

2015

Capacity of Sum-networks for Different Message Alphabets

Ardhendu Tripathy

Iowa State University, ardhendu@iastate.edu

Aditya Ramamoorthy

Iowa State University, adityar@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/ece_pubs



Part of the [Electrical and Computer Engineering Commons](#)

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/ece_pubs/106. For information on how to cite this item, please visit <http://lib.dr.iastate.edu/howtocite.html>.

This Article is brought to you for free and open access by the Electrical and Computer Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering Publications by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Capacity of Sum-networks for Different Message Alphabets

Ardhendu Tripathy and Aditya Ramamoorthy

Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011

Email: {ardhendu,adityar}@iastate.edu

Abstract—A *sum-network* is a directed acyclic network in which all terminal nodes demand the ‘sum’ of the independent information observed at the source nodes. Many characteristics of the well-studied multiple-unicast network communication problem also hold for sum-networks due to a known reduction between instances of these two problems. Our main result is that unlike a multiple unicast network, the coding capacity of a sum-network is dependent on the message alphabet. We demonstrate this using a construction procedure and show that the choice of a message alphabet can reduce the coding capacity of a sum-network from 1 to close to 0.

Index terms—capacity, function computation, linear network coding, network coding, sum-network.

I. INTRODUCTION

Function computation using network coding is an extensively studied research area [1], [2], [3], [4]. The typical setting is one of directed acyclic networks with error-free links, a set of source nodes that generate independent and identically distributed information and a set of terminal nodes with specific demands. The work of [1], [2] considers general functions, but networks with only one terminal. A different line of work considers networks with multiple terminals that each need a simple function such as the sum [3], [4].

In this work, we consider sum-networks. There are several results about the solvability of multiple unicast networks, many of which also hold for sum-networks by virtue of a reduction from a multiple unicast instance to a sum-network instance as described in [4]. Specifically, it was shown in [5] that linear network coding in the most general sense is unable to attain the coding capacity of multiple unicast networks. Reference [6] showed that there exists a multiple unicast network which is scalar linear solvable over a finite field \mathcal{F} if and only if a corresponding collection of integer-coefficient polynomials have a common root in \mathcal{F} . Analogous statements for sum-networks were noted to be true in [4] because of the equivalence. Reference [7] showed that the coding capacity of a general multiple unicast network is independent of the alphabet used.

A network code for a sum-network is said to have rate m/n if in n time slots, one can multicast the sum m times to all the terminals. A network is called solvable if it has a (m, m) code and not solvable otherwise. Prior work has investigated the effect of field characteristic on sum-network capacity.

Specifically, [4] constructs sum-networks whose solvability depends on the field characteristic under the restriction of linear network coding. Given a ratio $p/q < 1$, it was shown in [8] that there exist sum-networks which have p/q as their coding capacity. Reference [9] described a systematic construction procedure for sum-networks with capacity p/q ; these are typically smaller than those in [8]. The capacity of these sum-networks however was independent of the message alphabet.

A. Main contribution

- In this work, we construct sum-networks whose capacity depends on the characteristic of the finite field \mathcal{F} chosen as the message alphabet. As a specific example, we construct sum-networks for which the capacity is 1 if $ch(\mathcal{F}) = 2$ and can be made arbitrarily close to 0 if $ch(\mathcal{F}) \neq 2$.

The problem is formally posed in section II, our construction for the class of sum-networks is explained in section III. A cut-set bound on the coding capacity is described in section IV and a linear network code that attains this rate is described in section V.

II. PROBLEM FORMULATION

We consider communication over a directed acyclic graph (DAG) $G = (V, E)$ where V is the set of nodes and $E \in V \times V$ are the edges denoting the delay-free communication links between them. Subset $S \subset V$ denotes the source nodes and $T \subset V$ denotes the terminal nodes. The source nodes have no incoming edges and the terminal nodes have no outgoing edges. Each source node $s_i \in S$ generates an independent random process X_i , such that the sequence of random variables X_{i1}, X_{i2}, \dots indexed by time are i.i.d. and each X_{ij} takes values that are uniformly distributed over a finite alphabet \mathcal{F} that is assumed to be a finite field such that $|\mathcal{F}| = q$; the characteristic of \mathcal{F} will be denoted by $ch(\mathcal{F})$. An edge starting at u and ending in v in the DAG will be denoted as the ordered pair (u, v) . We define $head(u, v) = v$ and $tail(u, v) = u$. Each edge is of unit capacity and can transmit one symbol from \mathcal{F} per unit time. We use the notation $In(v)$ to represent the set of incoming edges at node $v \in V$. If $e = (u, v)$, then we set $In(e) = In(u)$.

A network code is an assignment of encoding functions (we call these the local encoding functions) to each edge in E and a decoding function to each terminal in T . The local

This work was supported in part by NSF grants CCF-1320416 and CCF-1149860.

encoding function for an edge connected to a set of sources depends only those particular source values. Likewise, the local encoding function for an edge that is not connected to any source depends on the values received on its incoming edges and the decoding function for a terminal depends only on its incoming edges. As we consider directed acyclic networks, it can be seen that we can also define a global encoding function that expresses the value transmitted on an edge in terms of the source values.

- Local encoding function for edge e .

$$\begin{aligned}\tilde{\phi}_e &: \mathcal{F}^m \rightarrow \mathcal{F}^n \quad \text{if } \text{tail}(e) \in S, \\ \tilde{\phi}_e &: \mathcal{F}^{n|\ln(\text{tail}(e))|} \rightarrow \mathcal{F}^n \quad \text{if } \text{tail}(e) \notin S.\end{aligned}$$

- Decoding function for the terminal $t_i \in T$.

$$\psi_{t_i} : \mathcal{F}^{n|\ln(t_i)|} \rightarrow \mathcal{F}^m$$

A network code is linear if all the edge and decoding functions are \mathcal{F} -linear. For the sum-networks that we consider, a (m, n) fractional network code solution over \mathcal{F} is such that the sum (over \mathcal{F}) of m source symbols can be communicated to all the terminals in n units of time. The rate of this network code is defined to be m/n . A network is said to be solvable if it has a (m, m) network coding solution for some $m \geq 1$. A network is said to have a scalar solution if it has a $(1, 1)$ solution. The supremum of all achievable rates is called the capacity of the network.

