follows. Let $\mathcal{P}$ be a polynomial collection and let K be a finite field such that $\mathcal{P}$ has a solution over K. Then, MacLane constructs a matroid $\mathcal{M}$ that is representable over K and such that, for any finite field F, if $\mathcal{M}$ is representable over F, then $\mathcal{P}$ has a solution over F.

The construction in [12] uses network elements to imitate MacLane's point-and-line constructions in order to get a network that is fully equivalent to $\mathcal{P}$ as stated above.

The difference between the matroid situation and the network situation needs some further explanation.

Given an instance of a problem that depends on a finite field F (e.g., matroid representability, network scalar-linear solvability, or polynomial system satisfiability), the *set of characteristics* of the problem instance is the set of prime numbers p such that the problem instance is solvable for some field F of characteristic p.

After a great deal of work built partially on MacLane's initial construction, it was shown that a set of primes is the set of characteristics for representability of a matroid if and only if the set is finite or cofinite [32, Sec. 6.8]. The corresponding fact about polynomial systems can be proved using quite easy constructions, so the equivalence presented here shows that the same result holds for network scalar-linear satisfiability.

But the matroid situation is substantially more complicated than that for the other two problems. MacLane's construction did not give a two-way equivalence between matroids and polynomial systems, and other results show that such an equivalence is impossible; for instance, a matroid that is representable over both $GF(2)$ and $GF(3)$ must be representable over all fields [32, Th. 6.6.3], while this is not true for polynomial systems (or networks). One possible reason for this is that, in a matroid, every set must be specified to be either definitely independent or definitely dependent. There is more flexibility in networks; if one has two edges in different parts of a network; the information carried by those two edges could be fully independent, fully dependent, or partially dependent, and the situation could be different for different solutions to the network. This extra flexibility allows us to get a full equivalence between network scalar-linear solvability and polynomial system satisfiability that could not be attained for matroids.

X. RELATED TOPICS

A. Multiple-Unicast Networks

The special case of multiple-unicast networks (where each message is sent by only one source and demanded by only one destination) has received attention for several reasons: many real-world applications are of this type; certain additional questions make more sense in this context (see, for instance, Section X-B); and such networks may be easier to work with (for instance, the algorithm given by Adler *et al.* [1] and Harvey *et al.* [20] for computing coding capacity bounds of networks applies as stated only to multiple-unicast networks).

It turns out that, if one is considering only questions of solvability or linear solvability (where the edge capacity is the same as the message size), there is no loss of generality in restricting to the case of multiple-unicast networks. This is shown in [15], where a technique is given for converting arbitrary networks into multiple-unicast networks. The conversion procedure preserves the solvability and linear solvability properties of the original network; it also preserves the property of a network being matroidal.

We want to convert a given network into an equivalent network in which each message has only one source and one demand node. Actually, the network definitions we presented here did not allow multiple sources for the same message, but even if they had, eliminating multiple sources for a given message would be easy: simply add a new node to be the sole source for this message, together with an edge from this node to each of the old sources of the message.

We eliminate multiple demand nodes by iterating the following construction. Suppose that we have two nodes n_x and n_y, which both demand message z. Then, we can add a gadget consisting of the nodes and edges shown in Fig. 16, so that the two demands for z are replaced with one demand for z and one demand for a new message w.

A solution to the old network can easily be extended to the modified network; edges $e_{x,2}$ and $e_{y,5}$ will carry z, and, as with the butterfly, the edge $e_{2,3}$ will carry the sum of w and z. Conversely, if one has a solution to the modified network, then one can use information-theoretic arguments to show that edges $e_{x,2}$ and $e_{y,5}$ must be carrying z (or equivalent forms under alphabet permutation), so the demands for z in the original network could be satisfied.

If the original network is matroidal over matroid M, then, as shown in [11], the modified network is matroidal over a matroid which is an amalgam (see [32]) of M and the uniform matroid $U_{2,3}$.

If a network has k demands for message z, then adding $k-1$ of these gadgets will eliminate the multiple demands for z (but add $k-1$ new messages, each with one demand). Once this is done for all multiply-demanded messages, the final network will be multiple-unicast, and equivalent to the original network for purposes of solvability and linear solvability.

