Linearity and Solvability in Multicast Networks

Randall Dougherty, Chris Freiling, and Kenneth Zeger

***Abstract*— It is known that for every solvable multicast network, there exists a large enough finite field alphabet such that a linear solution exists. We prove: (i) every binary solvable multicast network with at most two messages has a binary linear solution; (ii) for more than two messages, not every binary solvable multicast network has a binary linear solution; (iii) a multicast network that has a solution for a given alphabet might not have a solution for all larger alphabets.**

I. Introduction

A multicast network is a directed acyclic multigraph containing a single source node and a collection of destination nodes. The source node emits a message from a fixed alphabet on each of its out-edges and each destination node tries to recover all of the messages.

Each node in the graph receives an alphabet symbol on each of its in-edges and transmits a symbol on each of its out-edges. Each transmitted symbol is computed for an out-edge by a fixed function of the symbols received by a node. A multicast network is solvable with respect to the alphabet if the edge-functions can be assigned in such a way that the source messages can always be recovered at each destination node. The set of edge-functions and destination node functions constitutes a network code. A network code is called a solution if it allows each destination node to recover all the messages.

Ahlswede, Cai, Li, and Yeung [1] introduce the concept of network coding and give a condition based on the max-flow min-cut theorem for the solvability of multicast networks. Li, Yeung, and Cai [3] study network codes that are linear when the message alphabet is the underlying set of a finite field. They show that any solvable multicast network has a linear solution provided the finite field alphabet is of large enough cardinality.

Although linear solutions are guaranteed by [3] for large enough finite field alphabets, their results do not guarantee the existence of a linear solution for a solvable multicast network if the alphabet size is fixed, nor do they consider alphabets whose cardinalities are not integer powers of primes. In recent independent work, Riis (in collaboration with Ahlswede) [5] shows there there exists a multicast network with five messages that is solvable over the binary field but which has no linear solution over the binary field. Their network is based on the nonlinear Nordstrom-Robinson $(12, 5, 5)$ error-correcting code. It has been unknown whether binary solvable multicast networks with three or four messages necessarily were linearly solvable.

We address the fixed alphabet issue for binary alphabets, namely when each edge in a multicast network carries a single bit. Proofs of all lemmas are omitted.

It can be useful to impose an algebraic structure on the alphabet $\mathcal{A}$, such as a ring or a field. In such a case, a code is *linear* (respectively, *affine*) with respect to $\mathcal{A}$, if each edge function is linear (respectively, affine) over $\mathcal{A}$. If $\mathcal{A} = \mathbf{Z}_2$, then a solution is called a *binary solution*, and the network said to be *binary solvable*.

II. Linear codes suffice for two messages

A function $f : \mathcal{A}^k \to \mathcal{A}$ is *homogeneous* if $f(\{0\}^k) = 0$. A solution is *homogeneous* if the label of every edge is homogeneous.

Lemma II.1. *For a given finite field alphabet and multicast network, if there exists a solution, then there exists a homogeneous solution. Furthermore, if there exists an affine solution, there there exists a linear solution.*

Theorem II.2. *Every binary solvable network with at most two messages has a binary linear solution.*

Proof. A multicast network with exactly one message is solvable if and only if there is a directed path from the source node to every destination node, in which case a linear solution simply labels every edge in all such paths with the single message.

By Lemma II.1, one may assume without loss of generality that a solution to a solvable multicast network is homogeneous. So assume a binary solvable multicast network with exactly two messages x and y and a homogeneous solution.

It suffices to prove that every binary solvable multicast network that has at most two messages and only 2-input edge-functions has a linear solution. This is true because any k-input boolean function is logically equivalent to some circuit consisting of only 2-input boolean functions, the linearity of which implies the linearity of the multi-input configuration. Thus we assume all edge-functions have exactly two inputs.

