receive antennas, neglecting the rate loss factor. We also show that the STTC [12] is a special case of our proposed high-rate STTC.

REFERENCES

[1] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1073–1096, May 2003.

[2] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time block codes for high data rate wireless communication:Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, pp. 744–765, Mar. 1998.

[3] S. Baro, G. Bauch, and A. Hansmann, "Improved codes for space-time trellis-coded modulation," *IEEE Commun. Lett.*, vol. 4, pp. 20–22, Jan. 2000.

[4] Q. Yan and R. S. Blum, "Optimal space-time convolutional codes," in *Proc. IEEE WCNC'00*, Chicago, IL, Sep. 2000, pp. 1351–1355.

[5] D. M. Ionescu, K. K. Mukkavilli, Z. Yan, and J. Lilleberg, "Improved 8- and 16- state space-time codes for 4-PSK with 2 transmit antennas," *IEEE Commun. Lett.*, vol. 5, pp. 301–303, Jul. 2001.

[6] Z. Chen, B. Vucetic, Jinhong, and K. L. Lo, "Space-time trellis codes with two, three and four transmit antennas in quasi-static fading channel," in *Proc. IEEE ICC*, 2002, vol. 3, pp. 1589–1595.

[7] T. A. Narayanan and B. S. Rajan, "A general construction of space-time trellis codes for PSK signal sets," in *Proc. IEEE GLOBECOM*, 2003, pp. 1978–1983.

[8] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telcomm.*, vol. 10, no. 6, pp. 585–595, Nov.-Dec. 1999.

[9] S. Tavildar and P. Vishwanath, "Permutation codes: Achieving the diversity-multiplexing tradeoff," in *ISIT 2004*, Chicago, IL, Jun.-Jul. 27-2, 2004.

[10] ——, "Approximately universal codes over slow fading channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3233–3258, Jul. 2006.

[11] P. Elia, K. R. Kumar, S. Pawar, P. V. Kumar, and H.-F. Lu, "Explicit space–time codes achieving the diversity-multiplexing gain tradeoff," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3869–3884, Sep. 2006.

[12] R. Vaze, V. Shashidhar, and B. S. Rajan, "A high-rate generalized coded delay diversity scheme and its diversity-multiplexing trade-off," in *Proc. IEEE Int. Conf. Commun. (ICC 2005)*, Seoul, Korea, May 15-19, 2005.

[13] V. Shashidhar, B. S. Rajan, and P. V. Kumar, "Asymptotic-information lossless designs and diversity-multiplexing tradeoff," in *Proc. IEEE GLOBECOM 2004*, Dallas, TX, Nov.-Dec. 29-3, 2004.

[14] P. Elia, K. R. Kumar, S. A. Pawar, P. V. Kumar, and H.-F. Lu, "Explicit, space-time codes that achieve the diversity-multiplexing gain tradeoff," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Sep. 4-9, 2005, pp. 896–900.

[15] B. A. Sethuraman, B. S. Rajan, and V. Shashidhar, "Full-diversity, high-rate space-time block codes from division algebras," *IEEE Trans. Inf. Theory,* , ser. Special Issue on Space-Time Transmission, Reception, Coding and Signal Design, vol. 49, no. 10, pp. 2596–2616, Oct. 2003.

[16] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "The golden code: A 2 × 2 full-rate space-time code with non-vanishing determinants," *IEEE Trans. Inf. Theory*, vol. 51, Apr. 2005.

[17] N. Jacobson, *Basic Algebra I*, Second ed. New York: W. H. Freeman, 1985.

Nonreversibility and Equivalent Constructions of Multiple-Unicast Networks

Randall Dougherty and Kenneth Zeger, *Fellow, IEEE*

***Abstract*—We prove that for any finite directed acyclic network, there exists a corresponding multiple-unicast network, such that for every alphabet, each network is solvable if and only if the other is solvable, and, for every finite-field alphabet, each network is linearly solvable if and only if the other is linearly solvable. The proof is constructive and creates an extension of the original network by adding exactly $s + 5m(r-1)$ new nodes where, in the original network, m is the number of messages, r is the average number of receiver nodes demanding each source message, and s is the number of messages emitted by more than one source. The construction is then used to create a solvable multiple-unicast network which becomes unsolvable over every alphabet size if all of its edge directions are reversed and if the roles of source-receiver pairs are reversed.**

***Index Terms*—Flow, multiple unicast, network coding.**

I. INTRODUCTION

A *network* here will refer to a finite, directed, acyclic multigraph, some of whose nodes are information sources or receivers (e.g., see [18]). Associated with the sources are *messages*, which are assumed to be arbitrary elements of a fixed finite alphabet of size at least 2. At any node in the network, each out-edge carries an alphabet symbol which is a function (called an *edge function*) of the symbols carried on the in-edges to the node, and/or a function of the node's message symbols 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 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 one symbol can travel across each edge). Throughout this correspondence, if a network node in a figure is labeled by say x (inside a circle), then we refer to the node as n_x and we refer to an edge connecting n_x and n_y as $e_{x,y}$.

A network *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. A network *solution* is a network code which results in every receiver being able to compute its demands via its demand functions. A network is said to be *solvable* if it has a solution over some alphabet. A network is *linearly solvable* over a particular finite-field alphabet if it has a solution consisting of only linear edge functions and linear decoding functions over the field. A *multiple-unicast* network is a network for which every source message is emitted by exactly one source node and is demanded by exactly one receiver node. Multiple-

Manuscript received September 17, 2005; revised July 7, 2006. 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. The material in this correspondence was presented in part at the 43rd Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 2005.

R. Dougherty is with the Center for Communications Research, San Diego, CA 92121-1969 USA (e-mail: rdough@ccrwest.org).

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 R. W. Yeung, Associate Editor for Shannon Theory.

Digital Object Identifier 10.1109/TIT.2006.883634

unicast networks thus consist of communications between collections of pairs of nodes.

The solvability and linear solvability of networks have been a subject of interest (e.g., [3], [4], [6], [11], [12], [15], [16]). For example, it was shown in [12] that solvable multicast networks are always linearly solvable. The class of multiple-unicast networks has also been studied in various contexts (e.g., [1], [7], [8], [13], [14]).

We prove that for any network, there exists a corresponding multiple-unicast network, such that for every alphabet, each network is solvable if and only if the other is solvable, and, for every finite-field alphabet, each of the two networks is linearly solvable if and only if the other is linearly solvable (Theorem II.1). The proof is constructive and creates an extension of the original network by adding exactly $s + 5m(r-1)$ new nodes where, in the original network, m is the number of messages, r is the average number of receiver nodes demanding each source message, and s is the number of messages emitted by more than one source.

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 which emits messages in $\mathcal{N}$, instead demands the same messages in $\mathcal{N}'$.
4) Each node which 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 [9], [10], [17], that all linearly solvable multiple-unicast networks are linearly reversible over the same alphabet. In [9], [10], an elegant "duality" principle is given, connecting algebraic coding theory and linearly reversible networks, and applications of reversibility are discussed. In [17], a network is given which has a binary (nonlinear) solution but whose reverse does not have a binary solution.

However, it has been an open question whether a solvable network could be nonreversible (i.e., over all alphabets). Clearly (in light of the results in [9], [10], [17]), to achieve such a result, one would need to use a network which never has a linear solution over any finite field alphabet.

We modify a solvable network constructed in [4] in such a way that it becomes unsolvable (Lemma III.5). We show that the solvable network and the unsolvable network are either both reversible or both not reversible. Thus, at least one of the two networks demonstrates the existence of a solvable nonreversible network. We then modify these two networks, using the construction presented in the first part of this paper, so that they become multiple-unicast, while preserving the solvability properties (Corollary III.9). This proves that there exists a solvable nonreversible multiple-unicast network. Finally, we reveal which of the two candidate solvable nonreversible multiple-unicast networks is the true one (Theorem III.10).

II. Modifying Arbitrary Networks Into Multiple-Unicast Networks

In this section, we give a construction which creates a new network from an arbitrary network containing at least some message demanded by more than one receiver. The new network has one fewer receivers demanding a particular message in the original network. Also, the new network is solvable if and only if the original network is solvable, and this property holds for linear solvability as well. As a result, if the construction is iteratively applied to each new network until no source messages are demanded by more than one receiver, then a multiple-unicast network is achieved with the same solvability as the original network.