III. CONSTRUCTION USING BIBDS

We begin by defining a $2 - (v, k, \lambda)$ balanced incomplete block design (BIBD) [10]. It is a set system $\mathcal{D} = (P, \mathcal{B})$ which has the following components.

- *Points*: A set P consisting of v elements (called points), indexed in arbitrary order as $P = \{p_1, p_2, \dots, p_v\}$.
- *Blocks*: A set \mathcal{B} of size b whose elements are k -subsets of P such that $\mathcal{B} = \{B_1, B_2, \dots, B_b\}$. \mathcal{B} satisfies the following *regularity* property. For $p_i, p_j \in P, i \neq j$,

$$|\{B \in \mathcal{B} : p_i \in B, p_j \in B\}| = \lambda.$$

For such a design, we can define an incidence matrix A which is a $v \times b$ $(0, 1)$ -matrix that records the incidence between points and blocks, i.e.,

$$A(i, j) = \begin{cases} 1 & \text{if } p_i \in B_j, \\ 0 & \text{otherwise.} \end{cases}$$

It can be shown that each point is present in a fixed number of blocks (denoted by r). The following relations can be shown.

$$\begin{aligned}r &= \frac{\lambda(v-1)}{k-1}, \quad \text{and} \\ bk &= vr.\end{aligned}$$

For any $p \in P$ and $B \in \mathcal{B}$, let

$$\begin{aligned}\langle p \rangle &= \{B \in \mathcal{B} : p \in B\}, \text{ and} \\ \langle B \rangle &= \cup_{p \in B} \langle p \rangle = \cup_{p \in B} \{B' \in \mathcal{B} : p \in B'\}.\end{aligned}$$

Thus we have $|\langle p \rangle| = r, \forall p \in P$. For the case of $\lambda = 1$, we have that for any $p, p' \in B, \langle p \rangle \cap \langle p' \rangle = B$, i.e., a pair of points appear in a unique block.

Example 1: We describe the components of a $2 - (7, 3, 1)$ design, which is also called a Fano plane. Letting numerals denote points and alphabets denote blocks for this design, we can write:

$$\begin{aligned}P &= \{1, 2, 3, 4, 5, 6, 7\}, \quad \mathcal{B} = \{A, B, C, D, E, F, G\}, \\ A &= \{1, 2, 3\}, \quad B = \{3, 4, 5\}, \quad C = \{1, 5, 6\}, \quad D = \{1, 4, 7\}, \\ E &= \{2, 5, 7\}, \quad F = \{3, 6, 7\}, \quad G = \{2, 4, 6\}.\end{aligned}$$

The corresponding incidence matrix A is shown below.

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \quad (1)$$

We now construct a sum-network $G = (V, E)$ from any BIBD \mathcal{D} . We first describe the components of the vertex set V .

- 1) *Source node set*: S contains $v+b$ elements corresponding to points and blocks in \mathcal{D} , i.e.,

$$\begin{aligned}S &= \{s_{p_1}, s_{p_2}, \dots, s_{p_v}, s_{B_1}, s_{B_2}, \dots, s_{B_b}\} \\ &= \{s_p : p \in P\} \cup \{s_B : B \in \mathcal{B}\}.\end{aligned}$$

Each source node s_{p_i} (or s_{B_j}) observes an independent unit-entropy random process X_{p_i} (or X_{B_j}), respectively. The set of all source processes (or ‘‘sources’’) is $X = \{X_p : p \in P\} \cup \{X_B : B \in \mathcal{B}\}$.

- 2) *Terminal node set*: T also contains $v+b$ elements, in the same manner as S , i.e.,

$$T = \{t_p : p \in P\} \cup \{t_B : B \in \mathcal{B}\}.$$

- 3) *Intermediate nodes*: In addition to the above there are $2v$ vertices which are elements of $M^H \cup M^T$ (superscripts H and T denote head and tail, respectively) where

$$M^H = \{m_1^h, m_2^h, \dots, m_v^h\} \text{ and } M^T = \{m_1^t, m_2^t, \dots, m_v^t\}.$$

We set $V = S \cup T \cup M^H \cup M^T$ and thus V contains $2(v+b)+2v$ vertices. The edge set E of the directed acyclic network G has the following components.

- 1) *Bottleneck edges*: We introduce v unit-capacity edges $e_i = (m_i^t, m_i^h), i \in \{1, 2, \dots, v\}$. Thus the sets M^H and M^T denote the set of heads and tails of the bottleneck edges respectively. We also make the following connections for all $p_i \in P$

- (s_{p_i}, m_i^t) and (s_{B_j}, m_i^t) for all $B_j \in \langle p_i \rangle$,
- (m_i^h, t_{p_i}) and (m_i^h, t_{B_j}) for all $B_j \in \langle p_i \rangle$.

Thus, each m_i^t has $r+1$ incoming edges and each m_i^h has $r+1$ outgoing edges. Denoting the set of edges introduced in this step as M , we get $|M| = v+2v(r+1)$.

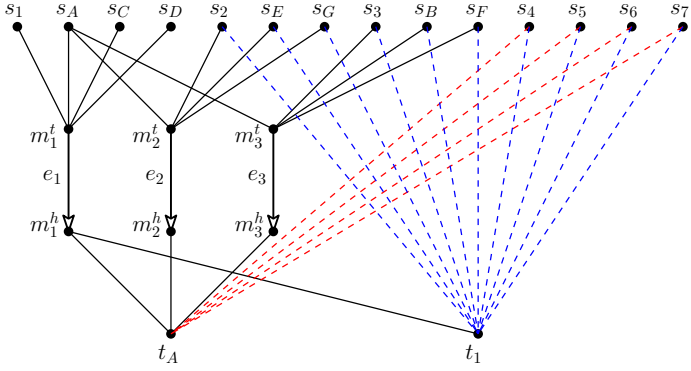


Fig. 1. Figure showing a part of the sum-network constructed using Fano plane. The 14 source nodes are shown at the top of the figure and 2 out of the 14 terminal nodes are depicted at the bottom. All edges have unit capacity and point downward. The 3 edges with the arrowheads (e_1, e_2, e_3) are the ‘bottleneck edges’. The red dashed lines are the ‘direct edges’ for terminal node t_A and the blue dashed lines are the ‘direct edges’ for terminal t_1 .

