Network Coding: Does the Model Need Tuning?

April Rasala Lehman* Eric Lehman†

ABSTRACT

We consider the *general network information flow problem*, which was introduced by Ahlswede et. al[1]. We show a *periodicity effect*: for every integer $m \geq 2$, there exists an instance of the network information flow problem that admits a solution if and only if the alphabet size is a perfect m^{th} power. Building on this result, we construct an instance with $O(m)$ messages and $O(m)$ nodes that admits a solution if and only if the alphabet size is an enormous $2^{\exp\left(\Omega\left(m^{1/3}\right)\right)}$. In other words, if we regard each message as a length-k bit string, then k must be *exponential* in the size of the network. For this same instance, we show that if edge capacities are slightly increased, then there is a solution with a modest alphabet size of $O(2^m)$. In light of these results, we suggest that a more appropriate model would assume that the network operates at slightly under capacity.

1. INTRODUCTION

Network information theory considers the information carrying capacity of a network. Formally, in the *network information flow problem*, introduced by Ahlswede at. al [1], a network is represented by a directed acyclic multigraph containing sources (each with a set of available messages), intermediate nodes, and sinks (each demanding a set of messages). Messages are single symbols from an alphabet Σ of size q. An edge with capacity c can transmit c symbols from the alphabet. Information can be duplicated and encoded at internal nodes. A solution is a set of encoding functions for internal nodes and a decoding function for each sink that collectively allow the sinks to receive all their requested messages.

To give perspective on this model, we consider two other classic problems which have been used to model communication networks. First, in the framework of multi-commodity flow, a message is a commodity that must be shipped from a sender to a receiver. Flow must be conserved at internal nodes, and two commodities can only share an edge if the sum of their flows is at most the capacity of the edge [3]. In contrast, in the information flow problem we view a message as information. Thus, flow conservation is no longer an issue; an internal node receiving one copy of a message can transmit exact copies of this message on two or more outgoing links. Similarly, messages can "share" an edge in a complex way, since an edge can carry an arbitrary function of two or more messages.

Second, in contrast to traditional information theory, which has concentrated on the single channel case, network information theory combines questions of routing and coding together in one framework. In the single channel case, a collection of senders attempts to transmit some messages to a collection of receivers over a possibly noisy channel. However, an entire network cannot be viewed as a single channel, since there can exist many entry and exit points with distinct characteristics.

Therefore, many interesting questions about network information flow are not properly studied in either of the above settings. A cannonical problem is shown in Figure 1. The source has two single-bit messages, each edge can transmit one bit, and each receiver must get both bits. A solution is to transmit the XOR of the two bits across the middle edge so each receiver can reconstruct the two transmitted messages. Thus, the capacity of the network in Figure 1 is dependent on the computational power of the internal nodes.

Initial approaches to network coding used *linear* codes and focused on the *multicast* problem [1, 11, 9, 8]. In this early work, the alphabet Σ is a finite field $\mathbb{F}$, and each symbol departing a node is a linear combination of the symbols entering that node. For such an approach, the field $\mathbb{F}$ must be sufficiently large. In particular, $|\mathbb{F}| = O(\#\text{ sinks})$ is always sufficient [7] and $|\mathbb{F}| = \Omega(\sqrt{\#\text{ sinks}})$ is sometimes necessary [10, 13].

For non-multicast problems, the situation is more complex. Specifically, there are solvable instances of the network coding problem for which no linear solution exists, regardless of the choice of field $\mathbb{F}$ [10]. Medard et. al [14] conjectured that every solvable instance can be solved with a *vector-linear* code. In this case, the alphabet consists of all length-

*MIT Laboratory for Computer Science. Cambridge, MA 02139. arasala@theory.lcs.mit.edu

†MIT Laboratory for Computer Science. Cambridge, MA 02139. e_lehman@theory.lcs.mit.edu

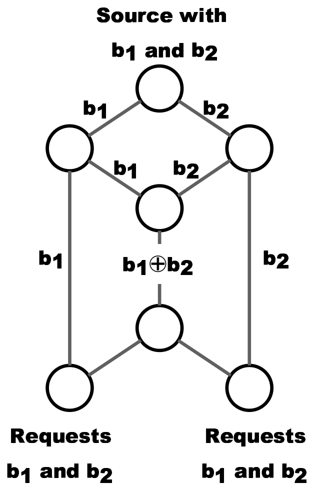

Figure 1: In this instance of the network information flow problem, all edges are directed downward and can transmit one bit. The goal is to transmit bits b_1 and b_2 from the source to both sinks.

k vectors over a finite field $\mathbb{F}$. Each component of a vector departing a node is a linear combination of all the components of all vectors entering that node. Recently, Doughtery, Freiling, and Zeger [5], constructed a solvable instance of the network information flow problem that does not admit a vector-linear solution.