Fig. 17 shows the result of applying this construction to the Vámos network; we refer to this as the multiple-unicast Vámos network. Recall that we defined the Vámos network to have a single source for each message, but did not depict these in the diagram in order to avoid clutter; we use the same convention here, so no further steps are

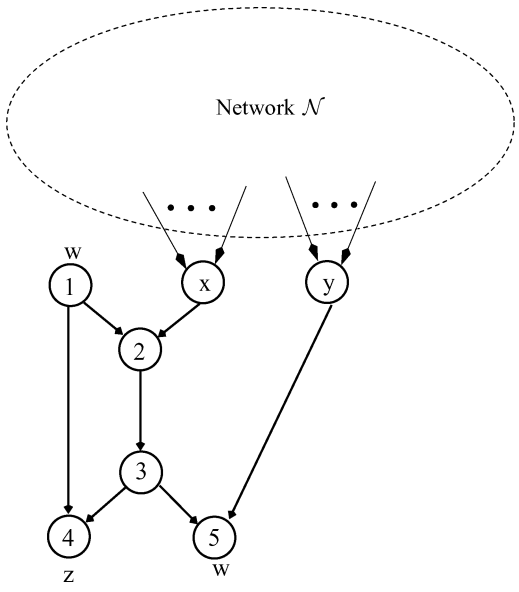

Fig. 16. *A gadget for eliminating a double demand for the message z.*

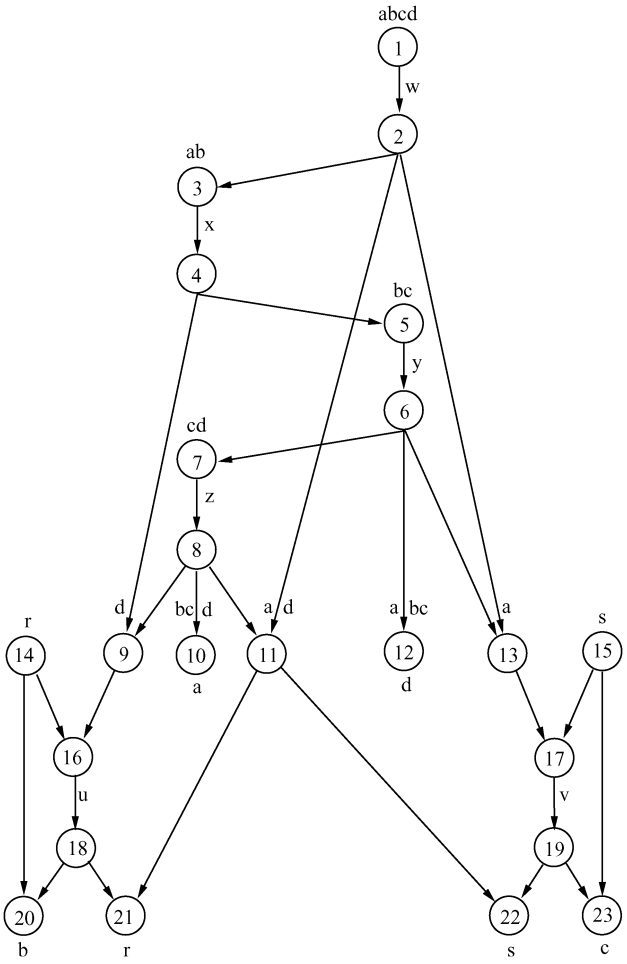

Fig. 17. ***The multiple-unicast Vámos network.***

needed to eliminate multiple sources. There are only two duplicated demands in the Vámos network, so two gadgets suffice to produce a multiple-unicast network. This new network is matroidal and unsolvable, just as the original was.

However, the adding of gadgets does not, in general, preserve capacities, so questions of capacity must be reexamined here. The same reasoning as in Section VIII-B shows that the polymatroid upper bound on the coding capacity of the multiple-unicast Vámos network is 1. But the improved bound given by the Zhang–Yeung inequality is slightly weaker here; one gets an upper bound of 12/13 for the capacity of the multiple-unicast Vámos network. (This is still enough to show that the algorithm given in [1] and [20] does not always yield the best possible coding capacity bound.)