2) *Direct edges*: The remaining edges are those that are not incident on either the tail or the head of the bottleneck edges and are referred to as the direct-edge set D . It consists of the following unit-capacity edges for every $p_i \in P$ and $B_j \in \mathcal{B}$,

- (s_{p_i}, t_{p_i}) for all $p_i \neq p_i$,
- (s_{B_i}, t_{p_i}) for all $B_i \notin \langle p_i \rangle$,
- (s_{p_i}, t_{B_j}) for all $p_i \notin B_j$, and
- (s_{B_i}, t_{B_j}) for all $B_i \notin \langle B_j \rangle$.

We then set $E = M \cup D$. This completes the construction of the network G . It can be verified now that each terminal is connected to every source node in the network by at least one path. To denote a general element of the set E we will use the letter e . Given a set of local encoding functions for each edge, denoted $\tilde{\phi}_e$, we can define the corresponding global encoding function for edge e , denoted ϕ_e . In what follows, we will use the notation $\phi_e(X)$ to denote the n -length vector that is transmitted on edge e . For convenience, we define

$$\begin{aligned} \phi_{\text{In}(s_{p_i})}(X) &= [X_{p_i}^T \quad \mathbf{0}_{1 \times (n-m)}]^T, \text{ for all } p_i \in P \\ \phi_{\text{In}(s_{B_j})}(X) &= [X_{B_j}^T \quad \mathbf{0}_{1 \times (n-m)}]^T, \text{ for all } B_j \in \mathcal{B} \\ \phi_{\text{In}(v)}(X) &= \{\phi_e(X) : e \in \text{In}(v)\}, \text{ for all } v \in V \setminus S \\ \phi_{\text{In}(e_i)}(X) &= \{\phi_e(X) : e \in \text{In}(m_i^t)\}. \end{aligned}$$

As will be apparent for our networks, nontrivial encoding functions will only be required on the bottleneck edges, $e_i, i = 1, \dots, v$. For brevity, we set $\phi_i(X) = \phi_{e_i}(X)$.

Example 2: Using the BIBD described in example 1, we construct a sum-network here. It has

- fourteen source nodes $\{s_1, \dots, s_7, s_A, \dots, s_G\}$,
- fourteen terminal nodes $\{t_1, \dots, t_7, t_A, \dots, t_G\}$,
- fourteen intermediate nodes $\{m_1^h, \dots, m_7^h, m_1^t, \dots, m_7^t\}$, and
- seven bottleneck edges $\{e_1 = (m_1^t, m_1^h), \dots, e_7\}$.

There will also be direct edges and edges connecting the bottlenecks to the sources and terminals. Part of the constructed sum-network is shown in figure 1. We have shown the

construction procedure on only three bottleneck edges, namely e_1, e_2, e_3 and two terminals t_1, t_A . The edges coming into and going out of other bottlenecks and terminals can be constructed similarly.

IV. UPPER BOUND ON CODING CAPACITY OF CONSTRUCTED SUM-NETWORK

Let $H(Y)$ be the entropy function for a random variable Y . We let $H(Y_1, Y_2, \dots, Y_l) = H(\{Y_1, Y_2, \dots, Y_l\}) = H(\{Y_i\}_1^l)$ for any $l > 1$. Suppose that there exists a (m, n) fractional network code assignment, i.e. $\tilde{\phi}_e$ for $e \in E$ (and corresponding global encoding functions $\phi_e(X)$) and decoding functions ψ_t for $t \in T$ so that all the terminals in T can recover the sum of sources denoted by $Z = \sum_{p \in P} X_p + \sum_{B \in \mathcal{B}} X_B$. The following two lemmas demonstrate that certain partial sums can be computed by observing subsets of the bottleneck edges (see Appendix VII-A, VII-B for proofs).

Lemma 1: The value $X'_i = X_{p_i} + \sum_{B \in \langle p_i \rangle} X_B$ can be obtained from $\phi_i(X)$ for all $i = 1, \dots, v$.

Lemma 2: The value of $\sum_{p \in B_j} X_p + \sum_{B \in \langle B_j \rangle} X_B$ can be computed from the set of values $\{\phi_i(X) : p_i \in B_j\}$ for all $B_j \in \mathcal{B}$.

Theorem 1: For a $2-(v, k, 1)$ design \mathcal{D} , the coding capacity of the sum-network obtained using the given construction is at most 1.

Proof: Let $X'_i = X_{p_i} + \sum_{B \in \langle p_i \rangle} X_B$ and $\phi_i(X)$ is the function transmitted on the i -th bottleneck. Under a (m, n) fractional network code, we have that $H(X'_i) = m \log_2 q$ bits and $H(\phi_i(X)) \leq n \log_2 q$ bits. We then have

$$\begin{aligned} H(\{\phi_i(X)\}_1^v) &\leq \sum_{i=1}^v H(\phi_i(X)) \leq vn \log_2 q \text{ bits, and} \\ H(\{\phi_i(X)\}_1^v) &= I(\{\phi_i(X)\}_1^v; \{X'_i\}_1^v) + H(\{\phi_i(X)\}_1^v | \{X'_i\}_1^v) \\ &= H(\{X'_i\}_1^v) - H(\{X'_i\}_1^v | \{\phi_i(X)\}_1^v) + H(\{\phi_i(X)\}_1^v | \{X'_i\}_1^v). \end{aligned} \quad (2)$$

We assume that all source random variables are independent and uniformly distributed over \mathcal{F}^m . That implies X'_i are also uniform i.i.d. over \mathcal{F}^m for all i (see Appendix VII-C). Thus, we have that $H(\{X'_i\}_1^v) = vm \log_2 q$ bits. We expand the second term of (2) as

$$H(\{X'_i\}_1^v | \{\phi_i(X)\}_1^v) = \sum_{i=1}^v H(X'_i | \{X'_j\}_1^{i-1}, \{\phi_j(X)\}_1^v).$$

