The Encoding Complexity of Network Coding

Michael Langberg Alexander Sprintson Jehoshua Bruck
California Institute of Technology
Email: {mikel,spalex,bruck}@caltech.edu

***Abstract*— In the multicast network coding problem, a source s needs to deliver h packets to a set of k terminals over an underlying network G. The nodes of the coding network can be broadly categorized into two groups. The first group incudes *encoding* nodes, i.e., nodes that generate new packets by combining data received from two or more incoming links. The second group includes *forwarding* nodes that can only duplicate and forward the incoming packets. Encoding nodes are, in general, more expensive due to the need to equip them with encoding capabilities. In addition, encoding nodes incur delay and increase the overall complexity of the network.**

Accordingly, in this paper we study the design of multicast coding networks with a limited number of encoding nodes. We prove that in an acyclic coding network, the number of encoding nodes required to achieve the capacity of the network is bounded by h^3k^2. Namely, we present (efficiently constructible) network codes that achieve capacity in which the total number of encoding nodes is independent of the size of the network and is bounded by h^3k^2. We show that the number of encoding nodes may depend both on h and k as we present acyclic instances of the multicast network coding problem in which $\Omega(h^2k)$ encoding nodes are required.

In the general case of coding networks with cycles, we show that the number of encoding nodes is limited by the size of *the feedback link set*, i.e., the minimum number of links that must be removed from the network in order to eliminate cycles. Specifically, we prove that the number of encoding nodes is bounded by $(2B+1)h^3k^2$, where B is the minimum size of the feedback link set. Finally, we observe that determining or even crudely approximating the minimum number of encoding nodes required to achieve the capacity for a given instance of the network coding problem is $\mathcal{NP}$-hard.

I. Introduction

The goal of communication networks is to transfer information between source and destination nodes. Accordingly, the fundamental question that arises in network design is how to increase the amount of information transferred by the network. Recently, it has been shown that the ability of the network to transfer information can be significantly improved by employing the novel technique of *network coding* [1]–[3]. The idea is to allow the intermediate network nodes to combine data received over different incoming links. Nodes with coding capabilities are referred to as *encoding* nodes, in contrast to *forwarding* nodes that can only forward and duplicate incoming packets. The network coding approach extends traditional routing schemes, which include only forwarding nodes.

The concept of network coding was introduced in a seminal paper by Ahlswede et. al. [1] and immediately attracted a significant amount of attention from the research community. A large body of research focused on the multicast network coding problem where a source s needs to deliver h packets to a set of k terminals T over an underlying network G. It was shown in [1] and [2] that the capacity of the network, i.e., the maximum number of packets that can be sent between s and T, is characterized by the size of the minimum *cut*[1] that separates the source s and a terminal $t \in T$. Namely, a source s can transmit at capacity h to a set of terminals if and only if the size of the minimum cut separating s and any one of the terminals is at least h. This capacity was shown to be achievable in [2] by using a linear network code, i.e., a code in which each packet sent over the network is a linear combination of the original packets. In a subsequent work, Koetter and Médard [3] developed an algebraic framework for network coding and investigated linear network codes for directed graphs with cycles. This framework was used by Ho et al. [4] to show that linear network codes can be efficiently constructed by employing a randomized algorithm. Jaggi et al. [5] proposed a deterministic polynomial-time algorithm for finding a feasible network code for a given multicast network.

In this study we focus on minimizing the total number of encoding nodes in multicast coding networks. More specifically, given a communication network G, a source node s, a set of terminals T, and a required number of packets h, our goal is to find a feasible network code with as few encoding nodes as possible. This problem is important for both theoretical and practical reasons. First, encoding nodes in a network are, in general, more expensive than forwarding nodes, mostly because of the need to equip them with coding capabilities. In addition, encoding nodes may incur delay and increase the overall complexity of the network.

This work was supported in part by the Caltech Lee Center for Advanced Networking and by NSF grants ANI-0322475 and CCF-0346991.