The bit passed along each edge of the network is a function of the two messages x, y. The collection of all possible functions is given by

$$\mathcal{P} = \{0, x, y, x + y, xy, xy + x, xy + y, xy + x + y\}.$$

This work was supported by the Institute for Defense Analyses, the National Science Foundation, and Ericsson.

R. Dougherty is with the Center for Communications Research, 4320 Westerra Court, San Diego, CA 92121-1969 (rdough@ccrwest.org).

C. Freiling is with the Department of Mathematics, California State University, San Bernadino, 5500 University Parkway, San Bernardino, CA 92407-2397 (cfreilin@csusb.edu).

K. Zeger is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 (zeger@ucsd.edu).

For any $f \in \mathcal{P}$, let $\mathrm{L}(f) \in \mathcal{P}$ be the linear function defined by

$$\mathrm{L}(f) = \begin{cases} f & \text{if } f \text{ is linear} \\ xy + f & \text{if } f \text{ is not linear.} \end{cases}$$

The function $\mathrm{L}(f)$ "linearizes" f by deleting the xy term if it appeared in the polynomial. If we can replace each edge function f with $\mathrm{L}(f)$ we will be done, since the messages themselves are linear functions, with $\mathrm{L}(x) = x$ and $\mathrm{L}(y) = y$. It only remains to show that this can be accomplished using linear functions at each node. That is, we need to show that if f, u, v are homogeneous boolean functions, then $\mathrm{L}(f(u, v))$ is a linear function of $\mathrm{L}(u)$ and $\mathrm{L}(v)$, a fact that can be readily checked. ■

III. Linear codes do not suffice for more than two messages

The circuit shown in Figure 1 will be used as a building block in part of Theorem III.3.

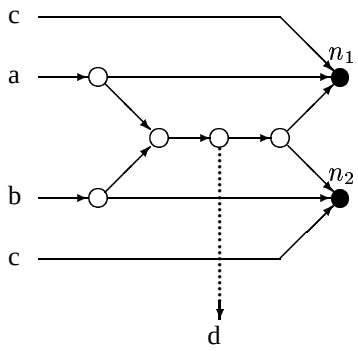


Fig. 1. Diagram of the circuit $\mathcal{G}_{a,b,c}$ where a, b, c, d are labels of the indicated edges. The nodes n_1 and n_2 are destination nodes, and the other nodes are interior nodes. The quantity d is called the *output*.

Lemma III.1. *If the circuit $\mathcal{G}_{a,b,c}$ appears in a multicast network with three messages, then for any homogeneous binary solution the edge label d must be $a + b$.*

The *majority vote* function is a 3-input boolean function M defined for all $a, b, c \in Z_2$ by

$$\mathrm{M}(a, b, c) = ab + bc + ca.$$

A set of functions $f_1, \ldots, f_j : \mathcal{A}^m \to \mathcal{A}$ is *complete* if every input can be recovered from the set of outputs. A 3-input boolean function f is *majority-linear* if there exists a complete set of 3-input linear boolean functions a, b, c such that $f = \mathrm{M}(a, b, c)$.

Lemma III.2. *If a, b, c, and d are linear functions such that $\{a, b, c\}$ and $\{a, b, d\}$ are complete, then the following sets are also complete:*

(a) $\{a + b, a + c, \mathrm{M}(a, b, c)\}$
(b) $\{a, b, \mathrm{M}(a, b, c), \mathrm{M}(a, d, a + b + d)\}$
(c) $\{b, \mathrm{M}(a, b, c), \mathrm{M}(a, a + b, c), \mathrm{M}(c, a + b, a + c)\}$
(d) $\{b, \mathrm{M}(a, b, c), \mathrm{M}(a, b + c, c), \mathrm{M}(b, a + b, a + c)\}$
(e) $\{b, \mathrm{M}(b, c, a + b + c), \mathrm{M}(c, a + b, b + c),$
$\mathrm{M}(c, a + b, a + c)\}$.

