Unachievability of Network Coding Capacity

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

***Abstract*—The coding capacity of a network is the supremum of ratios k/n, for which there exists a fractional (k, n) coding solution, where k is the source message dimension and n is the maximum edge dimension. The coding capacity is referred to as routing capacity in the case when only routing is allowed. A network is said to achieve its capacity if there is some fractional (k, n) solution for which k/n equals the capacity. The routing capacity is known to be achievable for arbitrary networks. We give an example of a network whose coding capacity (which is 1) cannot be achieved by a network code. We do this by constructing two networks, one of which is solvable if and only if the alphabet size is odd, and the other of which is solvable if and only if the alphabet size is a power of 2. No linearity assumptions are made.**

***Index Terms*—Capacity, flow, network coding, switching.**

I. Introduction

In this paper, unless stated otherwise, a *network* is a directed acyclic multigraph, some of whose nodes are information sources or receivers (e.g., see [11]). Associated with the sources are *messages*, which are assumed to be vectors of k arbitrary elements of a fixed finite alphabet of size at least 2. At any node in the network, each out-edge carries a vector of n alphabet symbols which is a function (called an *edge function*) of the vectors of symbols carried on the in-edges to the node, and/or a function of the node's message vectors if it is a source. Associated with each receiver are *demands*, which are a subset of all the messages of all the sources. Each receiver has *decoding functions* which map the receiver's inputs to vectors of symbols in an attempt to produce the messages demanded at the receiver. The goal is for each receiver to deduce its demanded messages from its in-edges and sources by having information propagate from the sources through the network. Each edge is allowed to be used at most once (i.e., at most n symbols can travel across each edge). Special cases of interest include linear codes, where the edge functions and decoding functions are linear, and routing codes, where the edge functions and decoding functions simply copy input components to output components.

A (k, n) *fractional code* is a collection of edge functions, one for each edge in the network, and decoding functions, one for each demand of each node in the network. For a (k, n) fractional code, the ratio k/n is called a *coding rate*. A (k, n) *fractional solution* is a (k, n) fractional code which results in every receiver being able to compute its demands via its demand functions. If a network has a (k, n) fractional solution over some alphabet, then we say the coding rate k/n is *achievable* for the network. A network is said to be *solvable* if it has a (k, n) fractional solution for the case $k = n = 1$.

The coding capacity of a network with respect to an alphabet $\mathcal{A}$ and a class $\mathcal{C}$ of network codes (a related definition appears in [11, p. 339]) is

$$\sup\{k/n : \exists\ (k, n) \text{ fractional coding solution in } \mathcal{C} \text{ over } \mathcal{A}\}.$$

If $\mathcal{C}$ consists of all network codes, then we simply refer to the above quantity as the *coding capacity* of the network with respect to $\mathcal{A}$. If the class $\mathcal{C}$ of network codes consists of all routing codes or all linear codes, then the coding capacity is referred to as the *routing capacity* or *linear coding capacity*, respectively. The coding capacity of a given network is said to be *achievable* if there is some fractional (k, n) solution for the network for which k/n equals the capacity.

Ahlswede, Cai, Li, and Yeung [1] showed that for a general network, the coding capacity can be larger than the routing capacity, and in fact, even the linear coding capacity can be larger than the routing capacity. Li, Yeung, and Cai [9] showed in the special case of a multicast network (i.e., a network with a single source and each receiver demanding all messages), the coding capacity and the linear coding capacity are equal. It was shown in [3] that for all networks the coding capacity is independent of the alphabet size. The basic idea is that for any network coding rate that can be achieved by a fractional code over a particular alphabet, an arbitrarily close coding rate can also be achieved over any other alphabet using a different fractional code whose message dimensions and edge dimensions are sufficiently large. Clearly, the routing capacity is also independent of the alphabet size. However, it was shown in [4] that the linear coding capacity of a network can depend on the alphabet size and the largest linear coding capacity of a network over any finite-field alphabet can be smaller than the network's coding capacity. It was also shown in [3] that the routing capacity is always rational, achievable, and computable by an algorithm. The algorithm given in [3] is not particularly efficient, as its purpose was merely to establish computability (whereas computability remains open for the general coding capacity and linear coding capacity of a network).

For an undirected network (i.e., using undirected edges for information flow), the coding capacity can be larger than the routing capacity. However, for undirected networks where each message is demanded by exactly one receiver (called "multiple unicast"), it is presently unknown whether the coding capacity

Manuscript received March 12, 2005; revised November 20, 2005. This work was supported by the Institute for Defense Analyses, the National Science Foundation, and the University of California, San Diego Center for Wireless Communications.

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, 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).

Communicated by N. Cai, Guest Editor.

Digital Object Identifier 10.1109/TIT.2006.874405

can be larger than the routing capacity. Li and Li [8] and Harvey, Kleinberg, and Rasala Lehman [5] have conjectured that the two capacities must always be equal for multiple unicast undirected networks. This conjecture remains an open question, although it was shown to be true for certain special undirected bipartite networks by Rasala Lehman [7].

It has been an open question whether the coding capacity of an arbitrary network must always be achievable by the network. We answer this question in the negative.