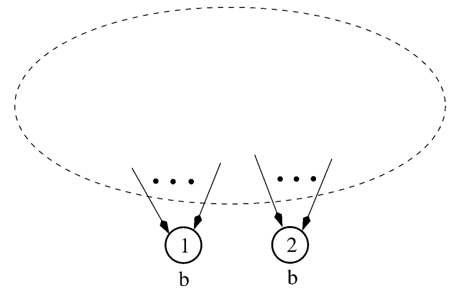

Fig. 1. An arbitrary network with two nodes n_1 and n_2 each demanding message b.

Definition: Two networks $\mathcal{N}$ and $\mathcal{N}'$ are CSLS-equivalent[1] if the following two conditions hold.

1) For any alphabet $\mathcal{A}$, $\mathcal{N}$ is solvable over $\mathcal{A}$ if and only if $\mathcal{N}'$ is solvable over $\mathcal{A}$.
2) For any finite field F and any positive integer k, $\mathcal{N}$ is vector solvable over F in dimension k if and only if $\mathcal{N}'$ is vector solvable over F in dimension k.

Theorem II.1: Any network is CSLS-equivalent to a multiple-unicast network.

Proof: Without loss of generality, we may assume that every source message is demanded by at least one receiver. For each message b that has multiple sources, add a new node to be the source for b with connections to all the old sources for b (which will no longer be sources for b). The new network is clearly CSLS-equivalent to the old one.

Now, if some message b is demanded at more than one node, select two such nodes n_1 and n_2 (as indicated in Fig. 1), and add a gadget consisting of five new nodes $n_{x1}, \ldots, n_{x5}$ connected as indicated in Fig. 2, to get a new network. Now n_{x1} is the source for a new message z, which is demanded at node n_{x5}.

A solution for the old network can be extended to the new network by putting

$$e_{1,x2} = e_{2,x5} = b$$
$$e_{x2,x3} = b + z$$

where $+$ is the given vector addition (in the vector linear case) or any group operation on $\mathcal{A}$ (in the arbitrary coding case). Now suppose that we have a solution to the new network. We will show that b can be computed from $e_{1,x2}$ and also from $e_{2,x5}$ (linearly in the vector linear case), which means that we get a solution to the old network with the same parameters. The solution to the new network gives us functions f, g, h (linear in the linear case) such that

$$e_{x2,x3} = f(z, e_{1,x2})$$
$$b = g(z, e_{x2,x3})$$
$$z = h(e_{x2,x3}, e_{2,x5}).$$

Fix an element α of the alphabet (in the linear case let $\alpha = 0$). We have

$$b = g(z, f(z, e_{1,x2}))$$

for any z, where $e_{1,x2}$ and b do not depend on z, so

$$b = g(\alpha, f(\alpha, e_{1,x2}))$$

[1]CSLS means "coding solvability and linear solvability."

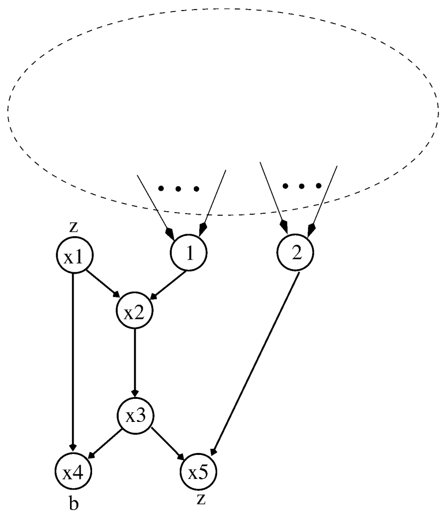

Fig. 2. An arbitrary network is modified by adding a gadget consisting of five new nodes $n_{x1}, \ldots, n_{x5}$ and some new edges from them. Node n_{x1} is a source emitting a new message z which is demanded by receiver node n_{x5}. Node n_{x4} demands message b, but nodes n_1 and n_2 no longer demand b.

and b is computable from $e_{1,x2}$. We also have

$$b = g(h(e_{x2,x3}, e_{2,x5}), e_{x2,x3}).$$

If we hold all the old messages fixed (i.e., all messages except z), then $e_{1,x2}$ and $e_{2,x5}$ are fixed and therefore,

$$e_{x2,x3} = f(z, e_{1,x2})$$

is a one-to-one (and therefore also onto) function of z, because

$$z = h(e_{x2,x3}, e_{2,x5}).$$

So, given all the old messages, we can choose z so that $e_{x2,x3} = \alpha$. In such a case,

$$b = g(h(\alpha, e_{2,x5}), \alpha)$$

so b is computable from $e_{2,x5}$.

So the new network is CSLS-equivalent to the old one; it has the same number of message demands, but the number of distinct messages has increased by one. Repeat such modifications until the number of distinct messages equals the number of message demands; then, the final network will be multiple-unicast. □

Suppose that in the original network s messages were emitted by two or more sources (e.g., if the network had three messages, which were emitted by four sources, one source, and five sources, respectively, then $s = 2$). Then s new nodes were added at the first stage of the construction. Suppose that in the original network, the ith (of m total) source message is demanded by d_i receiver nodes. Then, $d_i - 1$ iterations of the construction above will create a new network with the same solvability and where exactly one receiver node demands this message. Thus, the total number of iterations needed to avoid any messages being demanded by two or more receivers is $(d_1 - 1) + \cdots + (d_m - 1)$. Each such iteration adds five new nodes. If we define

$$r = \frac{1}{m} \sum_{i}^{m} d_i$$

then we can summarize this fact in Corollary II.2.

Corollary II.2: For any directed acyclic network, there exists a multiple-unicast network with $s + 5m(r-1)$ additional nodes which is solvable if and only if the original network is solvable, where, in the original network, m is the number of messages, r is the average number of receiver nodes demanding each source message, and s is the number of messages emitted by more than one source. The same result holds for linear solvability as well.

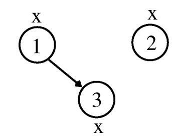

Fig. 3. An example illustrating that a linearly solvable network might not be reversible if the network is not multiple-unicast. Nodes n_1 and n_2 both are sources emitting the message x, and nodes n_3 is a receiver, demanding message x.

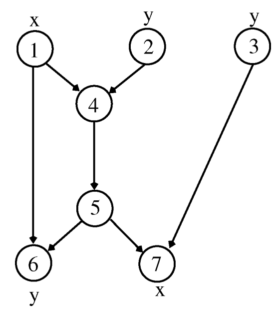

Fig. 4. An example illustrating that a linearly solvable network might not have a solvable reverse if the network is not multiple-unicast. Node n_1 is a source emitting message x, and nodes n_2 and n_3 are sources, both emitting message y. Nodes n_6 and n_7 are receivers, demanding messages y and x, respectively.

III. Nonreversibility of Multiple-Unicast Networks

The classification of reversible networks is of theoretical interest and has been considered in [9], [10], [17]. In this section, we demonstrate that not all solvable multiple-unicast networks are reversible.

First, note that a very simple example of a solvable network that is not reversible is shown in Fig. 3. The network is trivially solvable by sending message x along the edge $e_{1,3}$, and yet the network is not reversible since there is no way to get message x from n_3 to n_2. This network is redundant in the sense that the source node n_2 could be removed while retaining the network's solvability. If n_2 is removed, then the network becomes reversible too.

One could, more specifically, consider the reversibility of *minimal* solvable networks, namely, those for which no edge or source node can be removed without causing the network to become unsolvable. However, the following small example demonstrates the difficulty with such an approach.

Fig. 4 gives an example of a minimal network which is linearly solvable over every alphabet size (by taking $e_{4,5} = x + y$), and yet the network is not reversible (since in the reverse network, the demand y at n_3 cannot be met).

An alternative direction to pursue for classifying reversibility is to examine multiple-unicast networks. In [9], [10], [17], the interesting result that every linearly solvable network is linearly reversible was shown. In [17], a (nonlinearly) solvable network was demonstrated which is not reversible over a binary alphabet. However, up to now, it has been an open question whether or not all solvable multiple-unicast networks are reversible (i.e., over all alphabets). We prove here that not all solvable multiple-unicast networks are reversible.