[1] A cut (V_1, V_2) in graph $G(V,E)$ is a partition of G into two subsets V_1 and $V_2 = V \setminus V_1$. The size of the cut is equal to the total capacity of links that leave a node in V_1 and enter a node in V_2. We say that a cut (V_1, V_2) separates node s and t if $s \in V_1$ and $t \in V_2$.

Contribution

The contribution of our paper can be summarized as follows. We prove that to enable transmission at rate h from a source s to k terminals over an acyclic graph, one can efficiently construct network codes in which the number of encoding nodes is independent of the size of the underlying graph G and depends only on h and k. Our construction procedure is very simple and involves three steps: (1) Transform the original network into one which is minimal with respect to link removal and in which the degree of each internal node is at most three; (2) Find a feasible network code for the reduced network; (3) Reconstruct a network code for the original network. We show that such a procedure yields codes with only h^3k^2 encoding nodes. We also show that in the worst case the number of encoding nodes depends on both h and k. To that end, we present, for any values of h and k, a coding network that requires $\Omega(h^2k)$ encoding nodes.

Next, we consider the general case of coding networks with cycles. We show that in such networks, the number of encoding nodes required to enable transmission at capacity h from a source s to k terminals depends on the size of *the feedback link set* of the network, i.e., the minimum number of links that must be removed from the network in order to eliminate cycles. Specifically, we prove that the number of encoding nodes needed is bounded by $(2B+1)h^3k^2$, where B is the minimum size of the feedback link set. We also present a lower bound of $\Omega(hB)$ on the number of encoding nodes in a network with cycles.

Finally, we consider the problem of finding a network code that enables transmission at capacity h from a source s to k terminals with a *minimum* number of encoding nodes. We observe that determining, or even crudely approximating, the minimum number of encoding nodes needed to achieve capacity is $\mathcal{NP}$-hard.

Encoding links

A more accurate estimation of the total amount of computation performed by a coding network can be obtained by counting *encoding links*, rather than encoding nodes. A link (v,u), $v \notin s$, is referred to as an encoding link if each packet sent on this link is a combination of two or more packets received through the incoming links of v. Indeed, as the output degrees of nodes in G may vary, each encoding node might have different computation load. In addition, only some of the outgoing links of a node v can be encoding, while other outgoing links of v merely forward packets that arrive on v. Accordingly, we can consider the problem of finding a feasible network code that minimizes the total number of encoding links. It turns out that all upper and lower bounds on the minimum number of encoding nodes we present, as well as our inapproximability results, carry over to the case in which we want to minimize the number of encoding links. This follows from the fact that all our results are derived by studying networks in which internal nodes are of total degree three. In such networks, the number of encoding links is equal to the number of encoding nodes. For the remainder of this paper we state our results in terms of encoding nodes.

Related work

The problem of minimizing the number of encoding nodes in a network code is partially addressed in the works of Fragouli et al. [6], [7], and Tavory et al. [8]. The works of Fragouli et al. study the special case in which the given network is acyclic and one is required to transmit two packets from the source to a set of terminals of size k. For this specific case (i.e., $h = 2$) they show that the required number of encoding nodes is bounded by k. The proof techniques used in [6] and [7] rely on a certain combinatorial decomposition of the underlying network and seem difficult to generalize for the case in which the number of packets h is larger than two.

The problem of minimizing the number of encoding nodes in a network code is also studied by Tavory et al. [8]. They obtain partial results of nature similar to those of [6] and [7], mentioned above. Namely, they are able to prove, for the case $h = 2$, that the number of required encoding nodes is independent of the size of the underlying graph G. For general values of h, [8] presents several observations which lead to the conjecture that the number of encoding nodes needed to enable transmission at capacity h from a source s to k terminals over an acyclic graph, is independent of the size of the underlying graph. In our study we prove this conjecture.

Finally, encoding vs. forwarding nodes in the solution of network coding problems was also studied by Wu et al. [9]. Wu et al. do not consider the amount of encoding nodes in a given network code. Rather, they show the existence (and efficient construction) of network codes in which only nodes which are not directly connected to a terminal perform encoding. The results in [9] do not imply bounds on the number of encoding nodes needed in communicating over a network.