Also, by lemma 1 we have that for all $i = 1, 2, \dots, v$

$$H(X'_i | \phi_i(X)) = 0 \implies H(X'_i | \{X'_j\}_1^{i-1}, \{\phi_j(X)\}_1^v) = 0.$$

Using these in (2) we get that

$$\begin{aligned} vm \log_2 q + H(\{\phi_i(X)\}_1^v | \{X'_i\}_1^v) &\leq vn \log_2 q \\ \text{i.e., } vm \log_2 q \leq vn \log_2 q &\implies \frac{m}{n} \leq 1. \end{aligned}$$

Theorem 2: For a $2-(v, k, 1)$ design \mathcal{D} , the coding capacity of the sum-network obtained using the above construction is at most $\frac{k(k-1)}{k(k-1)+v-1}$ if $ch(\mathcal{F}) \nmid (k-1)$, i.e., $ch(\mathcal{F})$ is not a divisor of $(k-1)$. ■

Proof: Let $X'_i = X_{p_i} + \sum_{B \in \langle p_i \rangle} X_B$ and $\phi_i(X)$ be the function transmitted on the i -th bottleneck. Consider terminal t_B , for any $B \in \mathcal{B}$. Its incoming edges include $\{e_p : p \in B\}$. Then, from lemma 1 it can evaluate

$$\sum_{p_i \in B} X'_i = \sum_{p_i \in B} X_{p_i} + \sum_{p_i \in B} \sum_{B' \in \langle p_i \rangle} X_{B'}. \quad (3)$$

Since \mathcal{D} is a 2-design with $\lambda = 1$, for any $p, p' \in B$, we have that $\langle p \rangle \cap \langle p' \rangle = B$. By definition, $B \in \langle p \rangle \forall p \in B$ and $|B| = k$. These imply that the RHS in (3) simplifies to

$$\sum_{p_i \in B} X_{p_i} + kX_B + \sum_{B' \in \langle B \rangle \setminus B} X_{B'} \quad (4)$$

where $\langle B \rangle \setminus B = \{B' \in \langle B \rangle : B' \neq B\}$. From Lemma 2, t_B can evaluate

$$\sum_{p \in B} X_p + \sum_{B' \in \langle B \rangle} X_{B'} = \sum_{p_i \in B} X_{p_i} + X_B + \sum_{B' \in \langle B \rangle \setminus B} X_{B'}$$

from the bottleneck edges. This implies that terminal t_B can compute the value of $(k-1)X_B$. Since $ch(\mathcal{F}) \nmid (k-1)$, t_B can compute the value of X_B .

From lemma 1, t_{p_i} can evaluate X'_i for all $p_i \in P$. Consider a hypothetical terminal t_* which can observe the functions $\phi_i(X)$ for all $p_i \in P$. Since t_B can find the value of X_B from the bottleneck edges incident to it, so can t_* . Similarly, t_* can find X'_i for all $p_i \in P$ (since t_{p_i} can find it). Then t_* is able to find X_p and X_B for any p and B . Using a cut-based argument for t_* , we get that

$$q^{(v+b)m} \leq q^{vn} \implies \frac{m}{n} \leq \frac{v}{v+b}.$$

For \mathcal{D} , $b = \frac{v(v-1)}{k(k-1)}$. Substituting this above gives us the result. ■

V. LINEAR NETWORK CODE FOR CONSTRUCTED SUM-NETWORK

We now describe (m, n) -fractional linear network codes whose rate matches the upper bound described above.

A. Case when $ch(\mathcal{F}) \mid (k-1)$

In this case, we set $m = n = 1$. We define the edge functions and terminal functions as follows ($i = 1 \dots v; j = 1 \dots b$)

$$\begin{aligned} \phi_i(X) &= X'_i = X_{p_i} + \sum_{B \in \langle p_i \rangle} X_B, \\ \psi\left(\phi_{\text{In}(t_{p_i})}(X)\right) &= \phi_i(X) + \sum_{p \neq p_i} X_p + \sum_{B \notin \langle p_i \rangle} X_B, \end{aligned} \quad (5)$$

$$\psi\left(\phi_{\text{In}(t_{B_j})}(X)\right) = \sum_{p_i \in B_j} \phi_i(X) + \sum_{p \notin B_j} X_p + \sum_{B \notin \langle B_j \rangle} X_B. \quad (6)$$

All other edges e in G are such that $|\text{In}(e)| = 1$ and we set $\phi_e(X) = \phi_{\text{In}(e)}(X)$. The following is proved in VII-D.

Lemma 3: The network code described above enables all the terminals to compute the sum of sources $Z = \sum_{p \in P} X_p + \sum_{B \in \mathcal{B}} X_B$.

B. Case when $ch(\mathcal{F}) \nmid (k-1)$

Let $v' = v - x$, where $v \equiv x \pmod{k}$, and $b' = rv'/k$. We describe a (m, n) fractional linear network code with $m = v', n = v' + b' = v' + v'(v-1)/k(k-1)$. Note that the ratio m/n is then equal to the upper bound described in theorem 2. In order to specify the linear network code for this case, we first ‘‘color’’ the incidence matrix A of the underlying BIBD \mathcal{D} to obtain a matrix A_c of the same size as A . The coloring procedure assigns a successive natural number to each non-zero element present in every column of A . The numbers in each column start from 1 and go up to k .

Example 3: For an adjacency matrix as defined in (1), the coloring procedure returns

$$A_c = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 1 \\ 3 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 & 0 & 0 & 2 \\ 0 & 3 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 & 0 \end{bmatrix}. \quad (7)$$

We arrange the r blocks incident to each p_i in increasing order of their block indices and denote them as $(B'_{i1}, B'_{i2}, \dots, B'_{ir})$. Note that any $B'_{\alpha\beta}$ corresponds to a unique $B_\gamma \in \mathcal{B}$ where

$$\gamma = \min_{t \in \{1, 2, \dots, b\}} t \text{ such that } \sum_{l=1}^t A(\alpha, l) = \beta. \quad (8)$$

We can then define a *selector function* $U_{\alpha\beta}$ for any $\alpha \in \{1, 2, \dots, v\}$ and $\beta \in \{1, 2, \dots, r\}$ as follows.

$$U_{\alpha\beta} = \left[\mathbf{0}_{\frac{v'}{k} \times \frac{v'}{k}(d-1)} \quad I_{\frac{v'}{k}} \quad \mathbf{0}_{\frac{v'}{k} \times \frac{v'}{k}(k-d)} \right] X_{B'_{\alpha\beta}}, \text{ where } B'_{\alpha\beta} = B_\gamma \text{ and } d = A_c(\alpha, \gamma), \gamma \in \{1, 2, \dots, b\}.$$

Here I_a denotes the identity matrix of dimension $a \times a$ and $\mathbf{0}_{a \times b}$ denotes a zero matrix of dimension $a \times b$.