Theorem III.3. *For every $m \geq 3$, there exists a binary solvable multicast network with m messages that does not have a binary linear solution.*

Proof. We prove the result for $m = 3$; it is straightforward to extend it to all $m \geq 3$ by adding to our network $m - 3$ nodes which each receive one out-edge from the source, and then copy their inputs to all destination nodes. Throughout this proof, edge labels will be called "linear" (respectively, "majority-linear") to mean that they are linear (respectively, majority-linear) functions of the source messages x,y,z. Define a multicast network $\mathcal{N}_1$ with three messages x, y, z and a code over the binary alphabet to consist of the following components:

(i) A source node s with 3 out-edges labeled x, y, z.
(ii) Circuits $\mathcal{G}_{x,y,z}$, $\mathcal{G}_{y,z,x}$, $\mathcal{G}_{z,x,y}$ with output edge-functions x+y, y+z, x+z, respectively.
(iii) Circuit $\mathcal{G}_{x+y,z,y}$ with output x+y+z.
(iv) A 3-input, 1-output interior node $n^{(1)}(A, B, C)$ for each complete set of linear inputs $\{A, B, C\}$, and with output edge-function $\mathrm{M}(A, B, C)$.
(v) A 3-input destination node $n^{(2)}(A, B, M_1)$ for each complete set of inputs $\{A, B, M_1\}$, where A and B are linear and M_1 is majority-linear.
(vi) A 4-input destination node $n^{(3)}(A, B, M_1, M_2)$ for each complete set of inputs $\{A, B, M_1, M_2\}$, where A and B are linear and M_1 and M_2 are majority-linear.
(vii) A 4-input destination node $n^{(4)}(A, B, C)$ for each linear complete set $\{A, B, C\}$, where $\{B, \mathrm{M}(A, B, C), \mathrm{M}(A, A$+$B, C),$ $\mathrm{M}(C, A$+B, A+$C)\}$ is a complete set of inputs.
(viii) A 4-input destination node $n^{(5)}(A, B, C)$ for each linear complete set $\{A, B, C\}$, where is a complete set of inputs.
(ix) A 4-input destination node $n^{(6)}(A, B, C)$ for each linear complete set $\{A, B, C\}$, where $\{B, \mathrm{M}(B, C, A$+B+$C), \mathrm{M}(A$+B, B+$C, C),$ $\mathrm{M}(C, A$+B, A+$C)\}$ is a complete set of inputs.
(x) A 1-input, multiple-output interior node $n^{(7)}(A)$ for each A that is linear or majority-linear. Each edge-function on the out-edges of $n^{(7)}(A)$ is the identity function. The node's in-edge comes from the component which generates A; the out-edges lead to all other components which use A as input.

The code described in the definition of $\mathcal{N}_1$ is a solution. This follows from the fact that every destination node is defined to have a complete set of inputs.

We next show that the multicast network $\mathcal{N}_1$ does not have a binary linear solution. Assume to the contrary, that there exists a linear solution. We may assume without loss of generality that the source's three out-edges are labeled with the messages x, y, z (otherwise the out-edges could be equivalently relabeled).

Then each edge of the digraph which is labeled by some (possibly nonlinear) function gets replaced by some linear function. Also, no edge already labeled with a linear function can be replaced by anything other than

itself, since these edges only come from the output of a $\mathcal{G}$-circuit. So only the majority-linear functions can potentially get replaced by linear functions in $\mathcal{N}_1$.

Note that if A, B, C, D are linear functions and $\mathrm{M}(A,B,C)$ is replaced by D somewhere, then $\mathrm{M}(A,B,C)$ must be replaced by D everywhere in $\mathcal{N}_1$. We use the notation $\mathrm{M}(A,B,C) \rightsquigarrow D$ to mean that the nonlinear edge label $\mathrm{M}(A,B,C)$ occurs in the given solution for $\mathcal{N}_1$ and is replaced everywhere it occurs by the linear label D in the linearly labeled network. If S is a set of linear functions then the notation $\mathrm{M}(A,B,C) \rightsquigarrow S$ will mean that there exists $D \in S$ such that $\mathrm{M}(A,B,C) \rightsquigarrow D$.