Organization

The rest of the paper is organized as follows. In Section II, we define the model of communication and state our results in detail. In Section III, we define the notion of a *simple* network and describe our algorithm for finding network codes with a bounded number of encoding nodes.

Due to space limitations, proofs and some technical details are omitted from this version, and can be found in [10].

II. Model

The communication network is represented by a directed graph $G = (V,E)$ where V is the set of nodes in G and E is the set of links. We assume that each link $e \in E$ can transmit one packet per time unit. In order to model links whose capacity is higher than one unit, G may include multiple parallel links. An instance $\mathbb{N}(G, s, T, h)$ of the network coding problem is a 4-tuple that includes the graph G, a source node $s \in V$, a set of terminals T, and the number of packets h that must be transmitted from the source node s to every terminal $t \in T$. We assume that each packet is a symbol of some alphabet Σ.

Definition 1 (Network code $\mathbb{F}(\mathbb{N})$): A network code for $\mathbb{N}(G, s, T, h)$ is defined by functions $\mathbb{F}(\mathbb{N}) = \{f_e \mid e \in E\}$. For links $e = (s, u)$ leaving the source, $f_e : \Sigma^h \to \Sigma$. For other links $e = (v, u)$, $f_e : \Sigma^{d_{in}(v)} \to \Sigma$. Here, $d_{in}(v)$ is the in-degree of node v.

The function $f_{(v,u)}$ specifies the packet transmitted on link (v, u) for any possible combination of packets transmitted on the incoming links of v. For links leaving the source, f_e takes as input the h packets available at a source.

Definition 2 (Encoding and forwarding links and nodes): For a network code $\mathbb{F}(\mathbb{N})$, e is referred to as an *encoding link*, if it has a corresponding function f_e that depends on two variables or more. If f_e depends on a single variable, we refer to e as a *forwarding link*. We say that a node v, $v \neq s$, is an *encoding node* if at least one of its outgoing links (v, u) is encoding. If all outgoing links of a node v are forwarding, the node is referred to as a *forwarding node*.

Note, that there may be links e for which the function f_e depends on a single variable, but $f_e(x) \neq x$. We refer to such links as forwarding nevertheless, and do not count them as encoding links. It is not hard to verify that in the case that $\mathbb{F}(\mathbb{N})$ includes links with corresponding functions f_e that depend on a single variable but are not the identity function, one can construct a new network code $\mathbb{F}'(\mathbb{N})$ without such functions such that the number of encoding links in $\mathbb{F}'(\mathbb{N})$ and $\mathbb{F}(\mathbb{N})$ are equal.

The capacity of a multicast coding network is determined by the minimum size of a cut that separates the source s and any terminal $t \in T$ [2]. An instance $\mathbb{N}(G, s, T, h)$ of the network coding problem is said to be *feasible* if and only if the size of each such cut is at least h. Let $\mathbb{N}(G, s, T, h)$ be a feasible network. A network code $\{f_{(v,u)} \mid (v, u) \in E\}$ for $\mathbb{N}$ is said to be feasible if it *allows communication at rate h* between s and each terminal $t \in T$. For acyclic networks, a network code is said to allow communication at rate h if each terminal $t \in T$ can compute the original h packets available at the source from the packets received via its incoming links. To define the notion of rate for cyclic networks, one must take into consideration multiple rounds of transmission (in which h packets are sent from the source s over the network in each round), and require that over time each terminal $t \in T$ can compute the original packets available at the source from the packets received via its incoming links. In both the cyclic and acyclic case, if $\mathbb{N}(G, s, T, h)$ is feasible then there exists a feasible network code for $\mathbb{N}$ [2].

A. Statement of results

As mentioned in the Introduction, our goal is to find feasible network codes with a minimum number of encoding nodes. For a given instance $\mathbb{N}(G, s, T, h)$ of the network coding problem, we denote by $Opt(\mathbb{N})$ the minimum number of encoding nodes in any feasible network code for $\mathbb{N}(G, s, T, h)$.