We now define the edge functions for the linear network code. We let

$$\phi_i(X) = [X'_i{}^T \quad U_{i1}^T \quad U_{i2}^T \quad \dots \quad U_{ir}^T]^T \quad (9)$$

where $X'_i = X_{p_i} + \sum_{B \in \langle p_i \rangle} X_B$. All other edges e in G are such that $|\text{In}(e)| = 1$ and we set $\phi_e(X) = \phi_{\text{In}(e)}(X)$. By definition it is clear that $\phi_i(X)$ is a function of $\phi_{\text{In}(e_i)}(X)$. Furthermore, it is easy to see that $\phi_i(X)$ is of dimension $n \times 1$. This is because X'_i has dimension $v' \times 1$, and U_{ij} for all j is of dimension $\frac{v'}{k} \times 1$, so that X'_i has dimension $v' + r \left(\frac{v'}{k}\right) = v' + b' = n$.

The terminal functions used in this network code are defined as follows. For any $p_i \in P$, let

$$Z_1 = \sum_{p \neq p_i} X_p, \quad Z_2 = \sum_{B \notin \langle p_i \rangle} X_B$$

such that t_{p_i} can evaluate Z_1 and Z_2 from its direct edges. We define the terminal functions for $t_{p_i}, p_i \in P$ as

$$\begin{aligned} \psi\left(\phi_{\text{In}(t_{p_i})}(X)\right) &= [I_{v'} \quad \mathbf{0}_{v' \times b'}] \phi_i(X) + Z_1 + Z_2 \\ &= X'_i + Z_1 + Z_2 = Z. \end{aligned}$$

For terminals of the form $t_{B_j}, B_j \in \mathcal{B}$, we first let

$$Z_1 = \sum_{p \notin B_j} X_p, \quad Z_2 = \sum_{B \notin \langle B_j \rangle} X_B$$

such that both Z_1 and Z_2 are available to t_{B_j} via direct edges. The terminal functions for $t_{B_j}, B_j \in \mathcal{B}$ are evaluated as

$$\begin{aligned} \psi \left(\phi_{\text{In}(t_{B_j})}(X) \right) &= \sum_{p_i \in B_j} [I_{v'} \quad \mathbf{0}_{v' \times b'}] \phi_i(X) + Z_1 + Z_2 \\ &\quad - (k-1) [U_{\alpha_1 \beta_1}^T \quad U_{\alpha_2 \beta_2}^T \quad \dots \quad U_{\alpha_k \beta_k}^T]^T \end{aligned} \quad (10)$$

where $U_{\alpha\beta}$ is the selector function and

$$A_c(\alpha_u, j) = u, \quad B'_{\alpha_u \beta_u} = B_j \text{ for all } u \in \{1, 2, \dots, k\}.$$

From definition it can be seen that all the elements in the RHS of (10) are contained in the set $\text{In}(t_{B_j})$. In addition we have the following lemma.

Lemma 4: For all $B_j \in \mathcal{B}$, we have that

$$X_{B_j} = [U_{\alpha_1 \beta_1}^T \quad U_{\alpha_2 \beta_2}^T \quad \dots \quad U_{\alpha_k \beta_k}^T]^T$$

Proof: It can be seen that $U_{\alpha_u \beta_u}$ consists of components of the source X_{B_j} . In addition, we have that

$$\begin{bmatrix} U_{\alpha_1 \beta_1} \\ U_{\alpha_2 \beta_2} \\ \vdots \\ U_{\alpha_k \beta_k} \end{bmatrix} = \begin{bmatrix} I_{\frac{v'}{k}} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{\frac{v'}{k}} & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & I_{\frac{v'}{k}} \end{bmatrix} X_{B_j} = X_{B_j}.$$

The proof of the next lemma appears in VII-E.

Lemma 5: All terminals $t_{B_j}, B_j \in \mathcal{B}$ can evaluate the sum of sources, i.e. $H \left(Z | \psi \left(\phi_{\text{In}(t_{B_j})}(X) \right) \right) = 0$.

Example 4: We describe the linear network code for the sum-network described in example 2. Here $k = 3$ and we describe a linear network code for the case when $ch(\mathcal{F}) \nmid (k-1) = 2$, i.e., \mathcal{F} has odd characteristic. Then $v' = b' = 6$. The linear network code has rate $m/n = 6/12 = 1/2$. To describe the network code we first obtain a coloring of the adjacency matrix which is described in (7).

Suppose X is a 6×1 vector. Let

$$X(a : b)^T = [\mathbf{0}_{(b-a+1) \times (a-1)} \quad I_{(b-a+1)} \quad \mathbf{0}_{(b-a+1) \times (6-b)}] X.$$

Then the edge vectors are as follows:

$$\begin{aligned} \phi_1(X) &= [X_1'^T \quad X_A(1 : 2) \quad X_C(1 : 2) \quad X_D(1 : 2)]^T, \\ \phi_2(X) &= [X_2'^T \quad X_A(3 : 4) \quad X_E(1 : 2) \quad X_G(1 : 2)]^T, \\ \phi_3(X) &= [X_3'^T \quad X_A(5 : 6) \quad X_B(1 : 2) \quad X_F(1 : 2)]^T, \\ \phi_4(X) &= [X_4'^T \quad X_B(3 : 4) \quad X_D(3 : 4) \quad X_G(3 : 4)]^T, \\ \phi_5(X) &= [X_5'^T \quad X_B(5 : 6) \quad X_C(3 : 4) \quad X_E(3 : 4)]^T, \\ \phi_6(X) &= [X_6'^T \quad X_C(5 : 6) \quad X_F(3 : 4) \quad X_G(5 : 6)]^T, \\ \phi_7(X) &= [X_7'^T \quad X_D(5 : 6) \quad X_E(5 : 6) \quad X_D(5 : 6)]^T. \end{aligned}$$