Our approach exploits results from [4], [5], and the first part of the present correspondence. Specifically, we make use of the networks $\mathcal{N}_1$ (Fig. 5) and $\mathcal{N}_2$ (Fig. 6) to establish some useful lemmas.

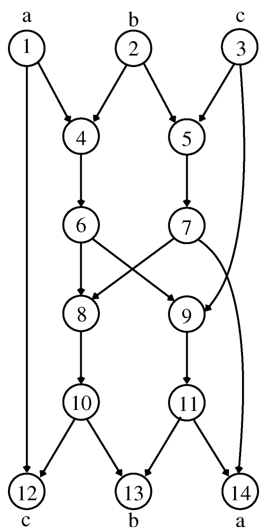

Fig. 5. The network $\mathcal{N}_1$. The network is solvable only for alphabets with power-of-two cardinalities. Also, the reverse of the network is itself.

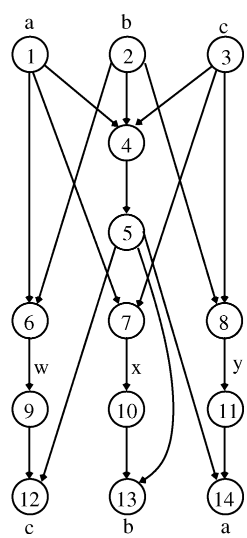

Fig. 6. The network $\mathcal{N}_2$. The network's solutions are characterized in terms of an Abelian group and certain fixed permutations.

More general versions of Lemma III.1 and Lemma III.2 were proved in [4] and [5], respectively.

Lemma III.1: The network $\mathcal{N}_1$ is solvable over an alphabet $\mathcal{A}$ if and only there exists a positive integer n, such that $|\mathcal{A}| = 2^n$.

Lemma III.2: For any solution to network $\mathcal{N}_2$ over an alphabet $\mathcal{A}$ and for any element $0 \in \mathcal{A}$, there exist permutations $\pi_1, \ldots, \pi_6$ of $\mathcal{A}$ and a mapping $+ : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ such that $(\mathcal{A}, +)$ is an Abelian group with identity element 0, $\pi_3(0) = 0$, and

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

We will refer to the network shown in Fig. 7 as the "Insufficiency" network. It was shown in [4] that the Insufficiency network has a nonlinear scalar solution over a 4-ary alphabet but has no vector linear solution over any finite field and any vector dimension.

One immediate consequence of Theorem II.1 is that linear coding is insufficient (even asymptotically) for multiple-unicast networks. This follows by adding gadgets to the Insufficiency network for messages that are demanded at two or more receiver nodes, in order to convert the Insufficiency network into a multiple-unicast network. The vector linear solvability properties of the resulting network are the same as the Insufficiency network, so the resulting multiple-unicast network remains solvable but not vector linearly solvable (as shown in [4]).

Let $\mathcal{N}_3$ be the network obtained by deleting nodes n_1, n_2, n_3 of the Insufficiency network, changing the messages at n_4, n_5, n_6 to a', b', and c', and merging nodes n_9 and n_{10}, as illustrated in Fig. 8.

Now create a new network $\mathcal{N}_4$, by modifying network $\mathcal{N}_3$, as shown in Fig. 9. To obtain $\mathcal{N}_4$, the six edges in $\mathcal{N}_3$ entering receiver n_{43} are replaced by four new nodes (i.e., n_{x1}, n_{x2}, n_{x3}, and a new n_{43}) and nine new edges.

Note that in the networks $\mathcal{N}_3$ and $\mathcal{N}_4$, the message c is demanded in each network by three receivers (n_{40}, n_{43}, and n_{46}) and all other messages are each demanded by exactly one receiver. We can create multiple-unicast networks from $\mathcal{N}_3$ and $\mathcal{N}_4$ by using the technique from the first part of this correspondence.

Specifically, create two multiple-unicast networks $\mathcal{N}_5$ (see Fig. 10) and $\mathcal{N}_6$ (see Fig. 11), by adding gadgets to networks $\mathcal{N}_3$ and $\mathcal{N}_4$, respectively, according to the construction given in the proof of Theorem II.1.

Lemma III.3: The network $\mathcal{N}_3$ is solvable.

Proof: It follows immediately from the solvability of the Insufficiency network [4]. □

Corollary III.4: The network $\mathcal{N}_5$ is solvable.

Proof: It follows immediately from Lemma III.3 and Theorem II.1. [4]. □

Lemma III.5: The network $\mathcal{N}_4$ is not solvable.

Proof: Suppose there is a solution to the right half of $\mathcal{N}_4$ on alphabet S. Fix an element 0 of the alphabet. By Lemma III.2, there exist permutations $\pi_1, \ldots, \pi_6, \hat{\pi}_1, \ldots, \hat{\pi}_6$, and Abelian groups operations $+$ and $+'$ on S, both with 0 as the identity element and with

$$\pi_3(0) = \hat{\pi}_3(0) = 0,$$

such that

$$\begin{aligned} e_{23,31} &= \pi_4(\pi_1(a) + \pi_2(b)) \\ e_{24,32} &= \pi_5(\pi_1(a) + \pi_3(c)) \\ e_{25,33} &= \pi_6(\pi_2(b) + \pi_3(c)) \\ e_{28,36} &= \hat{\pi}_4(\hat{\pi}_1(e) +' \hat{\pi}_2(d)) \\ e_{27,35} &= \hat{\pi}_5(\hat{\pi}_1(e) +' \hat{\pi}_3(c)) \\ e_{26,34} &= \hat{\pi}_6(\hat{\pi}_2(d) +' \hat{\pi}_3(c)). \end{aligned}$$

Therefore, in order for the demand at node n_{43} to be satisfied, there must exist functions $g, \hat{f}_1, \hat{f}_2, \hat{f}_3$ such that

$$\begin{aligned} c = g(&\hat{f}_1(\pi_5(\pi_1(a) + \pi_3(c)), \hat{\pi}_6(\hat{\pi}_2(d) +' \hat{\pi}_3(c))), \\ &\hat{f}_2(\pi_4(\pi_1(a) + \pi_2(b)), \hat{\pi}_4(\hat{\pi}_1(e) +' \hat{\pi}_2(d))), \\ &\hat{f}_3(\pi_6(\pi_2(b) + \pi_3(c)), \hat{\pi}_5(\hat{\pi}_1(e) +' \hat{\pi}_3(c)))). \end{aligned}$$

Define for all $x, y \in S$ the functions

$$\begin{aligned} f_1(x, y) &= \hat{f}_1(\pi_5(\pi_3(x)), \hat{\pi}_6(\hat{\pi}_3(y))) \\ f_2(x, y) &= \hat{f}_2(\pi_4(\pi_3(x)), \hat{\pi}_4(\hat{\pi}_3(y))) \\ f_3(x, y) &= \hat{f}_3(\pi_6(\pi_3(x)), \hat{\pi}_5(\hat{\pi}_3(y))) \end{aligned}$$

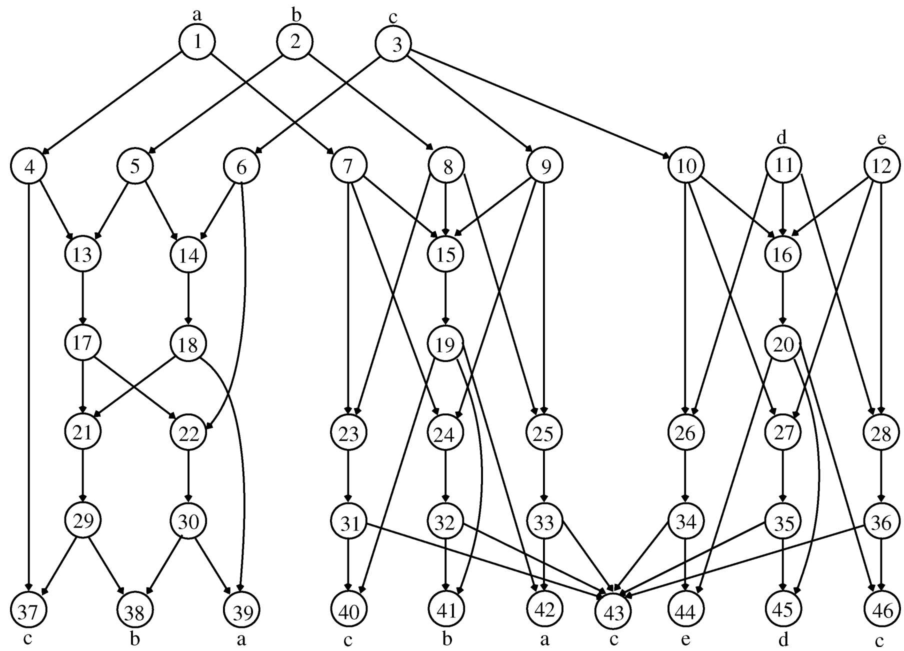

Fig. 7. The Insufficiency network.