B. Reversibility and Nonreversibility of Networks

The *reverse* of a network $\mathcal{N}$ is a network $\mathcal{N}'$ satisfying the following.

1) The nodes of $\mathcal{N}'$ are the same as in $\mathcal{N}$.
2) The edges of $\mathcal{N}'$ are the same as in $\mathcal{N}$ but each in the reversed direction.
3) Each node that emits messages in $\mathcal{N}$ instead demands the same messages in $\mathcal{N}'$.
4) Each node that demands messages in $\mathcal{N}$ instead emits the same messages in $\mathcal{N}'$.

A network is said to be *reversible* if its reverse is solvable. A network is *linearly reversible* if its reverse is linearly solvable. Note that the reverse of a multiple-unicast network is also multiple-unicast.

Clearly, if a multiple-unicast network has a routing solution, then it is reversible, by simply reversing the direction of information flow of the given routing solution. However, if network coding is used, then reversibility is not as straightforward. It was shown, however, in [23], [24], and [33] that all linearly solvable multiple-unicast networks are linearly reversible over the same alphabet. (This theorem applies to both the scalar-linear and vector-linear cases.) In [23] and [24], an elegant "duality" principle is given, connecting algebraic coding theory and linearly reversible networks, and applications of reversibility are discussed. In [33], a network is given that has a binary (nonlinear) solution but whose reverse does not have a binary solution.

This left open the question of whether a solvable network could be fully nonreversible (i.e., over all alphabets). In light of the results in [23], [24], and [33], such a network could not have a linear solution over any finite field alphabet.

This question is answered in [8], where it is shown that one can modify the network $\mathcal{N}_3$ in Fig. 14 (actually, a variant $\mathcal{N}'_3$ of this network with the sources of the two halves separated) by replacing the six-input node n_{43} with a suitable cascade of four nodes, to get a new network $\mathcal{N}^*_3$, as shown in Fig. 18. Such a modification would not have any important effect on linear solutions, since a linear combination of six inputs can easily be separated to fit on such a cascade. But the nonlinear solution to $\mathcal{N}'_3$ cannot be transferred over to $\mathcal{N}^*_3$; it turns out that $\mathcal{N}^*_3$ is unsolvable. On the other hand, if one looks at the reverses of the networks $\mathcal{N}'_3$ and $\mathcal{N}^*_3$, one sees that their only difference is that a node with one input and six outputs has been replaced with a four-node fan-out cascade. This change cannot possibly affect solvability of a network, since a node with one input cannot do anything useful other than copy its input to its outputs, and this could be done just as easily in the cascade. So either both $\mathcal{N}'_3$ and $\mathcal{N}^*_3$ are reversible or neither is. This leads to the interesting conclusion that there must be a solvable network that is not reversible, but it is not immediately clear which network it is; it is either $\mathcal{N}'_3$ (if $\mathcal{N}'_3$ is not reversible) or the reverse of $\mathcal{N}^*_3$ (if $\mathcal{N}'_3$ is reversible).

It turns out that $\mathcal{N}'_3$ is reversible, so the reverse of $\mathcal{N}^*_3$ is the solvable nonreversible network. Furthermore, the methods of Section X-A can be used to turn this network into a multiple-unicast network, which is solvable but not reversible.