In this paper, we consider how alphabet size scales with the size of the network. This problem is of interest because a large alphabet size may be associated with high latency, node storage requirements, and solution complexity. Previous results are few and somewhat surprising. Riis [15] showed that for every $k \geq 1$, there exist solvable instances of the network information flow problem which are not solvable with a vector-linear code using vectors of size k. However, Riis' networks do have vector-linear solutions for somewhat larger vector sizes. Dougherty, Freiling, and Zeger [4] have shown that every solvable instance with two single-bit messages admits a linear solution and that this does not hold for three or more single-bit messages. In addition, by relating the network information flow problem to orthogonal Latin squares, they created an instance of the multicast problem that has a, not necessarily linear, solution as long as the alphabet size is not 2 or 6. Thus, a problem solvable with a smaller alphabet (say, size 5) may not be solvable with larger alphabet (size 6).

Our Results

We show that network codes possess a curious property with regard to alphabet size. When using linear codes for multicast, one need only ensure that the underlying field $\mathbb{F}$ is sufficiently large for the problem at hand. If an instance of the multicast network information flow problem has a linear solution over one field, then it has a linear solution over every larger field. In contrast, we show that in the general setting, *there is no such thing as a "sufficiently large" alphabet.* In particular, for each integer $m \geq 2$, we exhibit an instance of the network information flow problem that admits a solution if and only if the alphabet size is a perfect m^{th} power. Thus, if $m = 3$, then the corresponding instance has a solution if the alphabet size is $2^3 = 8$, $3^3 = 27$, or $4^3 = 64$, but not if the alphabet size is $10,000$. Our construction has several implications:

- The power of network codes does not strictly increase with alphabet size, but rather increases as the size of the set of perfect roots of the alphabet size increases. Thus, an alphabet size of 2^6, which is a perfect square, a perfect cube, and a 6^{th} power is strictly better than a size of 2^2or 2^3, but an alphabet of size 2^7, which is only a 7^{th} power is not. In practice, one might be tempted to use an alphabet size of 2^{32} or 2^{64} so that a single alphabet symbol fits into a machine word. However, our construction suggests that these would actually be poor choices, since 32 and 64 have so few divisors.

- When linear coding is used to solve the multicast problem, an alphabet of size t suffices if there are t sinks. The situation with the general network information flow problem is dramatically different. By placing many of our constructions in parallel, we obtain an instance of the general network information flow problem with $O(n)$ nodes, including sinks, that requires an alphabet of size $2^{\exp(\Omega(n^{1/3}))}$. Thus, there exist network information flow problems that are solvable, but require extremely large alphabets. Naively, even describing the solution takes space exponential in the problem size.

- We show that our lower bound on the alphabet size does not hold if we slightly increase the capacity of the edges. In particular, the instance described above admits a vector-linear solution where messages are vectors of length n provided each edge can transmit a vector of length $n\left(1 + \frac{1}{n^{1/3}}\right)$.

In light of these results, we suggest that a better model for the study of network information flow problems would allow the network to operate at slightly under capacity, since this may avoid an exponential blowup in the solution complexity.

2. CONSTRUCTION

In this section we describe the construction of an instance of the network information flow problem that we denote I_k. In the next section, we prove that instance I_k admits a solution if and only if the alphabet size is a k^{th} power. The construction is shown for $k = 3$ in Figure 2.