Specifically, we give an example of a network whose coding capacity equals 1 and we demonstrate that this capacity cannot be achieved by any network code (Corollary 12). As part of the proof, we construct two subnetworks, one of which (i.e., $\mathcal{N}_2$) is solvable if and only if the alphabet size is odd (Corollary 9), and the other of which (i.e., $\mathcal{N}_3$) is solvable if and only if the alphabet size is a power of 2 (Corollary 11). Both subnetworks $\mathcal{N}_2$ and $\mathcal{N}_3$ have coding capacity equal to 1. To establish the solvability of these networks in terms of their alphabet sizes, we algebraically characterize all possible solutions to the subnetworks in terms of six permutations and certain finite Abelian group operations. We note that networks $\mathcal{N}_2$ and $\mathcal{N}_3$ (and a closely related network $\mathcal{N}_1$) were first introduced in [4].

No linearity assumptions are made in the present paper. We note, however, that it can be shown that the subnetwork $\mathcal{N}_2$ has linear coding capacity equal to $5/6$ over even-characteristic fields and 1 over odd-characteristic fields, and it can be shown from [3] to have routing capacity equal to $1/3$. Also, it was shown in [4] that the subnetwork $\mathcal{N}_3$ has linear coding capacity equal to $4/5$ over odd-characteristic finite fields and 1 over even-characteristic fields, and it was shown in [3] to have routing capacity equal to $2/3$.

II. Main Result

We say that two networks are *disjoint* if their node sets are disjoint. A network is said to be the *union* of two disjoint networks if its underlying graph is the union of the underlying graphs of the two disjoint networks (in particular, this means that there are no edges from one of the two subnetworks to the other), and if all the assignments of messages and demands to nodes remain unchanged.

Lemma 1 below was proven in [3].

Lemma 1: The coding capacity of a network is independent of the alphabet size.

Lemma 2: Every rational coding rate less than the coding capacity is achievable.

Proof: Let k/n be a rational number less than the coding capacity. Then there exists a (k', n') rational coding solution over some alphabet such that $k/n < k'/n'$. Therefore, $nk' > kn'$. If we repeat the (k', n') solution k times, we get a (kk', kn') fractional coding solution to the same network. By increasing the edge dimension from kn' to $k'n$, one obtains a $(kk', k'n)$ fractional coding solution. Thus, k/n is an achievable coding rate. □

We note that it is easy to generalize Lemma 2 to show that if a network has a (k, n) fractional coding solution over some alphabet and $k/n \geq k'/n'$, then the network has a (k', n') fractional coding solution over some (possibly different) alphabet.

Proposition 3: If two disjoint solvable networks are never solvable over the same alphabet, then their union has coding capacity equal to 1, which cannot be achieved.

Proof: Suppose two disjoint networks N_1 and N_2 are solvable on alphabets A and B, respectively, where $|A| \neq |B|$, but are never solvable over the same alphabet. The solvability of networks N_1 and N_2 implies that the coding capacities of the networks are at least 1 over alphabets A and B, respectively. Thus, by Lemma 1, the coding capacity of N_2 is also at least 1 over alphabet A. So, for any $\epsilon > 0$, there exists a (k, n) fractional coding solution over alphabet A to the network N_2 such that

$$\frac{k}{n} > 1 - \epsilon.$$

If $k > n$, then the (k, n) solution of N_2 over A induces a (k, k) solution of N_2 over A, by leaving $k - n$ components of N_2's edges unused. This implies there is a $(k, \max(k, n))$ fractional coding solution of the network N_2 over alphabet A. Likewise, since N_1 is solvable over A, it has a $(1, 1)$ fractional coding solution, which by repeating the solution k times gives a (k, k) fractional coding solution to N_1 over A. If $n > k$, then this (k, k) solution induces a (k, n) solution of N_1 over A. So the union network $N_1 \cup N_2$ has a $(k, \max(k, n))$ fractional coding solution over alphabet A. Since $k/\max(k, n) > 1 - \epsilon$, we obtain

$$\sup_{\epsilon > 0} \frac{k}{\max(k, n)} \geq 1$$

so the coding capacity of $N_1 \cup N_2$ is at least 1.

Now suppose there exists a (k, n) fractional coding solution for $N_1 \cup N_2$ over some alphabet C with $k \geq n$. Then for each network N_1 and N_2, this solution induces a (k, k) fractional coding solution over C, by leaving $k - n$ edge components unused. Such a fractional coding solution is actually a scalar solution (i.e., $k = n = 1$) to both N_1 and N_2 over an alphabet of size $|C|^k$, violating the supposition of the proposition.

Thus, any (k, n) fractional coding solution must have $k < n$ which implies 1 is not an achievable coding rate of $N_1 \cup N_2$. By Lemma 2, the coding capacity of $N_1 \cup N_2$ is therefore at most 1 (and hence equals 1); i.e., the coding capacity cannot be achieved. □

A special case of the proof of Proposition 3 is the fact that if a network has a (k, n) fractional coding solution with $k \geq n$, then the network is solvable. Thus, if the coding capacity of a network is greater than 1, then the network is solvable.

In some of the arguments to follow (in the case where $k = n = 1$), we will treat messages for networks as independent random variables distributed uniformly over the code alphabet. In calculating entropies for these variables and other variables dependent on them, the bases of all logarithms will be the alphabet size. Hence, the entropy of each individual message will be 1 and (since we are assuming unit-capacity edges) the entropy of each random variable specifying the contents of a given edge will be at most 1.

We say a set of edge values $u_1, \ldots, u_s$ in a network is *computable* from another set of edge values $v_1, \ldots, v_t$ in the same network, if there exist functions $f_i : \mathcal{A}^t \to \mathcal{A}^s$ such that $u_i = f_i(v_1, \ldots, v_t)$ for all $i = 1, \ldots, s$.