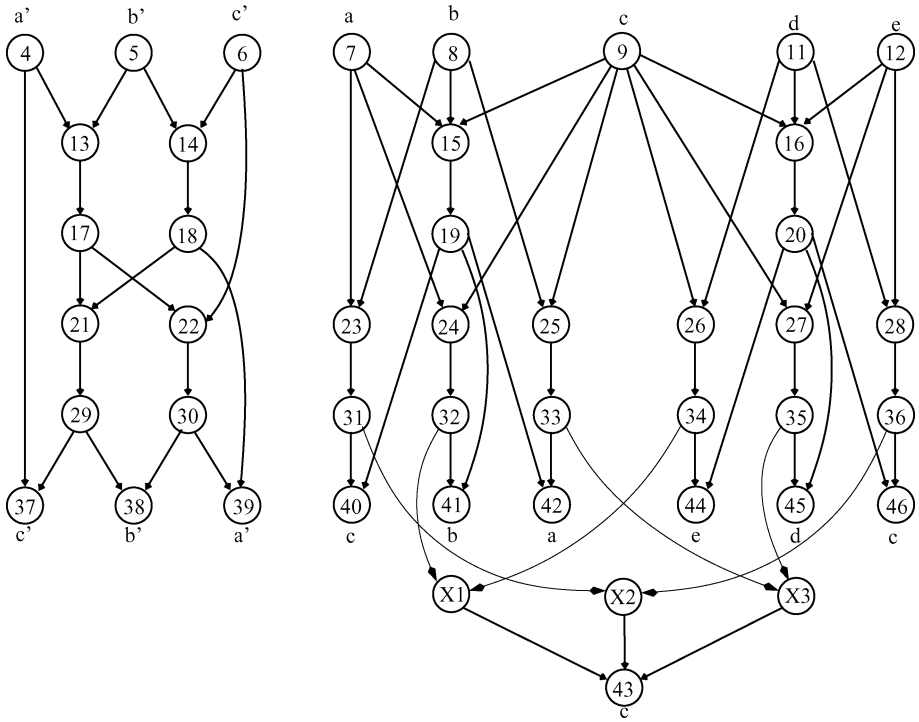

Fig. 18. ***The network $\mathcal{N}_3^*$, an unsolvable variant of $\mathcal{N}_3$.***

C. Index Coding

El Rouayheb *et al.* [16] have recently given a new approach to the connection between network coding and matroid theory. Their approach connects these two topics to a third topic, the *index coding* problem.

For the index coding problem, we have one source and multiple receivers. The source has a certain number of messages, each of which is an element of a fixed finite set (often a field or a vector space). Each receiver desires one of these messages, and already knows some of the other messages (and the source is aware of this knowledge). The source can now broadcast a limited number of elements of the finite set (perhaps the given messages, perhaps combinations of them) to all receivers; the goal is for the source to broadcast as few such elements as possible in order to allow the receivers, using the knowledge they already have, to deduce the messages they desire.

Here is an example from [16]. Suppose the source has four messages x_1, x_2, x_3, x_4, and there are four receivers: receiver 1 knows x_2 and x_3 and desires x_1, receiver 2 knows x_1 and x_3 and desires x_2, receiver 3 knows x_2 and x_4 and desires x_3, and receiver 4 knows x_1 and desires x_4.

If the source could broadcast four elements, the problem would be trivial. But it turns out that the source can satisfy all of the receivers by broadcasting only two elements, namely $x_1 + x_4$ and $x_1 + x_2 + x_3$, where $+$ is a given group operation on the finite set. (One can show that two elements is the optimal answer for this problem.) This is an example of a scalar-linear solution to the index coding problem; one can also consider vector-linear solutions as well as nonlinear solutions.

There is a straightforward way to get from an instance of the index coding problem to an instance of the network coding problem. One builds the network by starting with a source node for each source message of the index coding problem, a demand node for each receiver (demanding the message that receiver desires), and an internal edge for every broadcast element. There are edges from each source node to the tail of each broadcast edge, and from the head of each broadcast edge to each demand node; also, there is an edge from each source message node to each demand node corresponding to a receiver that is supposed to already know that message. It is easy to see that solutions to this network (scalar-linear, vector-linear, or arbitrary) correspond precisely to solutions to the given index coding problem.

El Rouayheb *et al.* [16] give another construction, which from a given matroid yields an instance of the index coding problem in such a way that linear representability of the matroid over a finite field corresponds exactly to scalar-linear solvability of the index coding problem, and multilinear representability of the matroid (a generalization of linear representability in which each matroid ground set element corresponds to multiple columns of a matrix rather than a single column) corresponds exactly to vector-linear solvability of the index coding problem. Combining this construction with the previous one gives a mapping from arbitrary matroids to network coding instances, which fully reflects the linearity properties of the matroids.