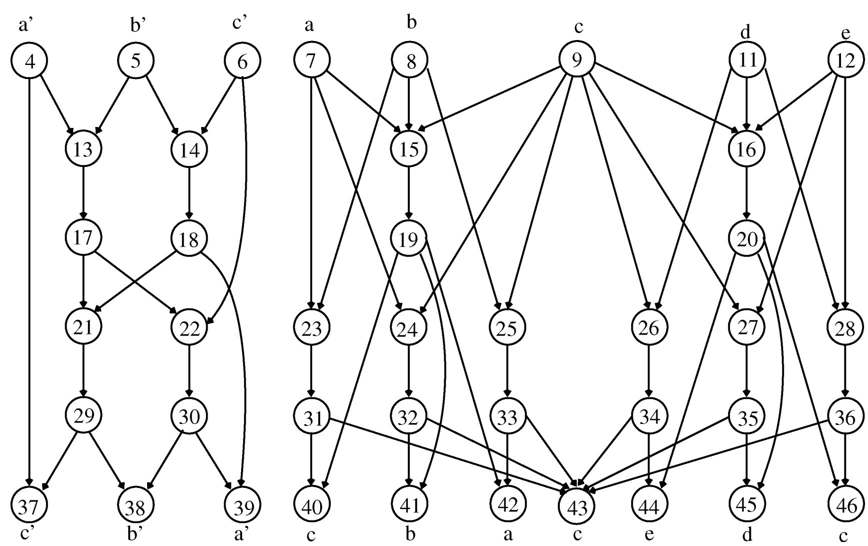

Fig. 8. The network $\mathcal{N}_3$, which is a modification of the Insufficiency network.

and the quantities

$$\begin{aligned} A &= \pi_3^{-1}(\pi_1(a)) \\ B &= \pi_3^{-1}(\pi_2(b)) \\ D &= \hat{\pi}_3^{-1}(\hat{\pi}_1(d)) \\ E &= \hat{\pi}_3^{-1}(\hat{\pi}_1(e)) \end{aligned}$$

and the binary operations

$$+_1 : S \times S \to S$$

and

$$+_2 : S \times S \to S$$

given by

$$\begin{aligned} x +_1 y &= \pi_3^{-1}(\pi_3(x) + \pi_3(y)) \\ x +_2 y &= \hat{\pi}_3^{-1}(\hat{\pi}_3(x) +' \hat{\pi}_3(y)). \end{aligned}$$

Then one can verify that $(\mathcal{A}, +_1)$ and $(\mathcal{A}, +_2)$ are Abelian groups, each with identity 0, and we can write

$$\begin{aligned} c = g(&f_1(A +_1 c, c +_2 D), \\ &f_2(A +_1 B, D +_2 E), \\ &f_3(B +_1 c, c +_2 E)). \end{aligned} \tag{1}$$

For any value of c, there exist values of a, b, d, e (and hence A, B, D, E) such that

$$A +_1 c = B +_1 c = c +_2 D = c +_2 E = 0.$$

Thus, for each c, there exist A, B, D, E such that

$$c = g(f_1(0,0), f_2(A +_1 B, D +_2 E), f_3(0,0));$$

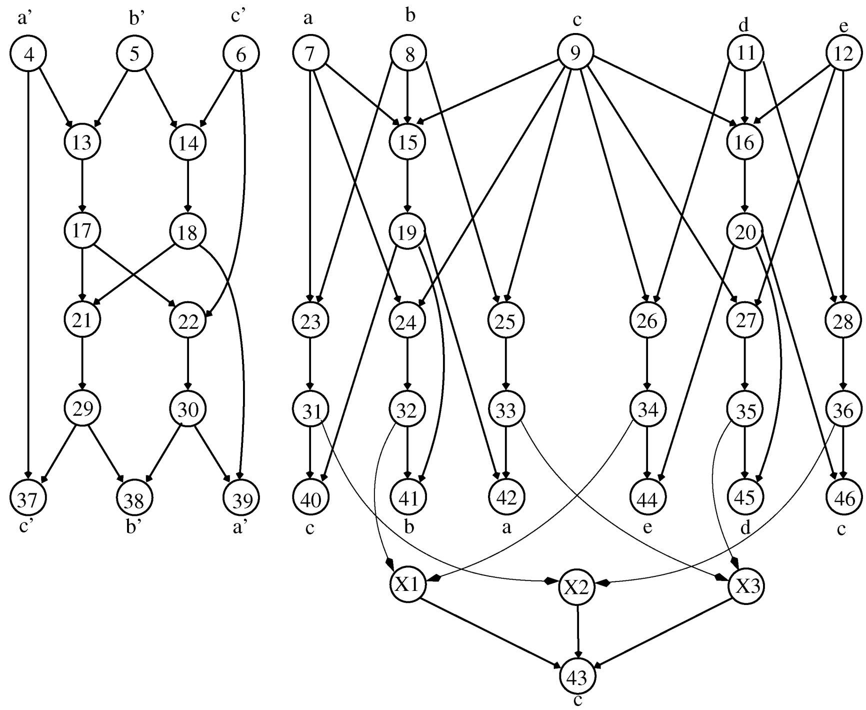

Fig. 9. The network $\mathcal{N}_4$, which is a modification of network $\mathcal{N}_3$.

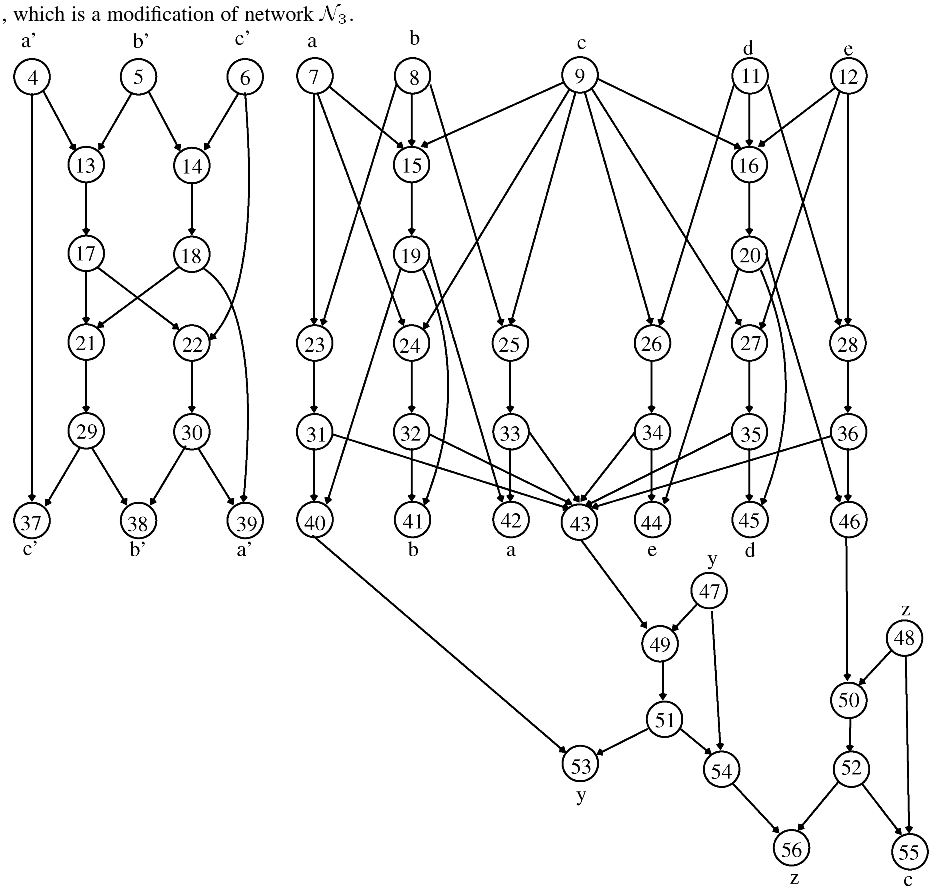

Fig. 10. The multiple-unicast network $\mathcal{N}_5$, which consists of $\mathcal{N}_3$ and two additional gadgets.

so since c assumes $|S|$ different values, $f_2(A +_1 B, D +_2 E)$ must also assume $|S|$ different values. Thus, the range of f_2 is all of S. A similar argument shows that the ranges of f_1 and f_3 are also all of S.