For any set of discrete random variables

$$x_1, \ldots, x_s, y_1, \ldots, y_t$$

let $H(x_1, \ldots, x_s)$ denote the (joint) entropy of $x_1, \ldots, x_s$ and let $H(x_1, \ldots, x_s \mid y_1, \ldots, y_t)$ denote the conditional entropy of $x_1, \ldots, x_s$ given $y_1, \ldots, y_t$. We will make use of the nonnegativity of entropy and also the following entropy identity:

$$H(x_1, \ldots, x_s, y_1, \ldots, y_t) = H(y_1, \ldots, y_t) + H(x_1, \ldots, x_s \mid y_1, \ldots, y_t). \quad (1)$$

(The reader is referred to [11] for the fundamental definitions and properties of entropy.) The following lemma is a basic information-theoretic result (e.g., see [11, Proposition 2.36]) whose proof we omit.

Lemma 4: Let $x_1, \ldots, x_s$, $y_1, \ldots, y_t$ be discrete random variables with only finitely many nonzero-probability outcomes. Then $H(x_1, \ldots, x_s, y_1, \ldots, y_t) = H(x_1, \ldots, x_s)$ if and only if $y_1, \ldots, y_t$ is computable from $x_1, \ldots, x_s$.

The following definition is tailored to specific networks used in the results to follow.

Definition: For a network containing messages a, b, c, and labeled edges w, x, y, z, we say that a code over an alphabet $\mathcal{A}$ has *Property* P if there exist permutations $\pi_1, \ldots, \pi_6$ of $\mathcal{A}$ and a mapping $\oplus : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ such that $(\mathcal{A}, \oplus)$ is an Abelian group and

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

Let $\mathcal{N}_1$ be the network shown in Fig. 1. The following proposition characterizes the most general form of a solution to the network $\mathcal{N}_1$.

Proposition 5: A code over an alphabet $\mathcal{A}$ is a solution for network $\mathcal{N}_1$ if and only if it satisfies Property P.

We provide a specific example to help the reader follow the proof of Proposition 5. The example is specified by a network coding solution in Table I using the ternary alphabet $\mathcal{A} = \{0, 1, 2\}$, and various quantities for this example that are used in the proof of Proposition 5 are shown in Table II. We refer back to this ternary example a number of times throughout the proof of Proposition 5. One can verify that this is indeed a solution to the network $\mathcal{N}_1$. For example, if $(a, b, c) = (2, 0, 1)$, then the truth tables indicate that $(w, x, y, z) = (2, 0, 1, 2)$, from which one can decode the following:

- $c = 1$ at node n_{12} since $(z, w) = (2, 2)$;
- $b = 0$ at node n_{13} since $(z, x) = (2, 0)$;
- $a = 2$ at node n_{14} since $(z, y) = (2, 1)$.

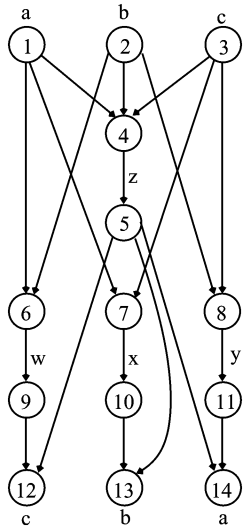

Fig. 1. The network $\mathcal{N}_1$ has sources n_1, n_2, n_3 emitting messages a, b, c, respectively, and receivers n_{12}, n_{13}, n_{14} with demands c, b, a, respectively.

TABLE I
A TERNARY NETWORK CODING SOLUTION FOR $\mathcal{N}_1$. EACH TRUTH TABLE SPECIFIES A FUNCTION (ON THE RIGHT) IN TERMS OF ITS INPUTS (ON THE LEFT). THE TOP FOUR TABLES ARE THE EDGE FUNCTIONS FOR z, w, x, y. THE BOTTOM THREE TABLES ARE THE DECODING FUNCTIONS AT NODES n_{12}, n_{13}, n_{14} FOR $\mathcal{N}_1$

abc	z
000	1
001	0
002	2
010	2
011	1
012	0
020	0
021	2
022	1
100	2
101	1
102	0
110	0
111	2
112	1
120	1
121	0
122	2
200	0
201	2
202	1
210	1
211	0
212	2
220	2
221	1
222	0

ab	w
00	1
01	0
02	2
10	0
11	2
12	1
20	2
21	1
22	0

ac	x
00	2
01	1
02	0
10	0
11	2
12	1
20	1
21	0
22	2

bc	y
00	0
01	1
02	2
10	2
11	0
12	1
20	1
21	2
22	0

zw	c
00	2
10	1
20	0
01	1
11	0
21	2
02	0
12	2
22	1

zx	b
00	1
10	2
20	0
01	0
11	1
21	2
02	2
12	0
22	1

zy	a
00	2
10	0
20	1
01	0
11	1
21	2
02	1
12	2
22	0

Proof: The "if" direction of the proof is easily verified by specifying the required edge functions and decoding functions. The demands are decoded as follows:

$$\begin{aligned} n_{12} : c &= \pi_3^{-1}\left(z \ominus \pi_4^{-1}(w)\right) \\ n_{13} : b &= \pi_2^{-1}\left(z \ominus \pi_5^{-1}(x)\right) \\ n_{14} : a &= \pi_1^{-1}\left(z \ominus \pi_6^{-1}(y)\right). \end{aligned}$$