There is a single source with $2k$ messages $M_1, M_2 \ldots M_{2k}$ and a single middle-layer node. There is an edge C of capacity 2 from the source to the middle layer node. There are $O(k^2)$ sinks. There is an edge of capacity 2 from the middle-layer node to each sink. One sink t^* requests all $2k$ messages and all other sinks request k messages. Let $S = \{M_1, M_2 \ldots M_k\}$ and $\overline{S} = \{M_{k+1}, M_{k+2} \ldots M_{2k}\}$. The sink t_S requests all messages in S, and the sink $t_{\overline{S}}$ requests all messages in $\overline{S}$. For all i and j such that $1 \leq i, j \leq k$ there is a sink t_{ij} that requests the k messages $(\overline{S} \cup \{M_i\}) - \{M_{k+j}\}$ and a complementary sink $t_{\overline{ij}}$ that requests the k messages $(S \cup \{M_{k+j}\}) - \{M_i\}$. From the source to t^* there is an

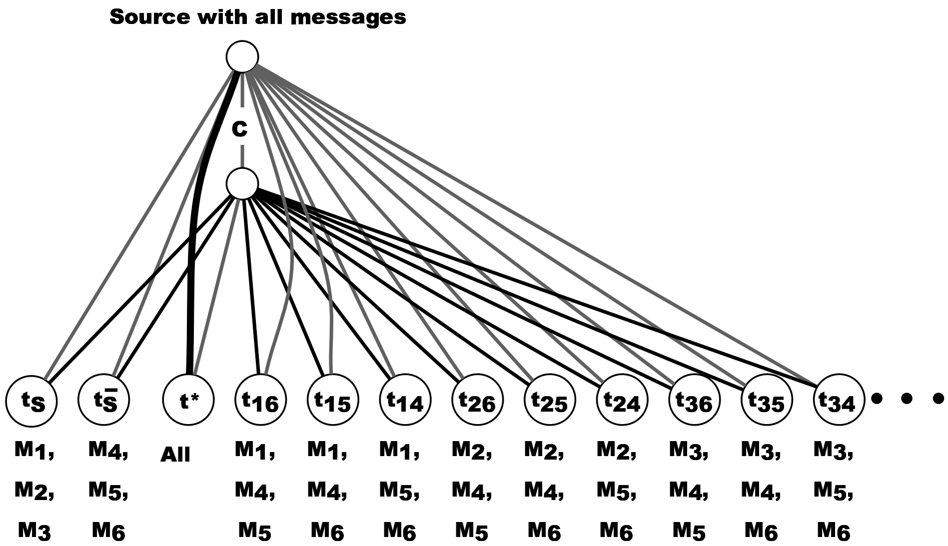

Figure 2: In this instance of the network information flow problem, all edges are directed downward. All edges have capacity 2, except for the thick, curved edge, which has capacity 4. There are six messages, $M_1, M_2, \ldots, M_6$. The top node is the only source and has every message. The bottom layer of nodes is the sinks, whose requests are listed below each node. The complementary sinks $t_{\overline{ij}}$ are not shown.

edge of capacity $2k-2$, and from the source to every other sink there is an edge of capacity $k-1$.

3. ANALYSIS

The analysis of the network I_k relies on understanding the function $F : \Sigma^{2k} \to \Sigma^2$ that describes the two symbols transmitted over edge C. We first prove a preliminary fact about the function F.

LEMMA 1. *The function $F : \Sigma^{2k} \to \Sigma^2$, which determines the symbol sent over edge C, is q^{2k-2}-to-1.*

PROOF. Sink t^* must recover all the messages from the symbol sent on edge C and the $2k-2$ other alphabet symbols it receives on the direct edge from the source. Let $g : \Sigma^{2k} \to \Sigma^{2k-2}$ be the function that determines the $2k-2$ symbols sent along the direct edge. If F is not q^{2k-2}-to-1, then F must map more than q^{2k-2} points in Σ^{2k} to some pair of symbols in Σ^2. Then g necessarily maps two of these points to the same value in Σ^{2k-2}. Thus, sink t^* receives identical symbols for two different sets of sent messages and can not distinguish them. □