We show that computing $Opt(\mathbb{N})$ for a given instance $\mathbb{N}(G, s, T, h)$ of a network coding problem is an $\mathcal{NP}$-hard problem. Furthermore, it is $\mathcal{NP}$-hard to approximate $Opt(\mathbb{N})$ within any multiplicative factor or within an additive factor significantly less than $|V|$. This result follows from the fact that it is $\mathcal{NP}$-hard to distinguish between instances $\mathbb{N}$ in which $Opt(\mathbb{N}) = 0$ and instances in which $Opt(\mathbb{N}) > 0$.

Theorem 3: Let $\varepsilon > 0$ be any constant. Let $\mathbb{N}(G, s, T, h)$ be an instance of the multicast network coding problem in which the underlying graph has $|V|$ nodes. Approximating the value of $Opt(\mathbb{N})$ within any multiplicative factor or within an additive factor of $|V|^{1-\varepsilon}$ is $\mathcal{NP}$-hard.

Although the problem of finding the exact or approximate value of $Opt(\mathbb{N})$ is $\mathcal{NP}$-hard, it is possible to establish upper bounds on $Opt(\mathbb{N})$ that hold for any instance $\mathbb{N}(G, s, T, h)$ of the multicast network coding problem. The main contribution of our paper is an upper bound on $Opt(\mathbb{N})$ which is independent of the size of the network and depends on h and $|T| = k$ only. Specifically, we show that $Opt(\mathbb{N}) \leq h^3k^2$ for any acyclic coding network $\mathbb{N}$ that delivers h packets to $k = |T|$ terminals. Our bound is constructive, i.e., for any feasible instance $\mathbb{N}(G, s, T, h)$ we present an *efficient* procedure that constructs a network code with at most h^3k^2 encoding nodes. In what follows, an algorithm is said to be efficient if its running time is polynomial in the size of the underlying graph G.

Theorem 4 (Upper bound, acyclic networks): Let G be an acyclic graph and let $\mathbb{N}(G, s, T, h)$ be a feasible instance of the multicast network coding problem. Then, one can efficiently find a feasible network code for $\mathbb{N}$ with at most h^3k^2 encoding nodes, i.e., $Opt(\mathbb{N}) \leq h^3k^2$, where $k = |T|$.

Theorem 5 (Lower bound, acyclic networks): Let r_1 and r_2 be arbitrary integers. Then, there exist instances $\mathbb{N}(G, s, T, h)$ of the network coding problem such that $h \geq r_1$, $|T| = k \geq r_2$, the underlying graph G is acyclic, and $Opt(\mathbb{N}) \geq \Omega(h^2k)$.

Finally, we establish upper and lower bounds on the number of encoding nodes in the general setting of communication networks with cycles. We show that the value of $Opt(\mathbb{N})$ in a cyclic network $\mathbb{N}$ depends on the size of the *minimum feedback link set*.

Definition 6 (Minimum feedback link set [11]): Let $G(V, E)$ be a directed graph. A subset $\hat{E} \subseteq E$ is referred to as a *feedback link set* if the graph G' formed from G by removing all links in $\hat{E}$ is acyclic. A feedback link set of minimum size is referred to as the *minimum feedback link* *set*. Given a network $\mathbb{N}(G, s, T, h)$, we denote by $B(\mathbb{N})$ the minimum feedback link set of its underlying graph G.

Theorem 7 (Upper bound, cyclic networks): Let $\mathbb{N}(G, s, T, h)$ be an instance of the multicast network coding problem. Then, one can efficiently find a feasible network code for $\mathbb{N}$ with at most $(2B(\mathbb{N})+1)h^3k^2$ encoding nodes, i.e., $Opt(\mathbb{N}) \leq (2B(\mathbb{N})+1)h^3k^2$, where $k = |T|$.