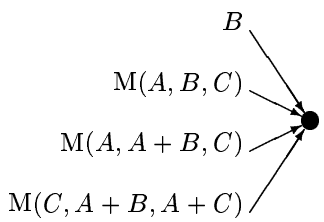
Lemma III.4. *In the multicast network $\mathcal{N}_1$, the following replacement rules hold:*

(a) For every complete set of linear functions $\{A,B,C\}$, we cannot have $\mathrm{M}(A,B,C) \rightsquigarrow 0$.
(b) $\mathrm{M}(A,B,C) \rightsquigarrow \{A,B,C,A+B+C\}$, for all complete sets of linear functions $\{A,B,C\}$.
(c) If $\mathrm{M}(A,B,C) \rightsquigarrow A$, then $\mathrm{M}(B,C,A+B+C) \rightsquigarrow A+B+C$.
(d) If $\mathrm{M}(A,B,C) \rightsquigarrow A+B+C$, then $\mathrm{M}(B,C,A+B+C) \rightsquigarrow A$.
(e) If $\mathrm{M}(A,B,C) \rightsquigarrow A$, then $\mathrm{M}(A,B,A+B+C) \rightsquigarrow B$.
(f) If $\mathrm{M}(A,B,C) \rightsquigarrow A$, then $\mathrm{M}(A,B,D) \rightsquigarrow \{A,B\}$ for all linear D such that $\{A,B,D\}$ is complete.
(g) If $\mathrm{M}(A,B,C) \rightsquigarrow \{C,A+B+C\}$, then $\mathrm{M}(A,B,D) \rightsquigarrow \{D,A+B+D\}$ for all linear D such that $\{A,B,D\}$ is complete.
(h) If $\mathrm{M}(A,B,C) \rightsquigarrow \{A,A+B+C\}$, then $\mathrm{M}(A,D,B+C+D) \rightsquigarrow \{A,A+B+C\}$ for all linear D such that $\{A,D,B+C+D\}$ is complete.
(i) If $\mathrm{M}(A,B,C) \rightsquigarrow \{B,C\}$, then $\mathrm{M}(A,D,B+C+D) \rightsquigarrow \{D,B+C+D\}$ for all linear D such that $\{A,D,B+C+D\}$ is complete.

By Lemma III.4b, either there exists a complete set of linear functions $\{A,B,C\}$ such that $\mathrm{M}(A,B,C) \rightsquigarrow A$, or else $\mathrm{M}(A,B,C) \rightsquigarrow A+B+C$ for every complete set of linear functions $\{A,B,C\}$.

First, let us assume (in Cases 1 and 2, below) that $\{A,B,C\}$ is a complete set of linear functions such that $\mathrm{M}(A,B,C) \rightsquigarrow A$. Then $\mathrm{M}(A,A+B,C) \rightsquigarrow \{A,C\}$ by Lemma III.4f. We show that this leads to a contradiction. Case 3 handles the remaining possibility.

Case 1: $\mathrm{M}(A,B,C) \rightsquigarrow A$ and $\mathrm{M}(A,A+B,C) \rightsquigarrow A$: Lemma III.2c (taking $a=A$, $b=B$, $c=C$) implies $n^{(4)}(A,B,C)$ is a destination node of $\mathcal{N}_1$, as shown.