So, for any $x, y, c \in S$, we can find $r, s, t, u \in S$ such that

$$f_1(r, s) = x$$
$$f_2(t, u) = y$$

and then we can find A, B, D, E such that

$$A +_1 c = r$$
$$c +_2 D = s$$
$$A +_1 B = t$$
$$D +_2 E = u.$$

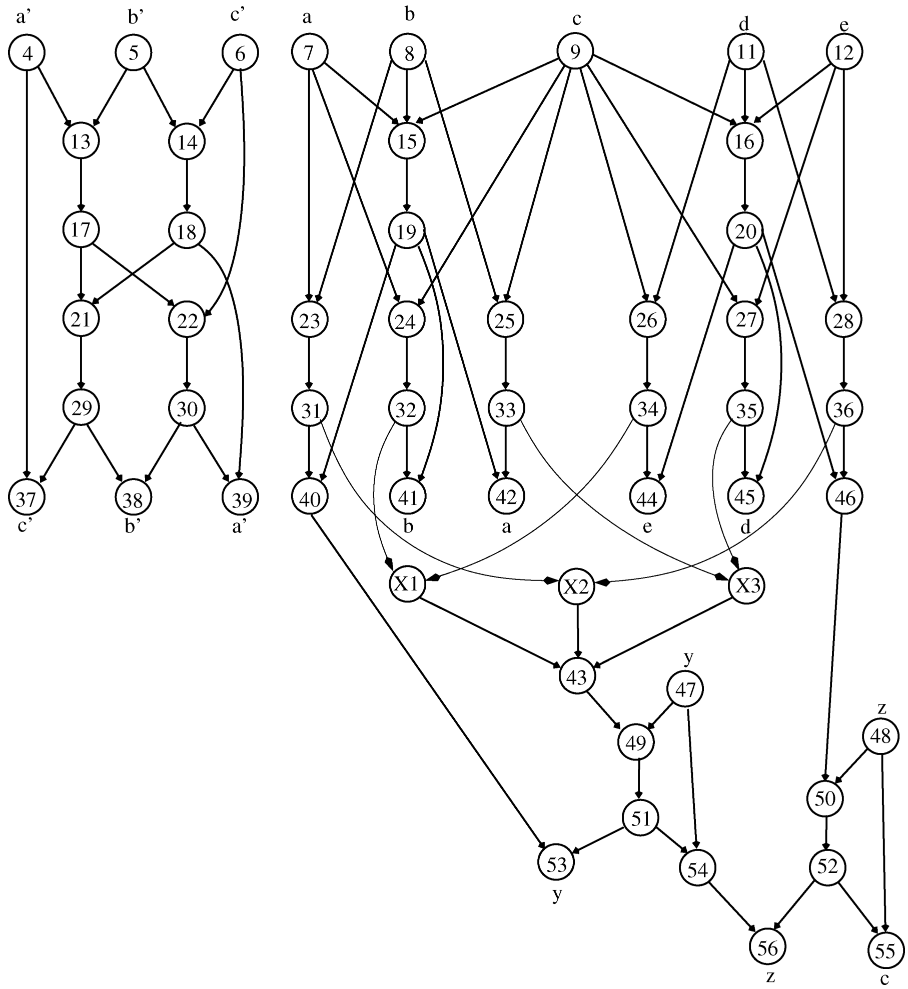

Fig. 11. The multiple-unicast network $\mathcal{N}_6$, which consists of $\mathcal{N}_4$ and two additional gadgets.

For such choices, if we let $z = f_3(B +_1 c, c +_2 E)$, then we get $c = g(x, y, z)$. Thus, g is surjective (and hence bijective) in its third argument. A similar argument shows that g is also bijective in its first and second arguments. Thus, g is a Latin cube.

Now if we set $A = c = D = 0$ in (1), we get

$$0 = g(f_1(0,0), f_2(B,E), f_3(B,E)).$$

Since g is Latin, it follows that

$$f_3(B,E) = f_3(B',E') \Longleftrightarrow f_2(B,E) = f_2(B',E').$$

This means that there exists a permutation π' of S such that $f_3 = \pi' \circ f_2$. Similarly, by setting $B = c = E = 0$, we can show that $f_1 = \pi'' \circ f_2$ for some permutation π'' of S. If we define

$$\hat{g}(x,y,z) = g(\pi''(x), y, \pi'(z))$$

then (1) can be rewritten as

$$\begin{aligned} c = \hat{g}(&f_2(A +_1 c, c +_2 D), \\ &f_2(A +_1 B, D +_2 E), \\ &f_2(B +_1 c, c +_2 E)). \end{aligned} \tag{2}$$

Note that since g is Latin and π' and π'' are permutations, the function $\hat{g}$ is also Latin.

Next, letting $c = 0$, we have

$$0 = \hat{g}(f_2(A,D), f_2(A +_1 B, D +_2 E), f_2(B,E)).$$

Since $\hat{g}$ is Latin, we get the following.

If $f_2(A,D) = f_2(A',D')$ and $f_2(B,E) = f_2(B',E')$, then $f_2(A +_1 B, D +_2 E) = f_2(A' +_1 B', D' +_2 E')$.

Therefore, we can define an operation

$$+_3 : S \times S \to S$$

as follows. For every $x, y \in S$, let $A, B, D, E \in S$ be such that

$$f_2(A,D) = x$$
$$f_2(B,E) = y$$

and let

$$x +_3 y = f_2(A +_1 B, D +_2 E).$$

Thus, f_2 is a homomorphism from $(S, +_1) \times (S, +_2)$ onto $(S, +_3)$. Since $(S, +_1) \times (S, +_2)$ is an Abelian group, one can verify that $(S, +_3)$ is an Abelian group with identity element $f_2(0,0)$.

We now show that the mapping

$$c \longmapsto f_2(c +_1 c, c +_2 c)$$

from S to S must be injective. Suppose there exist $c, c' \in S$ such that

$$f_2(c +_1 c, c +_2 c) = f_2(c' +_1 c', c' +_2 c').$$

From (2) we get

$$\begin{aligned} c &= \hat{g}(f_2(c' +_1 c, c +_2 c'), \\ &\quad f_2(c' +_1 c', c' +_2 c'), \\ &\quad f_2(c' +_1 c, c +_2 c')) \\ c' &= \hat{g}(f_2(c +_1 c', c' +_2 c), \\ &\quad f_2(c +_1 c, c +_2 c), \\ &\quad f_2(c +_1 c', c' +_2 c)). \end{aligned}$$

Since $+_1$ and $+_2$ are commutative, the arguments to $\hat{g}$ in the above two equations are the same, so $c = c'$.

So, the elements $f_2(c, c) +_3 f_2(c, c)$ of S are distinct for distinct c. Hence, the elements $f_2(c, c)$ are distinct, so the map $c \longmapsto f_2(c, c)$ is injective and therefore also surjective.

Now, suppose $(S, +_3)$ had an element x of order 2. Then there would exist a nonzero $c \in S$ such that

$$f_2(c, c) = x$$

which would imply that both $f_2(c, c) +_3 f_2(c, c)$ and $f_2(0, 0) +_3 f_2(0, 0)$ would be the identity element of $(S, +_3)$, which is a contradiction. Therefore, the group $(S, +_3)$ has no elements of order 2, so, by Cauchy's theorem, $|S|$ must be odd.