We now consider restrictions placed on the function F by a sink requesting a set of k messages. In particular, we consider the set of assignments to the $2k$ messages that are mapped by F to the same value. For the remainder of this section we let $\beta \in \Sigma^2$ be a fixed value sent by F down edge C and $B \subseteq \Sigma^{2k}$ denote the subset of size q^{2k-2} that F maps to β. Thus, B is the set of assignments to messages such that the edge C carries the value β. Consider a subset of the $2k$ messages $A = \{M_{s_1}, M_{s_2} \ldots M_{s_r}\}$. The set of assignments to the messages in A such that there exists an assignment to the remaining messages on which F takes on the value β is the projection of each point in B onto the coordinates $s_1, s_2 \ldots s_r$. We denote this set as $\pi_{s_1,s_2\ldots s_r}(B)$ or equivalently $\pi_A(B)$. For example, if $A = \{M_1, M_2, M_3\}$

$$\pi_A(B) = \{(x_1, x_2, x_3) \mid (x_1, x_2, x_3, y_4, \ldots y_{2k}) \in B \text{ for some } (y_4, y_5 \ldots y_{2k})\}$$

LEMMA 2. *Let t be a sink requesting the set A of k messages. Then $|\pi_A(B)| = q^{k-1}$.*

PROOF. In addition to the point in Σ^2 sent to the middle-layer node, the sink t also receives $k-1$ symbols on a direct edge from the source. If the sink t receives the value $\beta \in \Sigma^2$ from the middle-layer node, then the assignment to the k messages in A must be according to one of the points in $\pi_A(B)$. Each point in $\pi_A(B)$ represents a different assignment to the messages in A. Therefore, the sink t must receive a different set of $k-1$ symbols along the direct edge from the source for each of the points in $\pi_A(B)$. Since there are only q^{k-1} different assignments to the $k-1$ symbols sent down the direct edge from the source, we must have $|\pi_A(B)| \le q^{k-1}$. Otherwise, the messages in A can not be uniquely determined from the information received at the sink.

By construction, there is a complementary sink $\overline{t}$ requesting the subset $\overline{A}$ consisting of the other k messages. By the same argument as above, $|\pi_{\overline{A}}(B)| \le q^{k-1}$. The number of different points in Σ^{2k} on which F takes on the value β is at most $|\pi_A(B)| \cdot |\pi_{\overline{A}}(B)|$. By Lemma 1, F is a q^{2k-2}-to-1 function. Therefore,

$$\begin{aligned} q^{2k-2} &\leq |\pi_A(B)| \cdot |\pi_{\overline{A}}(B)| \\ &\leq |\pi_A(B)| \cdot q^{k-1} \\ q^{k-1} &\leq |\pi_A(B)| \end{aligned}$$

Therefore, $|\pi_A(B)| = q^{k-1}$. $\square$

We learn more about the structure of the set of points B on which F takes on the value β by applying the above lemma to sink t requesting the set A of k messages and its complementary sink $\overline{t}$ requesting the other k messages.

Lemma 3. *Let t be a sink requesting the set A of k messages, and let $\overline{t}$ be the sink requesting the other k messages $\overline{A}$. Then*

$$B = \pi_A(B) \times \pi_{\overline{A}}(B)$$

Proof. Consider an assignment $z \in B$ to the messages. Suppose z assigns the k messages in A according to an assignment z_A and assigns the messages in $\overline{A}$ according to an assignment $z_{\overline{A}}$. Then, $z_A \in \pi_A(B)$ and $z_{\overline{A}} \in \pi_{\overline{A}}(B)$ by the definition of projection. Therefore, $B \subseteq \pi_A(B) \times \pi_{\overline{A}}(B)$. By Lemma 2, $|\pi_A(B) \times \pi_{\overline{A}}(B)| = |\pi_A(B)| \cdot |\pi_{\overline{A}}(B)| = q^{2k-2}$. Since $|B| = q^{2k-2}$ by Lemma 1, B has the same size as the set containing it. Therefore, $B = \pi_A(B) \times \pi_{\overline{A}}(B)$. $\square$

The next lemma shows that for at least one sink, the projection of the set B onto the messages requested by that sink is "large". The proof makes use of the discrete Loomis-Whitney inequality relating the size of a set to the product of the sizes of projections of the set [12, 2, 17, 16] . Roughly, the discrete Loomis-Whitney inequality generalizes the intuition that a massive statue must look big from the front, the side, or the top; that is, a big region must have some big projection.