Then $\mathrm{M}(A,A+B,C) \rightsquigarrow A$ and Lemma III.4g (taking $D = A+C$) give $\mathrm{M}(C,A+C,A+B) \rightsquigarrow \{A+C,B\}$. Lemma III.4f and $\mathrm{M}(A,B,C) \rightsquigarrow A$ imply $\mathrm{M}(A,B,A+C) \rightsquigarrow \{A,B\}$. Lemma III.4h and $\mathrm{M}(A,A+B,C) \rightsquigarrow A$ imply $\mathrm{M}(A,B,A+C) \rightsquigarrow \{A,B+C\}$. Thus $\mathrm{M}(A,B,A+C) \rightsquigarrow A$. Lemma III.4f and $\mathrm{M}(A,A+B,C) \rightsquigarrow A$ imply $\mathrm{M}(A,A+B,A+C) \rightsquigarrow \{A,A+B\}$. Lemma III.4f and $\mathrm{M}(A,B,A+C) \rightsquigarrow A$ imply $\mathrm{M}(A,A+B,A+C) \rightsquigarrow \{A,A+C\}$. Thus $\mathrm{M}(A,A+B,A+C) \rightsquigarrow \{A,A+B\} \cap \{A,A+C\} = \{A\}$. Lemma III.4g and $\mathrm{M}(A,A+B,A+C) \rightsquigarrow A$ imply $\mathrm{M}(C,A+C,A+B) \rightsquigarrow \{B,C\}$. Thus we conclude that $\mathrm{M}(A+C,A+B,C) \rightsquigarrow \{B,C\} \cap \{B,A+C\} = \{B\}$. Together, these imply that the labels of the four inputs to the destination node $n^{(4)}(A,B,C)$ above lie in the set $\{A,B\}$, which is not complete, contradicting the solvability of $\mathcal{N}_1$. So Case 1 is impossible.

We omit the proofs that Cases 2 and 3 are also impossible.

Case 2: $\mathrm{M}(A,B,C) \rightsquigarrow A$ and $\mathrm{M}(A,A+B,C) \rightsquigarrow C$.

Case 3: $\mathrm{M}(A,B,C) \rightsquigarrow A+B+C$ for every complete set of linear functions $\{A,B,C\}$.

Since all three cases are impossible, there is no linear solution to the given network, even though it is solvable. ■

IV. Solvability for different alphabet sizes

If a network is solvable for a particular alphabet $\mathcal{A}$, then it is clearly solvable, using Cartesian products, for any alphabet of cardinality $|\mathcal{A}|^i$ and any integer $i \geq 1$. So, in this sense, solvability becomes somewhat "easier" as the alphabet size grows. In fact, the Li-Yeung-Cai linearity result in [3] guarantees not only a linear solution to a solvable network for large enough cardinality, but also a linear solution for any finite field larger than some specific size. This tends to add support to the notion that larger alphabets make solvability easier. However, the main result in this section shows that it is possible for a multicast network to be solvable for a certain alphabet but not solvable for some larger alphabet.

Theorem IV.2 considers network solutions with arbitrary alphabet sizes. No specific algebraic structure (e.g. a ring or field) is imposed on the alphabet, and hence the result does not depend on the notion of linearity.

First, we present a lemma about latin squares. A *latin square* of order n is an $n \times n$ square matrix, each row and column of which is a permutation of the integers $\{1,\ldots,n\}$. Two latin squares $\{a_{ij}\}$ and $\{b_{ij}\}$ are *orthogonal* if each ordered pair (a_{ij},b_{ij}) is distinct, for all i and j.

Lemma IV.1 ([2]). *Orthogonal latin squares exist if and only if the order of the matrices is neither 2 nor 6.*

The following theorem follows immediately from Lemma IV.3, which makes use of the multicast network $\mathcal{N}_2$ shown in Figure 2.