Thus, we have shown that the right half of $\mathcal{N}_4$ is solvable only for odd alphabet sizes. Since the left half of $\mathcal{N}_4$ is solvable only for power-of-two alphabet sizes (by Lemma III.1), the combined network $\mathcal{N}_4$ is not solvable. □

Corollary III.6: The network $\mathcal{N}_6$ is not solvable.

Proof: It follows from Lemma III.5 and Theorem II.1. □

Lemma III.7: Network $\mathcal{N}_3$ is reversible if and only if network $\mathcal{N}_4$ is reversible.

Proof: Given a solution to the reverse of $\mathcal{N}_3$, one can get a solution to the reverse of $\mathcal{N}_4$ by letting the new $e_{X2,31}$ be the same as the old $e_{43,31}$, the new $e_{X1,32}$ be the same as the old $e_{43,32}$, and so on (and letting the new $e_{43,X1}$, $e_{43,X2}$, and $e_{43,X3}$ be c). And given a solution to the reverse of $\mathcal{N}_4$, one can get a solution to the reverse of $\mathcal{N}_3$ by letting the new $e_{43,31}$ be the same as the old $e_{X2,31}$, the new $e_{43,32}$ be the same as the old $e_{X1,32}$, and so on (these new values are computable from c since they are computable from the old $e_{43,X1}$, $e_{43,X2}$, and $e_{43,X3}$, which are all computable from c). □

Lemma III.8: Network $\mathcal{N}_5$ is reversible if and only if network $\mathcal{N}_6$ is reversible.

Proof: Same as the proof of Lemma III.7 (using $e_{49,43}$ instead of c). □

Corollary III.9: There exists a solvable multiple-unicast network that is not reversible.

Proof: From Lemma III.8, either network $\mathcal{N}_5$ is solvable and its reverse is not solvable, or network $\mathcal{N}_6$ is not solvable and its reverse is solvable. □

Corollary III.9 gives an existence proof of the nonreversibility of solvable multiple-unicast networks. The proof is semi-constructive since it narrows down a specific network exhibiting the nonreversibility property to one of two closely related networks. In order to give a definitive network witnessing nonreversibility, we next show which of the two networks is solvable with a nonsolvable reverse.

Theorem III.10: The reverse of the multiple-unicast network $\mathcal{N}_6$ is solvable but not reversible.

Proof: We give an explicit nonlinear solution to the reverse of network $\mathcal{N}_6$ as shown in Fig. 12. The nontrivial edge functions are specified; all other edge functions are simply copied from their inputs. The solution is over a 4-ary alphabet $\mathcal{A} = \{0, 1, 2, 3\}$. The operations $+$ and $-$ are modulo-4 addition and subtraction (i.e., in the ring $\mathbf{Z}_4$) and $\oplus$ is bitwise modulo-2 addition (i.e., in the ring $\mathbf{Z}_2 \times \mathbf{Z}_2$) viewing the alphabet as $\mathcal{A} = \{00, 01, 10, 11\}$. The mapping $t : \mathcal{A} \to \mathcal{A}$ switches the bits in the binary representation of its argument. The quantities $H, L \in \{0, 1\}$ are, respectively, the high and low bits in the binary representation of $y \oplus z$. The messages can be recovered as follows:

$$\begin{aligned} n_4 &: a' = (a' \oplus c') \oplus c' \\ n_5 &: b' = (a' \oplus b' \oplus c') \oplus (a' \oplus c') \\ n_6 &: c' = (a' \oplus b' \oplus c') \oplus (a' \oplus b') \\ n_7 &: a = (a + b + t(y)) - (t(y) + L) - (b - L) \\ n_8 &: b = (a + b + t(y)) - (t(y) + L) - (a - L) \\ n_{11} &: d = (e + d + (c \oplus z)) - (e - H) - ((c \oplus z) + H) \\ n_{12} &: e = (e + d + (c \oplus z)) - (d - H) - ((c \oplus z) + H) \\ n_{47} &: y = (y \oplus z) \oplus z \\ n_{48} &: z = (c \oplus z) \oplus c. \end{aligned}$$

The message c can be recovered at node n_9 since

$$\begin{aligned} & t((a + b + t(y)) - (a - L) - (b - L)) \\ & \quad \oplus ((e + d + (c \oplus z) - (e - H) - (d - H)) \\ &= t(t(y) + 2L) \oplus ((c \oplus z) + 2H) \\ &= t(t(y) \oplus 2L) \oplus ((c \oplus z) \oplus 2H) \\ & \quad [\text{since } r + s = r \oplus s \text{ whenever } s \text{ is even}] \\ &= y \oplus t(2L) \oplus c \oplus z \oplus 2H \ [\text{since } t \text{ commutes with } \oplus] \\ &= c \oplus y \oplus z \oplus (t(2L) \oplus 2H) \\ &= c \oplus y \oplus z \oplus (y \oplus z) \\ &= c. \end{aligned}$$

□

While the reverse of $\mathcal{N}_6$ is an example of a network that is solvable but not reversible, the fact that it is a union of two disjoint networks might suggest that two different alphabets could be used when describing the network's solvability (i.e., one alphabet for each of the two disjoint subnetworks). We will show that a small modification of $\mathcal{N}_6$ can yield a connected network which is solvable but not reversible.

Theorem III.11: Suppose two disjoint multiple-unicast networks $\mathcal{N}'$ and $\mathcal{N}''$ are connected with a gadget to form a new network $\mathcal{N}$, as shown in Fig. 13. Then $\mathcal{N}$ is solvable (respectively, reversible) over alphabet $\mathcal{A}$ if and only if both $\mathcal{N}'$ and $\mathcal{N}''$ are solvable (respectively, reversible) over $\mathcal{A}$.

Proof: Suppose $\mathcal{N}'$ and $\mathcal{N}''$ are each solvable over an alphabet of size m, which we assume, without loss of generality, is $\{0, 1, \ldots, m-1\}$. We describe a solution for $\mathcal{N}$. Let nodes n_5, n_6, and n_8 simply copy their inputs to their outputs. Let edge $e_{7,8}$ add the two input edges of node n_7 modulo m (i.e., $e_{7,8}$ carries the symbol $p + q \pmod{m}$), so that node n_1 can deduce message p by subtracting $e_{5,1}$ from $e_{8,1}$ modulo m, and node n_3 can deduce message q by subtracting $e_{6,3}$ from $e_{8,3}$ modulo m. Let the remaining edges of $\mathcal{N}$ perform the same coding functions as in the solutions for $\mathcal{N}'$ and $\mathcal{N}''$. This gives a solution to $\mathcal{N}$.

Now, suppose $\mathcal{N}$ has a solution over some alphabet $\mathcal{A}$. Let $e_1, \ldots, e_j$ be the out-edges of n_1. In the solution for $\mathcal{N}$, for all i we can write