Theorem 4 (Discrete Loomis-Whitney Inequality). *Let $Q \subseteq \Sigma^h$ and $r \leq h$,*

$$|Q| \leq \prod_{1 \leq s_1 < \ldots < s_r \leq h} |\pi_{s_1,\ldots,s_r}(Q)|^{hr^{-1}\binom{h}{r}^{-1}}$$

Lemma 5. *There exists a set of k messages $A = (\overline{S} \cup \{M_i\}) - \{M_{k+j}\}$ such that $|\pi_A(B)| \geq \left\lceil q^{\frac{k-1}{k}} \right\rceil \left\lceil q^{\frac{(k-1)^2}{k}} \right\rceil$.*

Proof. We prove this in three steps. First we show that there exists a message $M_i \in S$ such that $|\pi_i(B)| \geq \left\lceil q^{\frac{k-1}{k}} \right\rceil$. Then we show that there exists a set of $k-1$ messages, $\overline{S} - \{M_{k+j}\} = \{M_{s_1}, M_{s_2} \ldots M_{s_{k-1}}\}$ such that $|\pi_{s_1,s_2\ldots s_{k-1}}(B)| \geq \left\lceil q^{\frac{(k-1)^2}{k}} \right\rceil$. To finish the proof, we use Lemma 3 to show that

$$|\pi_{i,s_1,s_2\ldots s_{k-1}}(B)| \geq |\pi_i(B)| \cdot |\pi_{s_1,s_2\ldots s_{k-1}}(B)|$$

Using the Loomis-Whitney Inequality for $r = 1$,

$$\begin{aligned} \prod_{1 \leq i \leq k} |\pi_i(B)| &\geq |\pi_S(B)| \\ &= q^{k-1} \end{aligned}$$

Therefore, there exists at least one message $M_i \in S$ for which $|\pi_i(B)| \geq \left\lceil q^{\frac{k-1}{k}} \right\rceil$.

Similarly, by the Loomis-Whitney Inequality for $r = k-1$,

$$\begin{aligned} \prod_{k+1 \leq s_1 < s_2 < \ldots < s_{k-1} \leq 2k} |\pi_{s_1,s_2\ldots s_{k-1}}(B)|^{\frac{1}{k-1}} &\geq |\pi_{\overline{S}}(B)| \\ &= q^{k-1} \\ \prod_{k+1 \leq s_1 < s_2 < \ldots < s_{k-1} \leq 2k} |\pi_{s_1,s_2\ldots s_{k-1}}(B)| &\geq q^{(k-1)^2} \end{aligned}$$

Since there are k terms in the product on the left, there exist $k-1$ messages $\{M_{s_1}, M_{s_2} \ldots M_{s_{k-1}}\} \subseteq \overline{S}$ such that $|\pi_{s_1,s_2\ldots s_{k-1}}(B)| \geq \left\lceil q^{\frac{(k-1)^2}{k}} \right\rceil$.

For each $x_i \in \pi_i(B)$, there exists $x \in \pi_S(B)$ that assigns message M_i the value x_i. Similarly, for each $(y_{s_1}, \ldots y_{s_{k-1}}) \in \pi_{s_1,s_2\ldots s_{k-1}}(B)$ there exists $y \in \pi_{\overline{S}}(B)$ that corresponds to assigning the messages $\{M_{s_1}, M_{s_2}, \ldots M_{s_{k-1}}\}$ the values $(y_{s_1}, y_{s_2} \ldots y_{s_{k-1}})$. By Lemma 3, $B = \pi_S(B) \times \pi_{\overline{S}}(B)$, and so there is a point in B that assigns the value x_i to message M_i and the values $y_{s_1}, y_{s_2}, \ldots, y_{s_{k-1}}$ to messages $M_{s_1}, M_{s_2}, \ldots M_{s_{k-1}}$. Therefore,

$$\begin{aligned} |\pi_{i,s_1,s_2\ldots s_{k-1}}(B)| &\geq |\pi_i(B)| \cdot |\pi_{s_1,s_2\ldots s_{k-1}}(B)| \\ &\geq \left\lceil q^{\frac{k-1}{k}} \right\rceil \left\lceil q^{\frac{(k-1)^2}{k}} \right\rceil \end{aligned}$$

$\square$

Theorem 6. *There exists a solution to network I_k if and only if the alphabet size q is a perfect k^{th} power.*

Proof. There are two steps. We first show how to construct a solution with an alphabet of size $q = \ell^k$ for any $\ell \geq 2$. Then we show that the network only admits a solution if the alphabet size is a perfect k^{th} power.