A converse construction is also given in [16]; given a network coding instance, it produces an index coding instance that has scalar-linear or vector-linear solutions over exactly the same fields as the given network coding instance. (A full converse construction back to matroids is not possible for the reasons given at the end of Section IX.) These results establish a very close and useful connection between three apparently quite different types of problem. ■

REFERENCES

[1] M. Adler, N. J. A. Harvey, K. Jain, R. D. Kleinberg, and A. Rasala Lehman, "On the capacity of information networks," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Miami, FL, Jan. 2006, DOI: 10.1145/1109557.1109585.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[3] D. Albert, R. Baldinger, and J. Rhodes, "Undecidability of the identity problem for finite semigroups," *J. Symb. Logic*, vol. 57, no. 1, pp. 179–192, Mar. 1992.

[4] T. Becker, V. Weispfenning, and H. Kredel, *Gröobner Bases: A Computational Approach to Commutative Algebra*. New York: Springer-Verlag, 1993.

[5] L. M. Blumenthal, *A Modern View of Geometry*. New York: Dover, 1961.

[6] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 777–788, Mar. 2006.

[7] T. H. Chan and A. Grant, "Dualities between entropy functions and network codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4470–4487, Oct. 2008.

[8] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[9] R. Dougherty, C. Freiling, and K. Zeger, "Unachievability of network coding capacity," *IEEE Trans. Inf. Theory/IEEE/ACM Trans. Netw.*, vol. 52, no. 6, pp. 2365–2372, Jun. 2006.

[10] R. Dougherty, C. Freiling, and K. Zeger, "Six new non-Shannon information inequalities," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, Jul. 2006, pp. 233–236.

[11] R. Dougherty, C. Freiling, and K. Zeger, "Networks, matroids, and non-Shannon information inequalities," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 1949–1969, Jun. 2007.

[12] R. Dougherty, C. Freiling, and K. Zeger, "Linear network codes and systems of polynomial equations," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2303–2316, May 2008.

[13] R. Dougherty, C. Freiling, and K. Zeger, "Some achievable rate regions with network coding," in preparation.

[14] R. Dougherty, C. Freiling, and K. Zeger, "Non-Shannon information inequalities in four random variables," in preparation.

[15] R. Dougherty and K. Zeger, "Nonreversibility and equivalent constructions of multiple-unicast networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5067–5077, Nov. 2006.

[16] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3187–3195, Jul. 2010.

[17] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ: Princeton Univ. Press, 1962.

[18] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*. Boston, MA: Now, 2007.

[19] D. Hammer, A. E. Romashchenko, A. Shen, and N. K. Vereshchagin, "Inequalities for Shannon entropy and Kolmogorov complexity," *J. Comput. Syst. Sci.*, vol. 60, pp. 442–464, 2000.

[20] N. J. A. Harvey, R. D. Kleinberg, and A. Rasala Lehman, "On the capacity of information networks," *IEEE Trans. Inf. Theory/IEEE/ACM Trans. Netw.*, vol. 52, no. 6, pp. 2345–2364, Jun. 2006.

[21] A. W. Ingleton, "Representation of matroids in combinatorial mathematics and its applications," in *Combinatorial Mathematics and Its Applications*, D. J. A. Welsh, Ed. London, U.K.: Academic, 1971, pp. 149–167.

[22] S. Jaggi, M. Effros, T. Ho, and M. Médard, "On linear network coding," in *Proc. 42nd Annu. Allerton Conf. Commun. Control Comput.*, Monticello, Illinois, pp. 40–49, Oct. 2004.

[23] R. Koetter, "A duality principle in network coding," presented at the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) Workshop on Algebraic Coding Theory and Information Theory, Piscataway, NJ, Dec. 2003.

[24] R. Koetter, M. Effros, T. Ho, and M. Médard, "Network codes as codes on graphs," presented at the 38th Annu. Conf. Inf. Sci. Syst., Princeton, NJ, Mar. 2004.

[25] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[26] G. Kramer and S. A. Savari, "Edge-cut bounds on network coding rates," *J. Netw. Syst. Manage.*, vol. 14, no. 1, pp. 49–67, Mar. 2006.

[27] A. Rasala Lehman and E. Lehman, "Network information flow: Does the model need tuning?" in *Proc. Symp. Discrete Algorithms*, Vancouver, BC, Canada, Jan. 2005, pp. 499–504.

[28] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[29] S. MacLane, "Some interpretations of abstract linear dependence in terms of projective geometry," *Amer. J. Math.*, vol. 58, no. 1, pp. 236–240, Jan. 1936.