We now verify that $\psi \left(\phi_{\text{In}(t_C)}(X) \right) = Z$. The same exercise can be repeated for other terminals. For $t_C, B_j = C$ and

$$A_c(\alpha_1, 3) = 1 \implies \alpha_1 = 1, \quad B'_{1\beta_u} = C \implies \beta_u = 2.$$

Similarly, we can compute that $\alpha_2 = 5, \beta_2 = 2, \alpha_3 = 6, \beta_3 = 1$. This implies that

$$\begin{bmatrix} U_{\alpha_1 \beta_1} \\ U_{\alpha_2 \beta_2} \\ U_{\alpha_3 \beta_3} \end{bmatrix} = \begin{bmatrix} U_{12} \\ U_{52} \\ U_{61} \end{bmatrix} = \begin{bmatrix} X_C(1 : 2)^T \\ X_C(3 : 4)^T \\ X_C(5 : 6)^T \end{bmatrix} = X_C.$$

It can then be seen that t_C can compute the sum of all sources.

C. Sum-networks from Steiner triple systems

A Steiner triple system (STS) of order v is a $2 - (v, k, \lambda)$ design with $k = 3$ and $\lambda = 1$. The Fano plane is an example of a STS. It is well known that there exist Steiner triple systems for all orders $v \equiv 1$ or $3 \pmod{6}$ [10].

Theorem 3: Sum-networks constructed from STS with v points are such that their capacity equals 1 if $ch(\mathcal{F}) = 2$ and $\frac{6}{5+v}$ is $ch(\mathcal{F})$ is odd.

Proof: For a sum-network constructed using a STS, applying Theorem 1, Lemma 3, Theorem 2 and Lemma 5, we have the desired result. Thus, the message alphabet can greatly impact the coding capacity of a sum-network. ■

VI. CONCLUSION

We have constructed a family of sum-networks whose network coding capacity is dependent on the message alphabet (specifically, the characteristic of the finite field) chosen for communication. Previous results in this line described only the solvability of sum-networks for different alphabets using linear network coding. It was not clear how the coding capacity of a sum-network would be affected due to choice of message alphabet, if at all. The results in this paper address that issue.

REFERENCES

- [1] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Network coding for computing: Cut-set bounds," *IEEE Trans. on Info. Th.*, vol. 57, no. 2, pp. 1015–1030, Feb. 2011.
- [2] —, "Linear codes, target function classes, and network computing capacity," *IEEE Trans. on Info. Th.*, vol. 59, no. 9, pp. 5741–5753, Sept. 2013.
- [3] A. Ramamoorthy and M. Langberg, "Communicating the sum of sources over a network," *IEEE Journal on Selected Areas in Communication: Special Issue on In-network Computation: Exploring the Fundamental Limits*, vol. 31(4), pp. 655–665, 2013.
- [4] B. K. Rai and B. K. Dey, "On network coding for sum-networks," *IEEE Trans. on Info. Th.*, vol. 58, no. 1, pp. 50–63, 2012.
- [5] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. on Info. Th.*, vol. 51, no. 8, pp. 2745–2759, 2005.
- [6] —, "Linear network codes and systems of polynomial equations," *IEEE Trans. on Info. Th.*, vol. 54, no. 5, pp. 2303–2316, May 2008.
- [7] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity," *IEEE Trans. on Info. Th.*, vol. 52, no. 3, pp. 777–788, March 2006.
- [8] B. Rai and N. Das, "On the capacity of sum-networks," in *51st Annual Allerton Conference on Communication, Control, and Computing*, Oct 2013, pp. 1545–1552.
- [9] A. S. Tripathy and A. Ramamoorthy, "Sum-networks from undirected graphs: construction and capacity analysis," in *52nd Annual Allerton Conference on Communication, Control, and Computing*, 2014, pp. 651–658.

VII. APPENDIX

A. Proof of Lemma 1

We let for any $p_i \in P$

$$Z_1 = \sum_{p \neq p_i} X_p, Z_2 = \sum_{B \in \langle p_i \rangle} X_B \text{ and } Z_3 = \sum_{B \notin \langle p_i \rangle} X_B$$

such that $Z = X_{p_i} + Z_1 + Z_2 + Z_3$. By our assumption, we know that Z can be evaluated from $\phi_{\ln(t_{p_i})}(X)$ for all $i = 1, 2, \dots, v$, i.e., $H(Z|\phi_{\ln(t_{p_i})}(X)) = 0$.

Since $\{X_p : p \neq p_i\}$ and $\{X_B : B \notin \langle p_i \rangle\}$ determine the value of Z_1 and Z_3 respectively and are also a subset of $\phi_{\ln(t_{p_i})}(X)$, we get that

$$H(Z|\phi_{\ln(t_{p_i})}(X)) = 0 \implies H(X_{p_i} + Z_2|\phi_{\ln(t_{p_i})}(X)) = 0, \\ \text{i.e., } H(X_{p_i} + Z_2|\phi_i(X), \{X_p : p \neq p_i\}, \{X_B : B \notin \langle p_i \rangle\}) = 0.$$

Since all the sources are i.i.d., it can be verified that $X_{p_i} + Z_2$ is conditionally independent of both $\{X_p : p \neq p_i\}$ and $\{X_B : B \notin \langle p_i \rangle\}$ given $\phi_i(X)$. Hence, we have that $H(X_{p_i} + Z_2|\phi_i(X)) = 0$.

B. Proof of Lemma 2

We let for any $B_j \in \mathcal{B}$

$$Z_1 = \sum_{p \in B_j} X_p, Z_2 = \sum_{p \notin B_j} X_p, Z_3 = \sum_{B \in \langle B_j \rangle} X_B, Z_4 = \sum_{B \notin \langle B_j \rangle} X_B$$

such that $Z = Z_1 + Z_2 + Z_3 + Z_4$. By our assumption, for all $i = 1, 2, \dots, b$, $H(Z|\phi_{\ln(t_{B_j})}(X)) = 0$.