Let Γ be a set of size ℓ. We regard each message as a length-k vector of symbols drawn from Γ. Recall that edge C has capacity 2. Therefore, we can send $2k$ symbols from Γ across edge C. We use these $2k$ symbols to transmit the first coordinate of each of the $2k$ messages. The sink t^*, which requests all $2k$ messages, must receive $2k$ length-k vectors.

Via edge C, it receives the first coordinate of each of these $2k$ vectors. Along the direct edge of capacity $2k-2$ from the source to t^*, we send the remaining $k-1$ coordinates of each of the $2k$ messages. Now consider a sink t requesting a subset A of k messages. Sink t receives the first coordinate of each message in A from edge C. The remaining $k-1$ coordinates of each of the messages in A can be transmitted across the direct edge of capacity $k-1$ from the source to t. Thus, each sink receives every coordinate of the messages it requests. The alphabet size is $q=\ell^k$.

Next, we show that the alphabet size must be a k^{th} power. Lemma 5 says that there exists a set $A=(\overline{S}\cup\{M_i\})-\{M_{k+j}\}$ of k messages with $|\pi_A(B)|\geq\left\lceil q^{\frac{k-1}{k}}\right\rceil\left\lceil q^{\frac{(k-1)^2}{k}}\right\rceil$. On the other hand, since there is a sink t_{ij} for every $1\leq i,j\leq k$ requesting the set of messages $(\overline{S}\cup\{M_i\})-\{M_{k+j}\}$, we must have $|\pi_A(B)|=q^{k-1}$. These two relationships can hold simultaneously only if q is a k^{th} power. □

4. A LOWER BOUND ON ALPHABET SIZE

We now construct an instance, J_n, of the network information flow problem with $\Theta(n)$ nodes that admits a solution if and only if the alphabet size is $q=2^{\exp(\Omega(n^{1/3}))}$. The construction is as follows. For each prime number $p\leq n^{1/3}$, we take the instance I_p of the preceding construction, which forces the alphabet size to be a p^{th} power. We place all of these constructions in parallel in order to create instance J_n.

COROLLARY 7. *Instance J_n with $\Theta(n)$ nodes admits a solution if only if the alphabet size is $2^{\exp\left(\Omega\left(n^{1/3}\right)\right)}$.*

PROOF. The number of nodes in J_n is at most:

$$\sum_{i=1}^{n^{1/3}} 2i^2+1 \quad = \quad \Theta(n)$$

Instance I_p, is solvable if and only if the alphabet size if a p^{th} power. Thus, instance J_n is solvable if and only if the alphabet size is a p^{th} power for every prime p less than $n^{1/3}$. The product of primes less than x is $e^{(1+o(1))x}$ (see [6]). Therefore, the minimum alphabet size is $q=2^{\exp\left(\Omega\left(n^{1/3}\right)\right)}$. □

The fact that J_n is made up of a collection of disjoint networks is not critical to the proof. In fact, one can add some sinks that join the networks and force some degree of coding. More generally, one can imagine problems in which the various instances requiring different vector sizes are embedded in a larger network and may not be easily detectable.

While the instance J_n requires a very large alphabet, not much storage is actually needed at the nodes. Also, the solution presented in Section 3 can be described concisely without resorting to a particularly powerful description language. An interesting question is whether other instances admit only solutions with not only enormous alphabets, but also comparable storage requirements and description complexity.

5. OPERATING BELOW CAPACITY

In this section we consider the effect of allowing the network to operate at slightly below full capacity. We model this using vector linear codes in which the edges are allowed to transmit vectors that are longer than the message vectors. In particular, suppose that each message is a length-k vector, but vectors transmitted over edges have length $(1+\epsilon)k$. We show that for vanishingly small ϵ, the network J_n in Corollary 7 admits a solution over any field with a vector length linear in the size of the network. Using a constant-size field, this corresponds to a vector-linear solution with an alphabet that is only exponential (instead of doubly exponential) in the size of the network.