$$e_i = f_i(e_{5,1}, e_{8,1}, x)$$

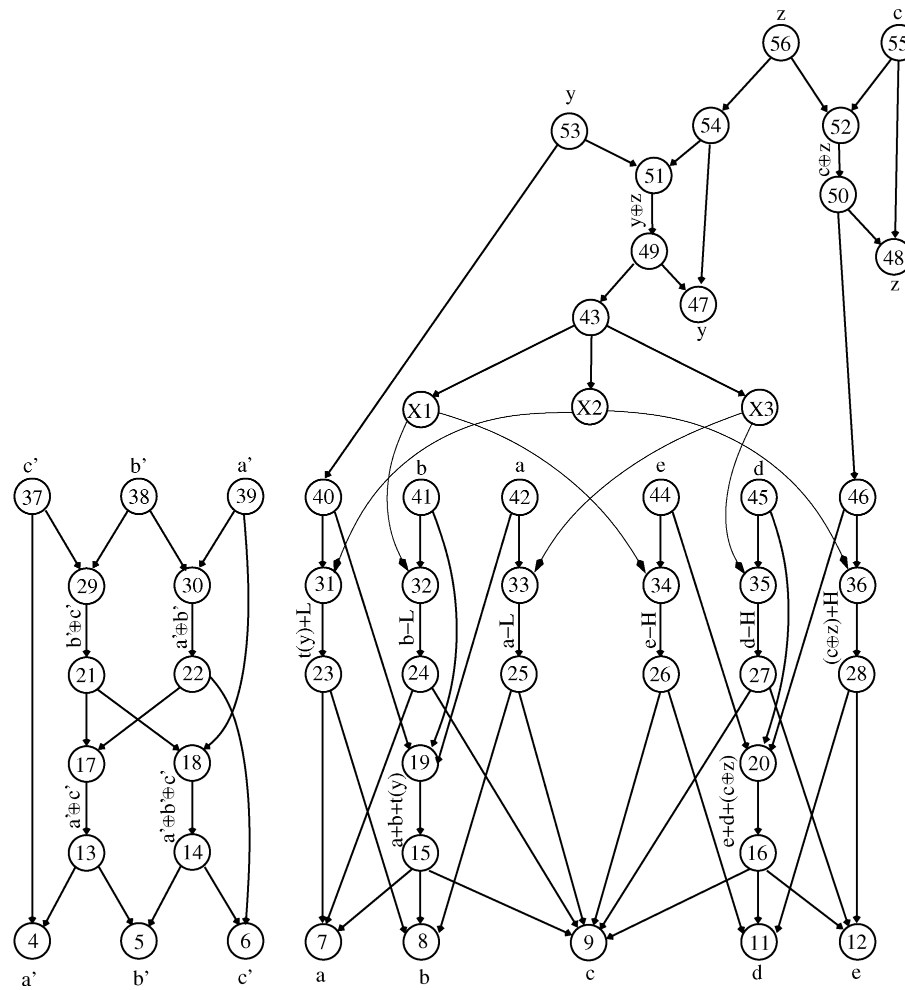

Fig. 12. A nonlinear solution over a 4-ary alphabet is shown for the reverse of network $\mathcal{N}_6$. The source messages are shown above the source nodes and the demand messages are shown below the receiver nodes.

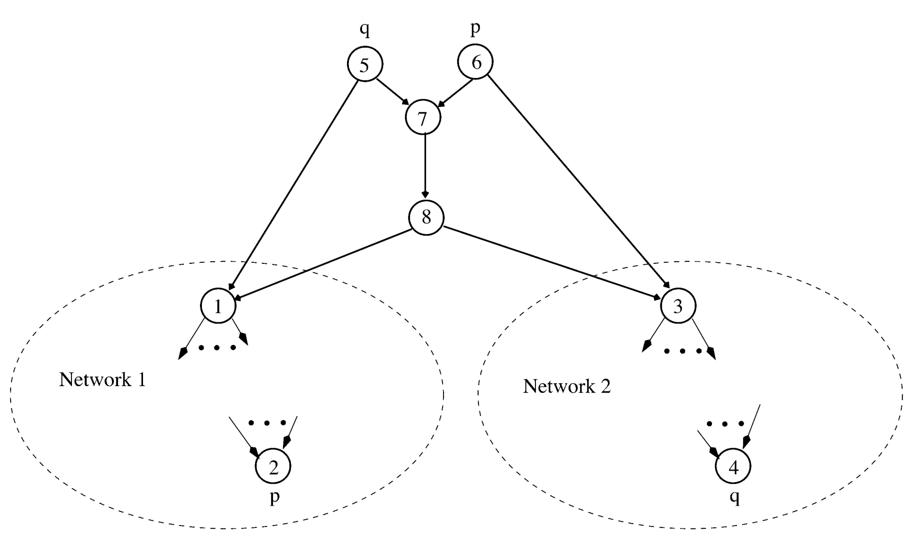

Fig. 13. Two arbitrary disjoint multiple-unicast networks $\mathcal{N}'$ and $\mathcal{N}''$ (depicted in the ovals) connected with a gadget to form a new multiple-unicast network $\mathcal{N}$. Network $\mathcal{N}'$ has a source node n_1 with message p which is demanded by node n_2, and network $\mathcal{N}''$ has a source node n_3 with message q which is demanded by node n_4. Network $\mathcal{N}$ is connected by a gadget, consisting of nodes n_5, n_6, n_7, n_8 and edges $e_{5,1}$, $e_{5,7}$, $e_{6,3}$, $e_{6,7}$, $e_{7,8}$, $e_{8,1}$, $e_{8,3}$. In $\mathcal{N}$, the messages p and q originate at nodes n_6 and n_5, respectively.

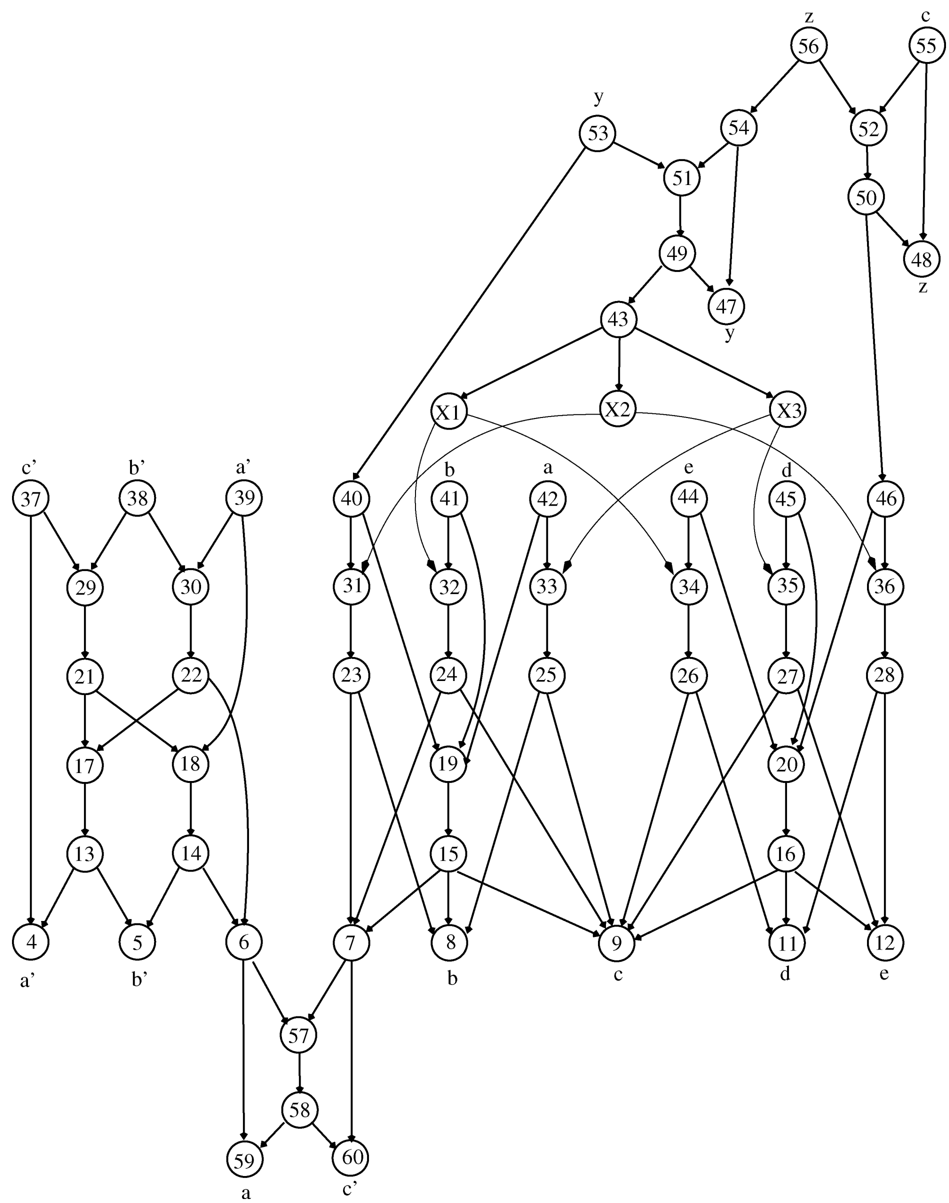

Fig. 14. A solvable connected multiple-unicast network $\mathcal{N}_7$ that is not reversible.

where x is the collection of all input source messages to n_1 and in-edges of n_1 other than $e_{5,1}$ and $e_{8,1}$. Since $e_{5,1}$ depends only on q and since $e_{8,1}$ depends only on p and q, we can write

$$e_i = g_i(p, q, x).$$

For each i, let us define a function

$$f'_i(p, x) = g_i(p, 0, x).$$

Now, to construct a solution for $\mathcal{N}'$, copy from the solution for $\mathcal{N}$ all edge functions from nodes other than n_1 and all demand functions other than at n_1. For the ith out-edge of n_1 in $\mathcal{N}'$, define the coding function

$$e_i = f'_i(p, x).$$

This gives a solution to $\mathcal{N}'$ on $\mathcal{A}$ and a similar construction gives a solution to $\mathcal{N}''$ on $\mathcal{A}$.