Theorem IV.2. *A multicast network that has a solution for a given alphabet might not have a solution for all larger alphabets.*

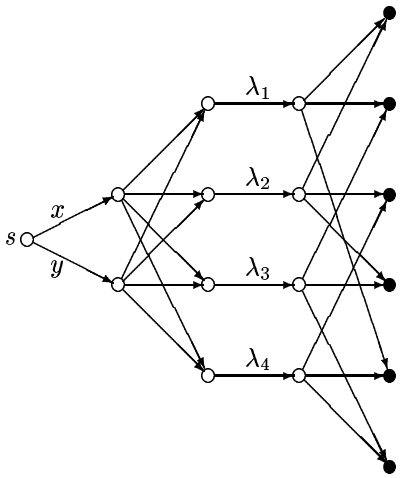


Fig. 2. Multicast network $\mathcal{N}_2$ with messages x, y, source node s, and 6 destination nodes. The edge labels λ_1, λ_2, λ_3, λ_4 are functions of x, y, and each of the 6 possible pairs of them constitute the in-edges of a unique destination node.

Lemma IV.3. *The multicast network $\mathcal{N}_2$ is solvable if and only if the alphabet size is neither 2 nor 6.*

Proof. The network $\mathcal{N}_2$ is solvable for a specific alphabet $\mathcal{A} = \{a_1, \ldots a_{|\mathcal{A}|}\}$ if and only if each of the 6 pairs $\{\lambda_i, \lambda_j\}$, for $1 \leq i, j \leq 4$ and $i \neq j$, is complete (so that the pair $\{x, y\}$ can be recovered for each destination node). Let $\mathcal{S}_k$ be the $|\mathcal{A}| \times |\mathcal{A}|$ square matrix whose entry in the i^{th} row and j^{th} column is n if $\lambda_k(a_i, a_j) = a_n$. Let us now temporarily assume $\lambda_1 = x$ and $\lambda_2 = y$, i.e. the projection functions.

Clearly $\{\lambda_1, \lambda_2\}$ is complete. The pair $\{\lambda_1, \lambda_3\}$ is complete if and only if for each $a_i \in \mathcal{A}$, no two elements of the i^{th} row of $\mathcal{S}_3$ are the same (for otherwise y could not be uniquely recovered from x and λ_3), i.e. every row of $\mathcal{S}_3$ must contain each of the integers in $\{1, \ldots, |\mathcal{A}|\}$ exactly once. Similarly, the pair $\{\lambda_2, \lambda_3\}$ is complete if and only if every column of $\mathcal{S}_3$ contains each of the integers in $\{1, \ldots, |\mathcal{A}|\}$ exactly once. Thus, the pairs $\{\lambda_1, \lambda_3\}$ and $\{\lambda_2, \lambda_3\}$ are both complete if and only if $\mathcal{S}_3$ is a latin square of order $|\mathcal{A}|$. A similar argument shows that the pairs $\{\lambda_1, \lambda_4\}$ and $\{\lambda_2, \lambda_4\}$ are both complete if and only if $\mathcal{S}_4$ is a latin square of order $|\mathcal{A}|$. Now, the pair $\{\lambda_3, \lambda_4\}$ is complete if and only if for all $i, j \in \{1, \ldots, |\mathcal{A}|\}$, the ordered pair of $(i, j)^{th}$ entries in $\mathcal{S}_3$ and $\mathcal{S}_4$ does not repeat at any other location in the two matrices (i.e. allowing unique recovery of i and j given the $(i, j)^{th}$ entries), if and only if each of the $|\mathcal{A}|^2$ pairs of integers from $\{1, \ldots, |\mathcal{A}|\}$ appears exactly once in the matrices $\mathcal{S}_3$ and $\mathcal{S}_4$ at the same positions. Hence, all 6 pairs $\{\lambda_i, \lambda_j\}$ are complete if and only if $\mathcal{S}_3$ and $\mathcal{S}_4$ are orthogonal latin squares. So, by Lemma IV.1 we conclude that $\mathcal{N}_2$ is solvable for all alphabets $\mathcal{A}$ such that $|\mathcal{A}| \notin \{2, 6\}$, and is not solvable for $|\mathcal{A}| \in \{2, 6\}$ under the assumptions $\lambda_1 = x$ and $\lambda_2 = y$.