THEOREM 8. *There exists a vector linear solution to the network J_n on $\Theta(n)$ nodes with message-vector length n and edge-vector length $\left(1+n^{-2/3}\right)n$.*

PROOF. Recall that J_n is constructed by placing instances $I_2, I_3, I_5 \ldots I_s$ in parallel, where s is the largest prime less than $n^{1/3}$. Consider prime p and the subnetwork I_p in J_n. In our solution, we send $\left\lceil\frac{n}{p}\right\rceil$ unencoded bits of each message across edge C in I_p. A sink requesting p messages must receive a total of pn message bits. A total of $p\left\lceil\frac{n}{p}\right\rceil$ of these message bits are sent via edge C. The remaining at most

$$pn-p\cdot\left\lceil\frac{n}{p}\right\rceil\leq(p-1)n$$

message bits can be transmitted along the direct edge from the source to the sink. Similarly the sink t^* requesting all the messages receives $2p\cdot\left\lceil\frac{n}{p}\right\rceil$ message bits from edge C and can receive the other at most $(2p-2)n$ message bits via the direct edge with capacity $2p-2$.

We can upper bound the length k' of the two vectors transmitted across edge C as follows. For each of the $2p$ messages, we transmit $\left\lceil\frac{n}{p}\right\rceil$ bits on edge C. Therefore, we have:

$$\begin{aligned} 2k' &= 2p\left\lceil\frac{n}{p}\right\rceil \\ &\leq 2n+2p \\ &= 2n\left(1+\frac{p}{n}\right) \\ &\leq 2n\left(1+n^{-2/3}\right) \end{aligned}$$

Therefore the length of each vector sent across edge C is at most $\left(1+n^{-2/3}\right)n$. We make no use of the extra capacity along any other edge. □

6. DISCUSSION

Our results suggest that using a network at full capacity may be undesirable; even if a solution exists, an enormous alphabet may be required. On the other hand, slightly increasing the network capacity eliminates this problem, at least for the instance we propose. (An interesting open question is whether *every* solvable instance admits a solution with moderate alphabet size, provided that the network operates just below capacity.) This points toward an exploration of network coding in a model where the network has a small amount of surplus of capacity.

Acknowledgments

We wish to thank Keith Ball and Paul Goodey for pointing us to the discrete Loomis-Whitney inequality. We would also like to thank Ken Zeger, Madhu Sudan and Nick Harvey for their comments.

7. REFERENCES

[1] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[2] Yu. D. Burago and V. A. Zalgaller. Geometric inequalities. In *Grundlehren der mathematischen Wiseenchaften*. Spinger-Verlag, 1988. Number 285.

[3] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2001.

[4] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks, December 2003.

[5] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow, Febuary 2004.

[6] G.H. Hardy and E.M. Wright. *An Introduction to the Theory of Numbers, 5th Edition*. Clarendon Press, Oxford, England, 1979.

[7] Sidharth Jaggi, Peter Sanders, Philip A. Chou, Michelle Effros, Sebastian Egner, Kamal Jain, and Ludo Tolhuizen. Polynomial time algorithms for multicast network code construction, July 2003. Submitted to IEEE Transactions on Information Theory.

[8] Ralf Koetter and Muriel Medard. Beyond routing: An algebraic approach to network coding. In *INFOCOM*, 2002.

[9] Ralf Koetter and Muriel Medard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, October 2003.

[10] April Rasala Lehman and Eric Lehman. Complexity classification of network information flow problems. In *14th Annual Symposium on Discrete Algorithms*, January 2004.

[11] Shuo-Yen Robert Li, Raymond Yeung, and Nig Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49:371 – 381, 2003.

[12] L. H. Loomis and H. Whitney. An inequality related to the isoperimetric inequality. *Bulletin of the American Mathematial Society*, 1949.

[13] D. Ron M. Feder and A. Tavory. Bounds on linear codes for network multicast. *Electronic Colloquim on Computational Complexity (ECCC)*, 10(033), 2003.

[14] Muriel Medard, Michelle Effros, Tracy Ho, and David Karger. On coding for non-multicast networks. In *41st Annual Allerton Conference on Communication Control and Computing*, 2003.

[15] Soren Riis. Linear versus non-linear boolean functions in network flow. In *Conference on Information Sciences and Systems (CISS)*, March 2003.

[16] Alexandre Tiskin. A generalization of the Cauchy and Loomis-Whitney inequalities. http://www.dcs.warwick.ac.uk/ tiskin/pub/.

[17] Vorlesungen ūber Inhalt. Oberflāche and isoperimetrie. In *Grundlehren der mathematischen Wissenchaften*. Springer-Verlag, 1957. Number 93.