TABLE II
SOME INTERMEDIATE FUNCTIONS IN THE PROOF OF PROPOSITION 5 FOR THE GIVEN TERNARY EXAMPLE

$a'b'$	h_{12}
00	2
01	1
02	0
10	1
11	0
12	2
20	0
21	2
22	1

$a'c'$	h_{13}
00	1
01	2
02	0
10	2
11	0
12	1
20	0
21	1
22	2

$b'c'$	h_{23}
00	0
01	2
02	1
10	2
11	1
12	0
20	1
21	0
22	2

$c'w$	g_3
00	2
10	0
20	1
01	1
11	2
21	0
02	0
12	1
22	2

rs	g'_3
00	0
10	1
20	2
01	1
11	2
21	0
02	2
12	0
22	1

π_1	(0 1 2)
π_2	()
π_3	(1 2)
π_4	(0 2)
π_5	(0 1 2)
π_6	(1 2)

Now to prove the "only if" direction, suppose that we have a solution to the network. We will show that the solution has Property P. Fix an arbitrary element of the alphabet $\mathcal{A}$ and call it 0. (In our ternary example, this corresponds to the element 0 in $\mathcal{A} = \{0,1,2\}$.)

We have

$$z = f(a,b,c) \tag{2}$$

for some function f. On the other hand, since node n_{14} has inputs z, y and demands the message a, one can compute a from y, z and hence from b, c, z; this means that, if b and c are held fixed, then $f(a,b,c)$ is a one-to-one function of a. In particular, we define

$$\pi_1(a) = f(a,0,0) \tag{3}$$

and note that π_1 is a one-to-one function from $\mathcal{A}$ to $\mathcal{A}$, i.e., a permutation. Similarly, b can be computed from a, c, z and c can be computed from a, b, z, so, if any two arguments of f are held fixed, then f is a one-to-one function of the other argument. So the functions π_2 and π_3 defined by

$$\pi_2(b) = f\left(\pi_1^{-1}(0), b, 0\right) \tag{4}$$

$$\pi_3(c) = f\left(\pi_1^{-1}(0), 0, c\right) \tag{5}$$

are permutations of $\mathcal{A}$. Note that

$$\pi_2(0) = \pi_3(0) = 0 \quad \text{[from (3)-(5)]}.$$

(In our ternary example, one can verify that $\pi_1 = (0\,1\,2)$, $\pi_2 = ()$, and $\pi_3 = (1\,2)$, where the permutations are written in cycle decomposition form.)

For the rest of this proof we will use the notation

$$a' = \pi_1(a) \tag{6}$$

$$b' = \pi_2(b) \tag{7}$$

$$c' = \pi_3(c). \tag{8}$$

Define the function f' by

$$f'(r,s,t) = f\left(\pi_1^{-1}(r), \pi_2^{-1}(s), \pi_3^{-1}(t)\right). \tag{9}$$

Then we have

$$z = f'(a',b',c') \quad \text{[from (2), (6)-(9)]} \tag{10}$$

and

$$f'(r,0,0) = f'(0,r,0) = f'(0,0,r) = r \quad \text{[from (3)-(5), (9)]}. \tag{11}$$

Since w is computed from a and b, it can be computed from a' and b'; write

$$w = h_{12}(a',b') \tag{12}$$

for some function h_{12}. Similarly, write

$$x = h_{13}(a',c')$$
$$y = h_{23}(b',c').$$

(In our ternary example, one can verify that h_{12}, h_{13}, and h_{23} are given by the truth tables in Table II.)

We now start using entropy arguments. By examining Fig. 1, it can be seen that if we know w, y, and z, then we can compute both c and a. Once c and a are known, this determines x. Then, since x and z are known, we can compute b. So, viewing these as random variables, we have

$$H(w,y,z) \geq H(a,b,c). \tag{13}$$

Since a, b, c are independent and

$$H(a) = H(b) = H(c) = 1$$

we obtain

$$H(a,b,c) = H(a) + H(b) + H(c) = 3. \tag{14}$$

Each of w, y, z is an edge variable and hence has entropy at most 1. Now

$$\begin{aligned} 3 &\leq H(w,y,z) \\ &\leq H(w) + H(y) + H(z) \\ &\leq 1 + 1 + 1 = 3 \quad \text{[from (13), (14)]} \end{aligned}$$

so equality must hold throughout and we have

$$H(w) = H(y) = H(z) = 1. \tag{15}$$

A similar argument starting with w, x, z shows that

$$H(x) = 1. \tag{16}$$

We will now begin an argument based on w to define functions g'_3, h'_{12}, and π_4 with certain properties. We will claim later that this argument can be duplicated for x and y. Since w is a function of a and b, which are (jointly) independent of c, we have that w is independent of c. Therefore,

$$\begin{aligned} H(c,w) &= H(w) + H(c|w) \quad \text{[from (1)]} \\ &= H(w) + H(c) \\ &= 2. \end{aligned} \tag{17}$$

But c is computable from w and z, so we get

$$\begin{aligned} H(c,w,z) &= H(w,z) \quad \text{[from Lemma 4]} \\ &\leq H(w) + H(z) \\ &\leq 2 \quad \text{[from (15)]} \\ &= H(c,w) \quad \text{[from (17)]}. \end{aligned}$$

This implies that z is computable from c and w by Lemma 4. Hence, z is computable from c' and w, so we can write

$$z = g_3(c', w) = g_3\left(c', h_{12}(a', b')\right) \quad \text{[from (12)]} \tag{18}$$
$$= f'(a', b', c') \quad \text{[from (10)]} \tag{19}$$

for some function g_3.