Theorem 8 (Lower bound, cyclic networks): Let r_1 and r_2 be arbitrary integers. Then (a) there exists instances $\mathbb{N}(G, s, T, h)$ of the multicast network coding problem such that $B(\mathbb{N}) \geq r_1$, $h \geq r_2$ and $Opt(\mathbb{N}) \geq \Omega(B(\mathbb{N})h)$; (b) there exist instances $\mathbb{N}(G, s, T, h)$ of the multicast network coding problem such that $|V| \geq r_1$, $h = |T| = 2$, and $Opt(\mathbb{N}) = \frac{|V|-5}{2}$ (here V is the set of nodes in G).

A couple of remarks are in place. First, note that Theorem 7 generalizes Theorem 4, as for acyclic networks the minimum feedback link set is of size 0. Second, note that Theorem 8 establishes two lower bounds. The first complements the upper bound of Theorem 7, while the second shows that in the case of cyclic networks the value of $Opt(\mathbb{N})$ is not necessarily independent of the size of the network and may depend linearly on the number of nodes in G.

III. "Simple" coding networks

Let $\mathbb{N}(G, s, T, h)$ be a feasible instance to the network coding problem. In order to establish a constructive upper bound on the minimal number of encoding nodes $Opt(\mathbb{N})$ of $\mathbb{N}$ we consider a special family of feasible networks, referred to as *simple networks*. In what follows we define simple coding networks, and show that finding network codes with few encoding nodes for this family of restricted networks suffices to prove Theorems 4 and 7. We start by defining feasible instances which are minimal with respect to link removal.

Definition 9 (Minimal Instance): A feasible instance of the network coding problem $\mathbb{N}(G, s, T, h)$ is said to be *minimal with respect to link removal* if any instance $\hat{\mathbb{N}}(\hat{G}, s, T, h)$ formed from G by deleting a link e from G is no longer feasible.

Definition 10 (Simple instance): Let $\mathbb{N}(G, s, T, h)$ be an instance of the network coding problem. $\mathbb{N}(G, s, T, h)$ is said to be simple if and only if (a) $\mathbb{N}$ is feasible; (b) $\mathbb{N}$ is minimal with respect to link removal; (c) the total degree of each node in G is at most 3 (excluding the source and terminal nodes); and (d) the terminal nodes T have no outgoing links.

We now present our reduction between general and simple networks.

A. Reduction to simple networks

Let $\mathbb{N}(G, s, T, h)$ be a feasible instance of the network coding problem. We construct a simple instance $\hat{\mathbb{N}}(\hat{G}, s, \hat{T}, h)$ corresponding to $\mathbb{N}$. The simple instance $\hat{\mathbb{N}}$ we construct corresponds to $\mathbb{N}$ in the sense that any feasible network code for $\hat{\mathbb{N}}$ yields a network code for $\mathbb{N}$ which includes the same or a smaller amount of encoding nodes. Our construction is computationally efficient and includes the three following steps.

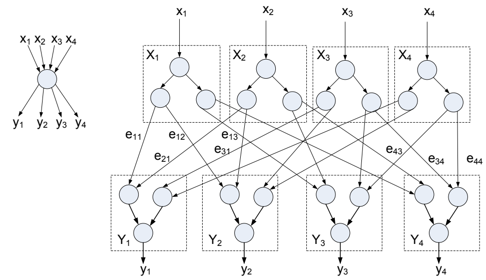

Fig. 1. Substituting a node v by a gadget Γ_v.

Step1: Replacing terminals. Let $T_1 \subseteq T$ be the set of terminals whose out-degree is not zero. For each terminal $t \in T_1$ we replace t by adding a new node t' to G and connecting t and t' by h parallel links. We denote the resulting set of terminals by $\hat{T}$, the resulting graph by G_1, and the resulting coding network by $\mathbb{N}_1(G_1, s, \hat{T}, h)$.