The sets $\{X_p : p \notin B_j\}$ and $\{X_B : B \notin \langle B_j \rangle\}$ determine the value of Z_2 and Z_4 respectively and are also subsets of $\phi_{\ln(t_{B_j})}(X)$. Hence, we have that

$$H(Z|\phi_{\ln(t_{B_j})}(X)) = 0 \implies H(Z_1 + Z_3|\phi_{\ln(t_{B_j})}(X)) = 0, \\ \text{where, } \phi_{\ln(t_{B_j})}(X) = \{\phi_i(X) : p_i \in B_j\} \cup \{X_p : p \notin B_j\} \\ \cup \{X_B : B \notin \langle B_j \rangle\}.$$

Since all the sources are i.i.d., it can be verified that $Z_1 + Z_3$ is conditionally independent of both $\{X_p : p \notin B_j\}$ and $\{X_B : B \notin \langle B_j \rangle\}$ given the set of random variables $\{\phi_i(X) : p_i \in B_j\}$. This gives us the result that $H(Z_1 + Z_3|\{\phi_i(X) : p_i \in B_j\}) = 0$.

C. Proof of claim in Theorem 1

We want to prove the identity

$$P(X'_1 = x'_1, \dots, X'_v = x'_v) = \prod_{i=1}^v P(X'_i = x'_i). \quad (11)$$

Let \mathcal{X} be the set of solutions to the following system of linear equations:

$$\begin{aligned} x_{p_1} + \sum_{B \in \langle p_1 \rangle} x_B &= x'_1 \\ x_{p_2} + \sum_{B \in \langle p_2 \rangle} x_B &= x'_2 \\ &\vdots \\ x_{p_v} + \sum_{B \in \langle p_v \rangle} x_B &= x'_v. \end{aligned}$$

Choosing any value from \mathcal{F}^m for each of the variables in $\{x_B : B \in \mathcal{B}\}$ fixes the value of x_{p_i} given x'_i for all $i \in \{1, 2, \dots, v\}$ and vice versa. The probability of the set in (11) is equal to the probability with which the set of random variables $\{\{X_{p_i}\}_1^v \cup \{X_{B_j}\}_1^b\}$ take values in \mathcal{X} . Since all random variables in $\{\{X_{p_i}\}_1^v \cup \{X_{B_j}\}_1^b\}$ are uniform i.i.d. over \mathcal{F}^m with $|\mathcal{F}| = q$ we can expand the LHS of (11) as

$$\sum_{\mathcal{X}} \prod_{i=1}^v P(X_{p_i} = x_{p_i}) \prod_{j=1}^b P(X_{B_j} = x_{B_j}) \\ = \left[\frac{1}{q^m}\right]^b \left[\frac{1}{q^m}\right]^v (q^m)^b = \left[\frac{1}{q^m}\right]^v.$$

Also, we have that

$$P(X'_i = x'_i) = \sum_{\mathcal{X}'} P(X_{p_i} = x_{p_i}) \prod_{B \in \langle p_i \rangle} P(X_B = x_B)$$

where \mathcal{X}' is the set of solutions to the variables $\{x_{p_i}, \{x_B\}_{B \in \langle p_i \rangle}\}$ such that $x'_i = x_{p_i} + \sum_{B \in \langle p_i \rangle} x_B$. Then

$$P(X'_i = x'_i) = \frac{1}{q^m} \left[\frac{1}{q^m}\right]^r (q^m)^r = \frac{1}{q^m},$$

and the RHS of (11) is

$$\prod_{i=1}^v P(X'_i = x'_i) = \left[\frac{1}{q^m}\right]^v.$$

D. Proof of Lemma 3

Consider $p_i \in P$, and let

$$Z_1 = \sum_{p \neq p_i} X_p, Z_2 = \sum_{B \in \langle p_i \rangle} X_B \text{ and } Z_3 = \sum_{B \notin \langle p_i \rangle} X_B$$

such that $\phi_i(X) = X_{p_i} + Z_2$. Then from (5) for $i = 1, 2, \dots, v$,

$$\psi(\phi_{\ln(t_{p_i})}(X)) = \phi_i(X) + Z_1 + Z_3 = X_{p_i} + Z_2 + Z_1 + Z_3 = Z.$$

Now we look at terminals of the form $t_{B_j}, B_j \in \mathcal{B}$. For this, we let for any $B_j \in \mathcal{B}$

$$\begin{aligned} Z_1 &= \sum_{p \in B_j} X_p, Z_2 = \sum_{p \notin B_j} X_p, \\ Z_3 &= \sum_{B \in \langle B_j \rangle \setminus B_j} X_B, \text{ and } Z_4 = \sum_{B \notin \langle B_j \rangle} X_B \end{aligned}$$

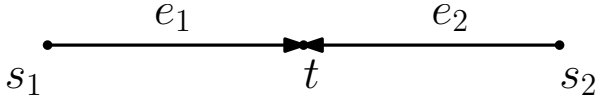


Fig. 2. A simple sum-network. Both edges can transmit one symbol in A from tail to head in one channel use.

such that $Z = X_{B_j} + Z_1 + Z_2 + Z_3 + Z_4$. Since \mathcal{D} is a 2-design with $\lambda = 1$, we have that for any $p, p' \in B$, $\langle p \rangle \cap \langle p' \rangle = B$. This implies that

$$\sum_{p_i \in B_j} \phi_i(X) = Z_1 + Z_3 + kX_{B_j}.$$

Since $ch(\mathcal{F}) \mid (k-1)$, we have that $kX_{B_j} = X_{B_j}$. Using this in (6) we get that for $j = 1, 2, \dots, b$

$$\psi\left(\phi_{\ln(t_{B_j})}(X)\right) = Z_1 + Z_3 + X_{B_j} + Z_2 + Z_4 = Z.$$