We also have

$$g_3(0, h_{12}(r, 0)) = f'(r, 0, 0) = r \quad \text{[from (9), (11)]}. \tag{20}$$

Now define

$$\pi_4(r) = h_{12}(r, 0) \tag{21}$$

and note that π_4 is one-to-one by (20), so π_4 is a permutation of $\mathcal{A}$ and $\pi_4^{-1}(s) = g_3(0, s)$. (In our ternary example, one can verify that $\pi_4 = (0\ 2)$.) Let

$$h'_{12}(r, s) = \pi_4^{-1}(h_{12}(r, s)) \tag{22}$$
$$g'_3(r, s) = g_3(r, \pi_4(s)). \tag{23}$$

Then we have

$$f'(a', b', c') = g'_3\left(c', h'_{12}(a', b')\right) \quad \text{[from (19), (23), (22)]} \tag{24}$$
$$w = \pi_4\left(h'_{12}(a', b')\right) \quad \text{[from (12), (22)]} \tag{25}$$
$$h'_{12}(r, 0) = r \quad \text{[from (21), (22)]} \tag{26}$$
$$g'_3(0, r) = r \quad \text{[from (23), (21), (20)]}. \tag{27}$$

We also get

$$g'_3(r, 0) = g'_3\left(r, h'_{12}(0, 0)\right)$$
$$= f'(0, 0, r) = r \quad \text{[from (26), (24), (11)]}$$
$$h'_{12}(0, r) = g'_3\left(0, h'_{12}(0, r)\right)$$
$$= f'(0, r, 0) = r \quad \text{[from (27), (24), (11)]}.$$

The same arguments can be applied to x and y (i.e., starting after (16)), so we get functions $g'_2, g'_1, h'_{13}, h'_{23}$ and permutations π_5, π_6 such that

$$f'(a', b', c') = g'_2\left(b', h'_{13}(a', c')\right)$$
$$= g'_1\left(a', h'_{23}(b', c')\right) \tag{28}$$
$$x = \pi_5\left(h'_{13}(a', c')\right) \tag{29}$$
$$y = \pi_6\left(h'_{23}(b', c')\right) \tag{30}$$
$$h'_{13}(r, 0) = h'_{13}(0, r) = h'_{23}(r, 0) = h'_{23}(0, r) = r \tag{31}$$
$$g'_2(r, 0) = g'_2(0, r) = g'_1(r, 0) = g'_1(0, r) = r.$$

(In our ternary example, one can verify that $\pi_5 = (0\ 1\ 2)$ and $\pi_6 = (1\ 2)$.)

Let us also define

$$h'_{ji}(r, s) = h'_{ij}(s, r) \tag{32}$$

for $(i, j) = (1, 2), (1, 3), (2, 3)$. Then we can say that, if (i, j, k) is any permutation of $(1, 2, 3)$, then

$$f'(r_1, r_2, r_3) = g'_i\left(r_i, h'_{jk}(r_j, r_k)\right) \tag{33}$$
$$g'_i(r, 0) = g'_i(0, r) = r \tag{34}$$
$$h'_{jk}(r, 0) = h'_{jk}(0, r) = r.$$

(We are using the fact that, as a, b, c vary, the triple (a', b', c') assumes all values $(r_1, r_2, r_3) \in \mathcal{A}^3$.)

We now have

$$f'(r, s, 0) = g'_1\left(r, h'_{23}(s, 0)\right) = g'_1(r, s) \quad \text{[from (28), (31)]}$$
$$f'(r, s, 0) = g'_3\left(0, h'_{12}(r, s)\right) = h'_{12}(r, s) \quad \text{[from (24), (34)]}.$$

By applying f' to the other five orderings of r, s, 0 as well, we get

$$g'_i(r, s) = h'_{ij}(r, s)$$

for all i and j. This then gives

$$h'_{ij}(r, s) = g'_i(r, s) = h'_{ik}(r, s)$$

from which we can get

$$h'_{ji}(r, s) = h'_{ij}(s, r) = h'_{ik}(s, r) = h'_{ki}(r, s) \quad \text{[from (32)]}. \tag{35}$$

Putting these together, we get

$$h'_{12} = h'_{13} = h'_{23} = h'_{21} = h'_{31} = h'_{32}$$

and $g'_i = h'_{ij}$, so all nine of the functions g'_i and h'_{ij} are the same.

Define $\oplus : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ by

$$r \oplus s = g'_1(r, s),$$

so that in fact

$$r \oplus s = g'_i(r, s) = h'_{ij}(r, s) \tag{36}$$

for any i and j. (In our ternary example, $\oplus$ is addition modulo 3.) We now have

$$w = \pi_4\left(h'_{12}(a', b')\right)$$
$$= \pi_4(\pi_1(a) \oplus \pi_2(b)) \quad \text{[from (25), (36), (6), (7)]}$$
$$x = \pi_5\left(h'_{13}(a', c')\right)$$
$$= \pi_5(\pi_1(a) \oplus \pi_3(c)) \quad \text{[from (29), (36), (6), (8)]}$$
$$y = \pi_6\left(h'_{23}(b', c')\right)$$
$$= \pi_6(\pi_2(b) \oplus \pi_3(c)) \quad \text{[from (30), (36), (7), (8)]}$$
$$z = f'(a', b', c')$$
$$= g'_1\left(a', h'_{23}(b', c')\right)$$
$$= \pi_1(a) \oplus (\pi_2(b) \oplus \pi_3(c)) \quad \text{[from (10), (28), (36), (6)-(8)]}$$

so in order to establish that the network solution has Property P, it remains to show that $(\mathcal{A}, \oplus)$ is an Abelian group.