Suppose the reverses of $\mathcal{N}'$ and $\mathcal{N}''$ are solvable over an alphabet of size m. Then in a similar manner as before, a solution to the reverse of $\mathcal{N}$ is obtained by having nodes n_1, n_3, and n_7 simply copy their inputs to their outputs and having edge $e_{8,7}$ add the two input edges of node n_8 modulo m.

Likewise, if the reverse of $\mathcal{N}$ is solvable, then in order for p to be obtained at n_6 and q to be obtained at n_5, message p must be deducible at n_1 and message q must be deducible at n_3, thus giving solutions to networks $\mathcal{N}'$ and $\mathcal{N}''$. □

Theorem III.11 provides a mechanism for connecting two disjoint networks into a single network which is solvable over a particular alphabet if and only if both of the disjoint pieces were solvable over that alphabet. In particular, if the two disjoint networks were both solvable, but never over a common alphabet, then the new connected network would not be solvable.

Corollary III.12: The connected multiple-unicast network $\mathcal{N}_7$ is solvable but not reversible.

Proof: The network $\mathcal{N}_7$ in Fig. 14 is the reverse of a network obtained by adding a gadget to the disjoint union network $\mathcal{N}_6$ in Fig. 11, connecting nodes n_6 and n_7. By Theorem III.11, the reverse of $\mathcal{N}_7$ is solvable over alphabet $\mathcal{A}$ if and only if the two disjoint pieces of $\mathcal{N}_6$ are solvable over $\mathcal{A}$. But, by Theorem III.10, network $\mathcal{N}_6$ is not solvable over any alphabet, so neither is the reverse of $\mathcal{N}_7$. Hence, $\mathcal{N}_7$ is not reversible. Also, by Theorem III.10, the reverse of $\mathcal{N}_6$ is solvable, and thus $\mathcal{N}_7$ is solvable, by Theorem III.11. □

IV. Conclusion

We have demonstrated a multiple-unicast network which is solvable, but whose reverse is not solvable. However, we note that the network and its reverse each have coding capacity equal to one. In particular, this implies that the reverse network is asymptotically solvable. An interesting open question is whether a multiple-unicast network could have coding capacity different from the coding capacity of its reverse. Another interesting open problem is to characterize the class of all non-reversible solvable multicast networks.

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 (SODA 06)*, Miami, FL, Jan. 2006.

[2] 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.

[3] R. Dougherty, C. Freiling, and K. Zeger, "Linearity and solvability in multicast networks," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2243–2256, Oct. 2004.

[4] ——, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[5] ——, "Unachievability of network coding capacity," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2365–2372, Jun. 2006, Joint issue of *IEEE Trans. Inf. Theory and IEEE/ACM Trans. Netw*.

[6] M. Feder, D. Ron, and A. Tavory, "Bounds on linear codes for network multicast," in *Proc. Electronic Colloquium on Computational Complexity (ECCC)*, 2003, pp. 1–9.

[7] N. J. A. Harvey, R. D. Kleinberg, and A. Rasala Lehman, "Comparing Network Coding with Multicommodity Flow for the k-Pairs Communication Problem," MIT Lab. Computer Science, 2004, Tech. Rep. 964.

[8] ——, "On the capacity of information networks," *IEEE Trans. Inf. Theory* , vol. 52, no. 6, pp. 2345–2364, Jun. 2006, Joint issue of *IEEE Trans. Inf. Theory and IEEE/ACM Trans. Netw*.

[9] 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.

[10] R. Koetter, M. Effros, T. Ho, and M. Médard, "Network codes as codes on graphs," in *Proc. 38th Annu. Conf. Information Sciences and Systems (CISS)*, Princeton, NJ, Mar. 2004.

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

[12] 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.

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

[14] A. Rasala Lehman, "Network Coding," Ph.D. dissertation, Dep. Electrical Engineering and Computer Science, MIT, Cambridge, MA, 2005.

[15] A. Rasala 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.

[16] S. Riis, "Linear versus non-linear Boolean functions in network flow," in *Proc. 38th Annu. Conf. Information Sciences and Systems (CISS)*, Princeton, NJ, Mar. 2004.

[17] ——, "Reversible and Irreversible Information Networks" [Online]. Available: http://www.dcs.qmul.ac.uk/~smriis, Preprint.

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

Min-Cost Selfish Multicast With Network Coding

Sandeep Bhadra, *Student Member, IEEE*, Sanjay Shakkottai, *Member, IEEE*, and Piyush Gupta, *Member, IEEE*

***Abstract*—The single-source min-cost multicast problem, which can be framed as a convex optimization problem with the use of network codes and convex increasing edge costs is considered. A decentralized approach to this problem is presented by Lun, Ratnakar *et al.* for the case where all users cooperate to reach the global minimum. Further, the cost for the scenario where each of the multicast receivers greedily routes its flows is analyzed and the existence of a Nash equilibrium is proved. An allocation rule by which edge cost at each edge is allocated to flows through that edge is presented. We prove that under our pricing rule, the flow cost at user equilibrium is the same as the min-cost. This leads to the construction of a selfish flow-steering algorithm for each receiver, which is also globally optimal. Further, the algorithm is extended for completely distributed flow adaptation at nodes in the network to achieve globally minimal cost in steady state. Analogous results are also presented for the case of multiple multicast sessions.**

***Index Terms*—Convex optimization, game theory, minimum cost multicast, Nash equilibrium, network coding.**

I. Introduction

The single-source multicast problem for network coding has received much attention in recent years due to the tractability of designing optimal linear network codes for this case. Ahlswede, *et al.* in [2] prove that for networks where the min-cut max-flow rate cannot be achieved by simple forwarding of packets, coding incoming packets at intermediate routers (network-coding) can help achieve the max-flow min-cut rate for such networks. Further, Ho *et al.* [3], [4] suggest the use of random linear codes (RLCs) that can achieve the above linear network code rate asymptotically in the size of the symbol alphabet used for encoding/decoding. Since the intermediate routers can code randomly independent of other routers in the network, RLCs offer the means for decentralized design of network codes and form the basis for practical network coding schemes [5].

The problem of finding the minimum-cost multicast tree for networks has been studied extensively. For a general directed graph with a cost function at each edge, a specified root (source) and a subset of the nodes (receivers), the problem of finding a minimum-cost arborescence rooted at the source and spanning all the receivers is called the Directed Steiner Tree (DST) problem. Approximation algorithms for the DST, which is known to be NP-hard, has received considerable attention in recent years. Charikar *et al.* [6] present a non-trivial

Manuscript received December 13, 2005; revised May 5, 2006. The work of S. Bhadra and S. Shakkottai was supported by the National Science Foundation under Grants ACI-0305644, CNS-0325788 and CNS-0347400. The work of P. Gupta was supported in part by the National Science Foundation under Grants CCR-0325673 and CNS-0519535. The material in this correspondence was presented in part as an invited talk at the Workshop on Mathematical Modeling and Analysis of Computer Networks, Waterloo, ON, Canada, May 2005 and at the Second Workshop on Network Coding, Theory and Applications (NETCOD), Boston, MA, April 2006.

S. Bhadra and S. Shakkottai are with the Wireless Networking and Communications Group, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: bhadra@ece.utexas.edu; shakkott@ece.utexas.edu).

P. Gupta is with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: pgupta@research.bell-labs.com).

Communicated by E. Modiano, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2006.883636