Step 2: Reducing degrees. Let G_2 be the graph formed from G_1 by replacing each node $v \in G_1$, $v \neq s$, $v \notin T_1$ whose degree is more than 3 by a subgraph Γ_v, constructed as follows. Let $\{x_i \mid i = 1, \ldots, d_{in}(v)\}$ and $\{y_i \mid i = 1, \ldots, d_{out}(v)\}$ be the incoming and outgoing links of v, respectively, where $d_{in}(v)$ and $d_{out}(v)$ are the in- and out-degrees of v. For each incoming link x_i, we construct a binary tree X_i that has a single incoming link x_i and $d_{out}(v)$ outgoing links $e_{i1}, \ldots, e_{id_{out}(v)}$ (with one or two links leaving each leaf). Similarly, for each outgoing link y_j we construct an inverted binary tree Y_j that has a single outgoing link y_j and $d_{in}(v)$ incoming links $e_{1j}, \ldots, e_{d_{in}(v)j}$ (again, with one or two links entering each leaf). Fig. 1 demonstrates the construction of the subgraph Γ_v for a node v with $d_{in}(v) = d_{out}(v) = 4$. Note that for any two links x_i and y_j there is a path in Γ_v that connects x_i and y_j. The resulting coding network is denoted by $\mathbb{N}_2(G_2, s, \hat{T}, h)$.

Step 3: Removing links. Let $\hat{G}$ be *any* subgraph of G_2 such that $\hat{\mathbb{N}}(\hat{G}, s, \hat{T}, h)$ is minimal with respect to link removal. The graph G_2 can be efficiently computed by employing the following greedy approach. For each link $e \in G_2$, in an arbitrary order, we check whether removal of e from G_2 would result in a violation of the min-cut condition. The min-cut condition can be easily checked by finding h link-disjoint paths between s and each terminal $t \in \hat{T}$ (via max-flow techniques, e.g., [12]). All links whose removal does not result in a violation of the min-cut condition are removed from G_2. The resulting coding network, denoted by $\hat{\mathbb{N}}(\hat{G}, s, \hat{T}, h)$, is the final outcome of our reduction. We are now able to prove ([10]):

Lemma 11: Let $\mathbb{N}(G, s, T, h)$ be a feasible instance of the

Algorithm NET-COD ($\mathbb{N}$):
input: Coding network $\mathbb{N}$

1 Transform $\mathbb{N}$ into a simple network $\hat{\mathbb{N}}$ as described in Section III-A.
2 Find **any** feasible network code for $\hat{\mathbb{N}}$, e.g., by using algorithms appearing in [4], [5].
3 Reconstruct the corresponding network code for $\mathbb{N}$.

Fig. 2. Algorithm NET-COD

network coding problem. Then, one can efficiently construct a simple instance $\hat{\mathbb{N}}(\hat{G}, \hat{s}, \hat{T}, h)$ for which (a) $|\hat{T}| = |T|$, (b) the size of the feedback link set of $\hat{\mathbb{N}}$ is less than or equal to that of $\mathbb{N}$, and (c) any feasible network code for $\hat{\mathbb{N}}$ with ℓ encoding nodes can be used to efficiently construct a feasible network code for $\mathbb{N}$ with at most ℓ encoding nodes.

B. The value of $Opt(\mathbb{N})$ in simple instances

In [10] we show that for simple instances $\mathbb{N}(G, s, T, h)$ the value of $Opt(\mathbb{N})$ is equal to the number of nodes in G (excluding the terminals) with in-degree 2.

Lemma 12: Let $\mathbb{N}(G, s, T, h)$ be a simple instance of the network coding problem. Let $\mathbb{F}(\mathbb{N})$ be any feasible network code for $\mathbb{N}$. Then, a node $v \in G$, $v \neq s$, $v \notin T$, is an encoding node in $\mathbb{F}(\mathbb{N})$ if and only if the in-degree of v is 2.