We have

$$r \oplus s = h'_{21}(r, s) = h'_{12}(s, r) = s \oplus r \quad \text{[from (36), (32)]} \tag{37}$$

so $\oplus$ is commutative. Now we get

$$r \oplus (s \oplus t) = g'_1\left(r, h'_{23}(s, t)\right)$$
$$= f'(r, s, t) \quad \text{[from (36), (33)]} \tag{38}$$
$$(r \oplus s) \oplus t = t \oplus (r \oplus s)$$
$$= g'_3\left(t, h'_{12}(r, s)\right)$$
$$= f'(r, s, t) \quad \text{[from (37), (36), (33)]} \tag{39}$$

so $\oplus$ is associative. Since

$$0 \oplus r = r \oplus 0 = g'_1(r, 0) = r \quad \text{[from (37), (34), (36)]}$$

0 is an identity element for $\oplus$. Finally, since f has the property that it is one-to-one on its third argument when the other two

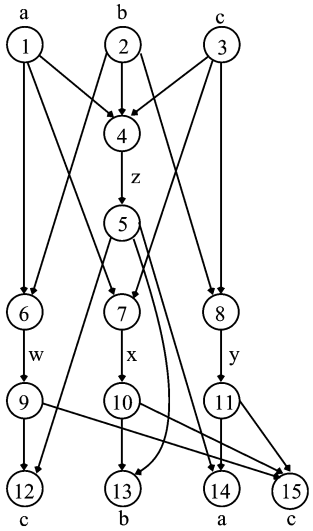

Fig. 2. The network $\mathcal{N}_2$ has sources n_1, n_2, n_3 emitting messages a, b, c, respectively, and receivers n_{12}, n_{13}, n_{14}, n_{15} with demands c, b, a, c, respectively.

are held fixed (as noted in the comment before (4)), f' also has this property. So, for any fixed $r \in \mathcal{A}$, the function $\mu_r(s) = f'(r, 0, s)$ is one-to-one, so it must map onto $\mathcal{A}$; hence, we can find an s such that $\mu_r(s) = f'(r, 0, s) = 0$. For this s we have

$$r \oplus s = h'_{13}(r, s) \quad \text{[from (36)]} \tag{40}$$
$$= g'_2(0, h'_{13}(r, s)) \quad \text{[from (34)]} \tag{41}$$
$$= f'(r, 0, s) = 0 \quad \text{[from (28)]}. \tag{42}$$

Therefore, every element $r \in \mathcal{A}$ has an inverse, so we are done. □

The following lemmas are due respectively to Cauchy and Lagrange (e.g., [2]).

Lemma 6: Every prime divisor of a finite group's order is the order of some element of the group.

Lemma 7: A finite group's order is divisible by the order of every element of the group.

Let $\mathcal{N}_2$ be the network shown in Fig. 2.

Proposition 8: A code over an alphabet $\mathcal{A}$ is a solution for network $\mathcal{N}_2$ if and only if it satisfies Property P and the group $(\mathcal{A}, \oplus)$ has no elements of order 2.

Proof: Suppose $\mathcal{N}_2$ has a solution over alphabet $\mathcal{A}$. The network $\mathcal{N}_2$ is just the network $\mathcal{N}_1$ with the extra demand node n_{15}, so any solution to $\mathcal{N}_2$ is a solution to $\mathcal{N}_1$, and hence must satisfy Property P. Suppose the group $(\mathcal{A}, \oplus)$ has a nonzero element r of order 2. Then the two different message triples

$$(a, b, c) = \left(\pi_1^{-1}(0), \pi_2^{-1}(0), \pi_3^{-1}(0)\right)$$
$$(a, b, c) = \left(\pi_1^{-1}(r), \pi_2^{-1}(r), \pi_3^{-1}(r)\right)$$

have different values for c, but since Property P implies

$$w = \pi_4(\pi_1(a) \oplus \pi_2(b)) = \pi_4(r \oplus r) = \pi_4(0)$$
$$x = \pi_5(\pi_1(a) \oplus \pi_3(c)) = \pi_5(r \oplus r) = \pi_5(0)$$
$$y = \pi_6(\pi_2(b) \oplus \pi_3(c)) = \pi_6(r \oplus r) = \pi_6(0)$$

both of the (a, b, c) message triples give

$$(w, x, y) = (\pi_4(0), \pi_5(0), \pi_6(0)),$$

so the extra demand is not met (one cannot distinguish between $c = \pi_3^{-1}(0)$ and $c = \pi_3^{-1}(r)$, and thus cannot determine c given w, x, y). Therefore, for a solution to the present network, $(\mathcal{A}, \oplus)$ cannot have an element or order 2.

Conversely, suppose a code for $\mathcal{N}_2$ satisfies Property P and the group $(\mathcal{A}, \oplus)$ has no elements of order 2. Then, by Proposition 5, the demands at nodes n_{12}, n_{13}, n_{14} are met, so it suffices to show the demand at node n_{15} is met.