[30] F. Matúš, "Infinitely many information inequalities," in *Proc. Int. Symp. Inf. Theory*, Nice, France, 2007, pp. 41–44.

[31] M. Médard, M. Effros, T. Ho, and D. Karger, "On coding for non-multicast networks," in *Proc. 41st Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, pp. 21–29, Oct. 2003.

[32] J. G. Oxley, *Matroid Theory*. New York: Oxford Univ. Press, 1992.

[33] S. Riis, "Reversible and irreversible information networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4339–4349, Nov. 2007.

[34] C. E. Shannon. (1948, Jul./Oct.). A mathematical theory of communication. *Bell Syst. Tech. J.* [Online]. 27, pp. 379–423. Available: http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html

[35] L. Song, R. W. Yeung, and N. Cai, "Zero-error network coding for acyclic networks," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3129–3139, Dec. 2003.

[36] D. J. A. Welsh, *Matroid Theory*. London, U.K.: Academic, 1976.

[37] N. L. White, Ed., *Combinatorial Geometries*. Cambridge, U.K.: Cambridge Univ. Press, 1987.

[38] H. Whitney, "On the abstract properties of linear dependence," *Amer. J. Math.*, vol. 57, pp. 509–533, 1935.

[39] W. Xu, J. Wang, and J. Sun, "A projection method for derivation of non-Shannon-type information inequalities," in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 2116–2120.

[40] R. W. Yeung, *A First Course in Information Theory*. Norwell, MA: Kluwer, 2002.

[41] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*. Boston, MA: Now, 2006.

[42] Z. Zhang and R. W. Yeung, "On characterization of entropy function via information inequalities," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1440–1452, Jul. 1998.

ABOUT THE AUTHORS

Randall Dougherty was born in 1961. He received the A.B. and Ph.D. degrees in mathematics from the University of California Berkeley, Berkeley, in 1981 and 1985, respectively.

After postdoctoral work at the California Institute of Technology, Pasadena, and the University of California Los Angeles, Los Angeles, he joined the mathematics faculty of The Ohio State University, Columbus, in 1989, becoming an Associate Professor in 1992 and a Professor in 1998. In 2000, he left Ohio State to work for the software company LizardTech, Seattle, WA; in 2002, he started to work full time at the Center for Communications Research (CCR), La Jolla, CA. He also spent shorter periods working as a Consultant in various organizations, such as CCR and the Los Alamos National Laboratory. His mathematical research interests range from foundations (logic and set theory) to other pure mathematics (real analysis, group theory, combinatorics) to applied work (numerical methods, network design, computational geometry, etc.).

Dr. Dougherty is a three-time International Mathematical Olympiad medalist and a three-time Putnam Fellow. He received a National Science Foundation (NSF) Presidential Young Investigator Award in 1991 and a Sloan Research Fellowship in 1992.

Christopher Freiling was born in 1954. He received the Ph.D. degree in mathematics from the University of California Los Angeles (UCLA), Los Angeles, in 1981, under the direction of D. Martin.

He has been on the faculty of the California State University at San Bernardino, San Bernardino, since 1983, interrupted by two visiting appointments at UCLA. His mathematical interests include foundations and real analysis.

Dr. Freiling is the recipient of the 2003 Andy Award in Real Analysis.

Kenneth Zeger (Fellow, IEEE) was born in Boston, MA, in 1963. He received the S.B. and S.M. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1984 and the M.A. degree in mathematics and the Ph.D. in electrical engineering from the University of California in Santa Barbara, Santa Barbara, in 1989 and 1990, respectively.

He was an Assistant Professor of Electrical Engineering at the University of Hawaii, Honolulu, from 1990 to 1992. He was at the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, as an Assistant Professor from 1992 to 1995, and as an Associate Professor from 1995 to 1996. He has been with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, as an Associate Professor from 1996 to 1998, and as a Professor from 1998 to present.

Dr. Zeger received a National Science Foundation (NSF) Presidential Young Investigator Award in 1991. He served as an Associate Editor At-Large for the IEEE Transactions on Information Theory during 1995–1998, and as a member of the Board of Governors of the IEEE Information Theory Society during 1998–2000, 2005–2007, and 2008–2010.