C. The algorithm

Lemma 12 implies that for any given simple network $\mathbb{N}(G, s, T, h)$, any upper bound on the number of nodes of in-degree 2 is also an (efficiently constructible) upper bound on $Opt(\mathbb{N})$. Accordingly, in [10] we prove the following theorem:

Theorem 13: Let $\mathbb{N}(G, s, T, h)$ be a simple instance of the network coding problem. Then, the number of nodes of in-degree 2 in G is bounded by $h^3k^2(2B + 1)$, where B is the size of the minimum feedback link set of G and $k = |T|$. In particular, if G is acyclic, the number of such nodes is bounded by h^3k^2.
This theorem constitutes the main result of our study. The proof of the theorem is rather involved and omitted from this version due to space constraints.

Theorem 13 leads to the following efficient procedure for finding a feasible network code with a bounded number of encoding nodes (implied by Lemma 11). The procedure works for a general (not necessarily simple) instance of the network coding problem $\mathbb{N}(G, s, T, h)$. We begin by transforming $\mathbb{N}$ into a simple network $\hat{\mathbb{N}}$. Then, we find any feasible network code for $\hat{\mathbb{N}}$. Finally, we reconstruct the corresponding network code for the original network $\mathbb{N}$. The description of our procedure appears in Figure 2.

IV. CONCLUSION

We consider the design of network codes which enable the source to transmit at rate h to k terminals and include a bounded number of encoding nodes. For acyclic networks, we present an efficient and simple procedure which finds a network code that enables the source to transmit at capacity, in which the number of encoding nodes is independent of the size of the network and is bounded by h^3k^2. We show that our bound on the number of encoding nodes may depend both on h and k as we present networks in which any feasible network code has at least $\Omega(h^2k)$ encoding nodes. It would be interesting if the hk gap between our upper and lower bound could be settled.

For general (cyclic) networks we present results of similar nature. Namely, we present an upper bound which depends on the size of the minimum feedback link set B of the network of size $(2B + 1)h^3k^2$. Our lower bound in this case is of order $\max(|V|/2, Bh, h^2k)$ where $|V|$ is the total number of nodes in the network.

In the proof of Theorem 5 that appears in [10] we present instances $\mathbb{N}$ to the network coding problem which establish a lower bound of $\frac{(h-1)h}{2}(k-1)$ on $Opt(\mathbb{N})$. Matthew Cook [13] has suggested an elaborated construction that establishes an improved lower bound of $(h-1)h(k-1)$.

Acknowledgments

We would like to thank Matthew Cook for useful discussions.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear Network Coding. *IEEE Transactions on Information Theory*, 49(2):371 – 381, 2003.

[3] R. Koetter and M. Medard. An Algebraic Approach to Network Coding. *IEEE/ACM Transactions on Networking*, 11(5):782 – 795, 2003.

[4] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros. The Benefits of Coding over Routing in a Randomized Setting. In *Proceedings of the IEEE International Symposium on Information Theory*, 2003.

[5] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial Time Algorithms for Multicast Network Code Construction. *Submitted to IEEE Transactions on Information Theory*, 2003.

[6] C. Fragouli, E. Soljanin, and A. Shokrollahi. Network Coding as a Coloring Problem. In *Proceedings of CISS*, 2004.

[7] C. Fragouli and E. Soljanin. Information Flow Decomposition for Network Coding. *Submitted to IEEE Transactions on Information Theory*, 2004.

[8] A. Tavory, M. Feder, and D. Ron. Bounds on Linear Codes for Network Multicast. *Electronic Colloquium on Computational Complexity (ECCC)*, 10(033), 2003.

[9] Y. Wu, K. Jain, and S.Y. Kung. A Unification of Menger's and Edmonds' Graph Theorems and Aslswede et al's Network Coding Theorem. *Allerton Conference on Communications, Control, and Computing*, 2004.

[10] M. Langberg, A. Sprintson, and J. Bruck. The Encoding Complexity of Network Coding. ETR063, California Institute of Technology, November, 2004. Available from: http://www.paradise.caltech.edu/ETR.html.

[11] M.R. Garey and D.S. Johnson. *Computers and Intractability*. Freeman, San Francisco, CA, USA, 1979.

[12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Networks Flows*. Prentice-Hall, NJ, USA, 1993.

[13] Matthew Cook. Private communication. 2004.