We have that $r \oplus r = s \oplus s$ implies $(s \ominus r) \oplus (s \ominus r) = 0$ and hence $s \ominus r = 0$, so $s = r$. Thus, r is computable from $r \oplus r$ since the map $r \mapsto r \oplus r$ is one-to-one. Given w, x, y one can get

$$\pi_3(c) \oplus \pi_3(c) = \pi_5^{-1}(x) \oplus \pi_6^{-1}(y) \ominus \pi_4^{-1}(w),$$

so one can compute $\pi_3(c)$ and c, thus satisfying the demand at node n_{15}. □

Corollary 9: The network $\mathcal{N}_2$ is solvable if and only if the alphabet size is odd.

Proof: First, suppose $\mathcal{N}_2$ has a solution over the alphabet $\mathcal{A}$. By Proposition 8, there is a binary operation $\oplus$ on $\mathcal{A}$ such that $(\mathcal{A}, \oplus)$ is an Abelian group with no elements of order 2. It now follows from Lemma 6 that the order of this group is not divisible by 2; in other words, $|\mathcal{A}|$ is odd.

Conversely, suppose $|\mathcal{A}|$ is odd. If we view $\mathcal{A}$ as the set $\{0, 1, \ldots, |\mathcal{A}| - 1\}$ and the binary operation $\oplus$ as integer addition modulo $|\mathcal{A}|$, then $(\mathcal{A}, \oplus)$ is an Abelian group. By Lemma 7, the order of every element of $(\mathcal{A}, \oplus)$ must divide $|\mathcal{A}|$ and thus no element can have order 2. A network code which assigns $w = a \oplus b$, $x = a \oplus c$, $y = b \oplus c$, and $z = a \oplus b \oplus c$ satisfies Property P (where π_i is the identity permutation for $i = 1, \ldots, 6$). Thus, by Proposition 8, the network $\mathcal{N}_2$ is solvable over $\mathcal{A}$. □

Let $\mathcal{N}_3$ be the network shown in Fig. 3.

Proposition 10: A code over an alphabet $\mathcal{A}$ is a solution for network $\mathcal{N}_3$ if and only if it satisfies Property P and all the nonzero elements of the group $(\mathcal{A}, \oplus)$ are of order 2.

Proof: Suppose a code over alphabet $\mathcal{A}$ satisfies Property P and all the nonzero elements of the group $(\mathcal{A}, \oplus)$ are of order 2. The extra hypothesis on $(\mathcal{A}, \oplus)$ is used to verify that x can be computed from w and y:

$$x = \pi_5\left(\pi_4^{-1}(w) \oplus \pi_6^{-1}(y)\right)$$

because the two occurrences of $\pi_2(b)$ cancel. A quick inspection shows that each output over every network node can be computed from the inputs. Also, the demands of $\mathcal{N}_3$ are met as follows:

$$n_{27} : c = \pi_3^{-1}\left(\pi_5^{-1}(x) \ominus \pi_1(a))\right)$$
$$n_{28} : b = \pi_2^{-1}\left(z \ominus \pi_5^{-1}(x)\right)$$
$$n_{28} : a = \pi_1^{-1}\left(z \ominus \pi_6^{-1}(y)\right).$$

Thus, this code is a solution over $\mathcal{A}$.

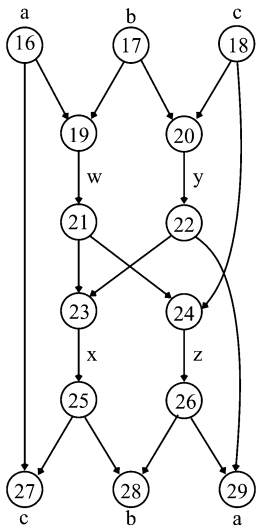

Fig. 3. The network $\mathcal{N}_3$ has sources n_{16}, n_{17}, n_{18} emitting messages a, b, c, respectively, and receivers n_{27}, n_{28}, n_{29} with demands c, b, a, respectively.

Conversely, suppose we have a solution to $\mathcal{N}_3$ over alphabet $\mathcal{A}$. We will first show that this solution (given by specifying w, x, y, z as functions of a, b, c) also gives a solution of the network $\mathcal{N}_1$ in Proposition 5. This comes down to verifying the following:

i) z is computable from a, b, c,
ii) w is computable from a, b,
iii) x is computable from a, c,
iv) y is computable from b, c,
v) c is computable from w, z,
vi) b is computable from x, z,
vii) a is computable from y, z.

Facts i), ii), iv), vi), and vii) follow from direct inspection of the network $\mathcal{N}_3$. We have that c is computable from a, x, so viewing a, b, c as independent random variables uniformly distributed over $\mathcal{A}$ gives

$$H(a,c,x) \leq H(a,x) \leq 2 = H(a,c).$$

This gives iii) by Lemma 4. From w, y, z one can compute successively x, a, b, and c, so

$$H(w,y,z) \geq H(a,b,c) = 3$$

and hence,

$$\begin{aligned} H(w,z) &= H(w,y,z) - H(y \mid w,z) \quad [\text{from (1)}] \\ &\geq H(w,y,z) - H(y) \\ &\geq 3 - 1 = 2. \end{aligned}$$

Now, since z is computable from c, w we get

$$H(c,w,z) \leq H(c,w) \leq 2 \leq H(w,z)$$

which gives v), by Lemma 4. So, our solution to $\mathcal{N}_3$ is also a solution to $\mathcal{N}_1$.

Therefore, Proposition 5 implies that our solution to network $\mathcal{N}_3$ satisfies Property P.