Now let us relax the assumptions that $\lambda_1 = x$ and $\lambda_2 = y$ and suppose that $\mathcal{N}_2$ has a solution. Let $w = \lambda_1$ and $z = \lambda_2$. Then the pair $\{w, z\}$ is complete, so that x and y can each be recovered from w and z. Thus, since λ_3 and λ_4 are each functions of x and y, they are also functions of w and z. Formally, since the mapping $(\lambda_1, \lambda_2) : \mathcal{A}^2 \to \mathcal{A}^2$ is a bijection, it has an inverse h, and then λ_3 and λ_4 can be identified with $\hat{\lambda}_3 = \lambda_3 \circ h$ and $\hat{\lambda}_4 = \lambda_4 \circ h$, respectively. Matrices $\hat{\mathcal{S}}_3$ and $\hat{\mathcal{S}}_4$ can be defined for $\hat{\lambda}_3$ and $\hat{\lambda}_4$ analogously as before. Now the same argument as earlier implies that the 6 pairs $\{w, z\}$, $\{w, \hat{\lambda}_3\}$, $\{w, \hat{\lambda}_4\}$, $\{z, \hat{\lambda}_3\}$, $\{z, \hat{\lambda}_4\}$, $\{\hat{\lambda}_3, \hat{\lambda}_4\}$ are each complete if and only if the matrices $\hat{\mathcal{S}}_3$ and $\hat{\mathcal{S}}_4$ are orthogonal latin squares. Thus, if $\mathcal{N}_2$ were solvable for $|\mathcal{A}| \in \{2, 6\}$ then there would exist a pair of orthogonal latin squares of order 2 or 6, contradicting their known nonexistence. ■

Note that if we allow coding over two units of time (i.e., two uses of the network) then a linear solution exists, by choosing edge-function vectors $\lambda_1 = \binom{x_1}{x_2}$, $\lambda_2 = \binom{y_1}{y_2}$, $\lambda_3 = \left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)\binom{x_1}{x_2} + \left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)\binom{y_1}{y_2}$, $\lambda_4 = \left(\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right)\binom{x_1}{x_2} + \left(\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right)\binom{y_1}{y_2}$, where the message vectors are $\binom{x_1}{x_2}$ and $\binom{y_1}{y_2}$. This vector solution is valid for any alphabet size by viewing $\mathcal{A}$ as the ring of integers modulo $|\mathcal{A}|$ (i.e., $x_1, x_2, y_1, y_2 \in \mathcal{A}$). This is true even though $\mathcal{N}_2$ does not have a scalar solution for $|\mathcal{A}| \in \{2, 6\}$.

The multicast network in Figure 2 was one member in a family of networks used in [4] to show that the minimum alphabet size of multicast network solutions might have to be at least about the square root of the number of destination nodes. It was also used in independent work in [5] in examining codes over multiple time units, where a binary vector linear solution was given. Our characterization of which alphabet sizes admit solutions to $\mathcal{N}_2$ gives some more insight about the role of alphabet sizes and solvability.

V. Acknowledgment

The authors would like to thank Søren Riis for providing a preprint of his manuscript [5], and Raymond Yeung for some helpful pointers.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. IT-46, no. 4, pp. 1204–1216, July 2000.

[2] R. C. Bose, S. S. Shrikhande, E. T. Parker, "Further results on the construction of mutually orthogonal latin squares and the falsity of Euler's conjecture", *Canadian Journal of Mathematics*, vol. 12, pp. 189-203, 1960.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. IT-49, no. 2, pp. 371-381, February 2003.

[4] A. Rasala Lehman and E. Lehman, "Complexity classification of network information flow problems", *41st Annual Allerton Conference on Communication Control and Computing*, Monticello, IL, October 2003.

[5] S. Riis, "Linear versus non-linear boolean functions in network flow", *preprint* (available on-line at http://nick.dcs.qmul.ac.uk/∼smriis), December 2003.