E. Proof of Lemma 5

We let

$$\begin{aligned} Z_1 &= \sum_{p \notin B_j} X_p, \quad Z_2 = \sum_{B \notin \langle B_j \rangle} X_B, \\ Z_3 &= \sum_{B \in \langle B_j \rangle \setminus B_j} X_B, \quad \text{and } Z_4 = \sum_{p \in B_j} X_p \end{aligned}$$

such that $Z = X_{B_j} + Z_1 + Z_2 + Z_3 + Z_4$. Then using (4),

$$\sum_{p_i \in B_j} [I_{v'} \quad \mathbf{0}_{v' \times b}] \phi_i(X) = \sum_{p_i \in B_j} X_i' = Z_3 + Z_4 + kX_{B_j}.$$

Using this and lemma 4 in (10) we get that

$$\psi(\phi_{\ln(t_{B_j})}(X)) = Z_3 + Z_4 + kX_{B_j} - (k-1)X_{B_j} + Z_1 + Z_2 = Z.$$

F. Remark about non-applicability of Theorem VI.5 in [7] for sum-networks

Theorem VI.5 in [7] states that the message capacity of a network is independent of the alphabet used. Consider a simple sum-network shown in Figure 2, terminal t wants to evaluate $X_1 + X_2$ where $X_1, X_2 \in A$ are random variables observed at source nodes s_1, s_2 respectively. We have a simple scalar network code (rate $k/n = 1, k = n = 1$) that satisfies the problem, described as follows.

1) Edge functions:

$$\begin{aligned} \phi_{e_1}(X_1) &\equiv f_{e_1}(X_1) = X_1, \\ \phi_{e_2}(X_2) &\equiv f_{e_2}(X_2) = X_2. \end{aligned}$$

2) Decoding function:

$$\begin{aligned} \psi(\phi_{e_1}(X_1), \phi_{e_2}(X_2)) &\equiv f_{t, \Sigma}(f_{e_1}(X_1), f_{e_2}(X_2)) \\ &= f_{e_1}(X_1) + f_{e_2}(X_2) \end{aligned}$$

where $\Sigma = X_1 + X_2$ is the only message terminal t is interested in.

We use the procedure outlined in [7] to extend the network code for another alphabet B . Let $A = GF(3), B = GF(2)$.

Setting $\epsilon = 2^{1-\gamma}/\log_2 3$ where $\gamma > 1$, we obtain the following values

$$t = 2^\gamma, n' = \left\lceil \frac{2^\gamma}{\log_2 3} \right\rceil \quad \text{and } k' = \lfloor n' \rfloor - 1.$$

Let $h_0 : B \rightarrow A$ be such that

$$h_0(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x = 1. \end{cases}$$

and let $\hat{h}_0 : A \rightarrow B$ such that $\hat{h}_0(h_0(x)) = x$ for all $x \in B$ and arbitrary otherwise. Then we can define an injection $\mathbf{h}_0 : B^{k'} \rightarrow A^t$ as the componentwise application of h_0 to each of the elements in the argument. That is

$$\mathbf{h}_0(b_1, b_2, \dots, b_{k'}) = [h_0(b_1) \quad h_0(b_2) \quad \dots \quad h_0(b_{k'}) \quad \mathbf{0}_{t-k'}]$$

where $b_1, b_2, \dots, b_{k'} \in B$ and $\mathbf{0}_{t-k'}$ is a zero vector with $t-k'$ components. We define $\hat{\mathbf{h}}_0 : A^t \rightarrow B^{k'}$ as

$$\hat{\mathbf{h}}_0(a_1, a_2, \dots, a_t) = [\hat{h}_0(a_1) \quad \hat{h}_0(a_2) \quad \dots \quad \hat{h}_0(a_{k'})]$$

where $a_1, a_2, \dots, a_t \in A$.

Also we let $\mathbf{h} : A^t \rightarrow B^{n'}$ be an arbitrary injection and $\hat{\mathbf{h}} : B^{n'} \rightarrow A^t$ is such that $\hat{\mathbf{h}}(\mathbf{h}(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x} \in A^t$ and arbitrary otherwise. We now use the extended network code to satisfy the sum network for when the source random variables take values in the alphabet $B^{k'}$. Suppose a particular realization of $\mathbf{X}_1 \in B^{k'}$ and $\mathbf{X}_2 \in B^{k'}$ is such that

$$\mathbf{x}_1 = (1, 1, \dots, 1) = \mathbf{1}_{k'} \quad \text{and } \mathbf{x}_2 = (1, 1, \dots, 1) = \mathbf{1}_{k'}.$$

Following steps in [7] for the decoding function we get that

$$\begin{aligned} g_{t, \Sigma}(f_{e_1}(\mathbf{x}_1), f_{e_2}(\mathbf{x}_2)) &= \hat{\mathbf{h}}_0(f_{t, \Sigma}(f_{e_1}(\mathbf{h}_0(\mathbf{x}_1)), f_{e_2}(\mathbf{h}_0(\mathbf{x}_2)))) \\ &= \hat{\mathbf{h}}_0(\mathbf{h}_0(\mathbf{x}_1) + \mathbf{h}_0(\mathbf{x}_2)) \\ &= \hat{\mathbf{h}}_0([\mathbf{1}_{k'} \quad \mathbf{0}_{t-k'}] + [\mathbf{1}_{k'} \quad \mathbf{0}_{t-k'}]) \\ &= \hat{\mathbf{h}}_0([\mathbf{2}_{k'} \quad \mathbf{0}_{t-k'}]) \end{aligned}$$

where $\mathbf{2}_{k'}$ is a vector of k' 2's.

Since $\hat{h}_0(2)$ is arbitrarily assigned, $\hat{\mathbf{h}}_0([\mathbf{2}_{k'} \quad \mathbf{0}_{t-k'}])$ need not equal $\mathbf{0}_{k'}$ which is the right value of $\mathbf{X}_1 + \mathbf{X}_2$. Thus the extension of the network code does not correctly evaluate the sum in $B^{k'}$.

The characteristics of a particular alphabet only affect the value of a function of the source random variables and not the random variables themselves. The extension of the network code from one alphabet to another works for the case of multiple unicast as the messages demanded by any terminal are a subset of all the messages observed in the network and not a function of them.