Now, suppose the group $(\mathcal{A}, \oplus)$ has a nonzero element not of order 2. That is, suppose there exists an $r \in \mathcal{A}$ such that $r \oplus r \neq 0$. Then the two difference message triples

$$\begin{aligned} (a,b,c) &= \left(\pi_1^{-1}(0), \pi_2^{-1}(0), \pi_3^{-1}(0)\right) \\ (a,b,c) &= \left(\pi_1^{-1}(r), \pi_2^{-1}(\ominus r), \pi_3^{-1}(r)\right) \end{aligned}$$

would both give $w = \pi_4(0)$ and $y = \pi_6(0)$, but one would give $x = \pi_5(0)$ and the other would give $x = \pi_5(r \oplus r) \neq \pi_5(0)$. This is impossible because x is computed from w and y here. So all $r \in \mathcal{A}$ satisfy $r \oplus r = 0$; that is, all elements of $\mathcal{A}$ have order 1 or 2. □

Corollary 11: The network $\mathcal{N}_3$ is solvable if and only if the alphabet size is a power of 2.

Proof: First, suppose $\mathcal{N}_3$ has a solution over the alphabet $\mathcal{A}$. Thus, by Proposition 10, the solution satisfies Property P and every nonzero element of $(\mathcal{A}, \oplus)$ is of order 2. By Lemma 6, for every prime divisor p of $|\mathcal{A}|$, the group $(\mathcal{A}, \oplus)$ has an element of order p. Thus, the only prime divisor of $|\mathcal{A}|$ is 2, so $|\mathcal{A}|$ is a power of 2.

Conversely, suppose $|\mathcal{A}| = 2^i$ for some $i \geq 1$. If we view A as the set $\{0, \ldots, 2^i - 1\}$ and the binary operation $\oplus$ as bit-wise addition modulo 2, then $(\mathcal{A}, \oplus)$ is the Abelian group $\mathbf{Z}_2^i$ and all the nonidentity elements have order 2. A network code which assigns $w = a \oplus b$, $x = a \oplus c$, $y = b \oplus c$, and $z = a \oplus b \oplus c$ satisfies Property P (where π_i is the identity permutation for $i = 1, \ldots, 6$). Thus, by Proposition 10, the network $\mathcal{N}_3$ is solvable over $\mathcal{A}$. □

Corollary 12: There exists a directed acyclic network whose coding capacity is not achievable.

Proof: Follows immediately from Proposition 3 and Corollaries 9 and 11. □

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 two networks $\mathcal{N}_2$ and $\mathcal{N}_3$ 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 $\mathcal{N}_2$ and $\mathcal{N}_3$. This will create a connected network having the same property of not being able to achieve its coding capacity.

III. Conclusion and Open Questions

The coding capacity of a network can, by definition, be arbitrarily closely approximated by using sufficiently long message dimensions and edge capacities. However, this paper has demonstrated that the coding capacity of a network might not be exactly achievable by any particular network code. This is in contrast to what is known about routing capacity.

There are a number of interesting fundamental open questions concerning the coding capacities of networks. We briefly mention some of these here.

Does there exist an algorithm for computing the coding capacity of an arbitrary network? Can the coding capacity of a network be irrational? If the coding capacity of a network is achiev-

able, then clearly it is rational since it would equal the ratio of the message dimension and edge dimension which achieve the capacity. However, as demonstrated in this paper, some networks may not have an achievable coding capacity, so conceivably they might have an irrational capacity. The particular network shown in this paper not to have an achievable coding capacity, turned out, in fact, to have a rational coding capacity (i.e., 1).

Does there exist a multiple unicast undirected network for which the coding capacity is larger than the routing capacity [5], [8]? If not, then there would be no capacity-type advantage to using network coding for such undirected networks, although there could still possibly be a complexity improvement.

Each of the above questions remains open if the phrase "coding capacity" is replaced by "linear coding capacity."

The solutions to networks $\mathcal{N}_1, \mathcal{N}_2$, and $\mathcal{N}_3$ were characterized using Property P. One might generalize the notion of Property P so that certain edge functions in a code can be computed in terms of arbitrary permutations of sums of fixed permutations of the inputs, where the "sum" is an Abelian group operation. It would be interesting to discover if this idea has further utility.

The authors discovered after completion of the review process that portions of the proofs of some of the results can be deduced from results in [10, pp. 179,185], which were given in a context that did not discuss network coding.

REFERENCES

[1] 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.

[2] P. B. Bhattacharya, S. K. Jain, and S. R. Nagpaul, *Basic Abstract Algebra*. New York: Cambridge Univ. Press, 1986.

[3] 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.

[4] 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.

[5] N. J. A. Harvey, R. D. Kleinberg, and A. R. Lehman, "Comparing Network Coding With Multicommodity Flow for the k-Pairs Communication Problem," M.I.T. LCS, Tech. Rep. 964, 2004.

[6] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proc. 41st Annu. Allerton Conf. Communication, Control and Computing*, Monticello, IL, Oct. 2003.

[7] A. R. Lehman, "Network Coding," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 2005. Available [Online] at http://theory.lcs.mit.edu/~arasala/thesis.pdf.

[8] Z. Li and B. Li, "Network coding in undirected networks," in *Proc. 38th Annu. Conf. Information Sciences and Systems (CISS)*, Princeton, NJ, Mar. 2004.

[9] 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.

[10] F. Matúš, "Matroid representations by partitions," *Dis. Math.*, vol. 203, pp. 169–194, 1999.

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