Comparing Network Coding with Multicommodity Flow for the k-pairs Communication Problem

Nicholas J. A. Harvey* Robert D. Kleinberg† April Rasala Lehman*

November 24, 2004

Abstract

Given a graph $G = (V, E)$ and k source-sink pairs of vertices, this papers investigates the maximum rate r at which all pairs can simultaneously communicate. We view this problem from two perspectives and compare their advantages. In the multicommodity flow formulation, a solution provides dedicated bandwidth r between each source-sink pair. In the information flow formulation, a vertex can transmit a function of the information it received thereby allowing multiple source-sink pairs to share bandwidth. For directed acyclic graphs with n vertices, we show that the rate achievable in the information flow formulation can be a multiplicative factor n larger than the rate achievable in the multicommodity flow formulation. It is well known [5] that for undirected graphs with n vertices, in the multicommodity flow formulation, the maximum rate achievable can be an $O(1/\log|V|)$ multiplicative factor smaller than the value of the sparsest cut. We extend this result to show that the maximum rate achievable in the information flow setting can be an $O(1/\log|V|)$ multiplicative factor smaller than the sparsest cut value.

For directed acyclic graphs G, we define a parameter called the *value of the most meager cut* which is an upper bound for the maximum rate achievable in the information flow setting. We also present an example illustrating that this upper bound is not always tight.

1 Introduction

We consider the k-pairs communication problem. The k-pairs communication problem has been studied in two different models: the multicommodity flow model and the network coding model. In this section we describe the k-pairs communication problem and then describe each of these models.

Given a graph $G = (V, E)$ and k pairs of vertices $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, the k-pairs communication problem is to provide a way for all k pairs to communicate simultaneously at a specified rate. The graph G may be either directed or undirected. When the distinction is important, we will specify which sort of graphs are under consideration, and we will use the term *arc* for directed edges. Each source s_i has access to an independent information source and each sink t_i wishes to be able to receive enough information to replicate this information source. The rate of the ith information source is denoted $dem(i)$. Each edge $(u, v) \in E$ represents a rate $c(u, v)$ zero-error communication channel with latency $l(u, v)$ between u and v. Unless otherwise

*MIT Computer Science and Artificial Intelligence Laboratory. {nickh,rasala}@mit.edu.

†MIT Department of Mathematics. rdk@math.mit.edu.

specified we assume that all information sources and channels operate at the same rate. If edge latencies are not specified then the network is assumed to be delay-free.

In the multicommodity flow formulation of the k-pairs communication problem, the rate $dem(i)$ is called the *demand* between s_i and t_i. A flow f_i of rate r_i is a collection of paths $\mathcal{P}_i$ from s_i to t_i and a positive real value $f_i(P)$ for each $P \in \mathcal{P}_i$ such that

$$\sum_{P \in \mathcal{P}} f_i(P) = r_i.$$

A multicommodity flow $F = \{f_1, f_2, \ldots, f_k\}$ is said to be feasible if

$$\sum_{P : e \in P} \sum_{i : P \in \mathcal{P}_i} f_i(P) \leq c(e) \quad \forall e \in E,$$

where $c(e)$ is the called the *capacity* of the edge e. The maximum total multicommodity flow problem is to maximize the sum of the flow rates between the source-sink pairs. The maximum concurrent multicommodity flow problem is to find a feasible flow maximizing the minimum fraction of demand routed between each source-sink pair.

In the network coding (or information flow) formulation of the problem, vertices in G are allowed to replicate and encode information they receive before transmitting a signal on an outgoing edge. The model is simplest to describe on a directed acyclic graph. In Section 4, we describe the extension of the model to undirected graphs. For simplicity, we assume that every communication pair is demanding a rate 1 connection and each edge is a rate 1 channel. Note that it is straightforward to transform a problem with k communication pairs requesting arbitrary integer rates to a problem with more communication pairs each requesting a rate 1 connection. Similarly, by scaling the rates appropriately and adding multiple links between pairs of vertices, one can simulate arbitrary integer capacities on the edges.

Assume $G = (V, E)$ is a directed acyclic graph. For each of the k communication pairs there is a source s_i with a message M_i and a sink t_i that needs to be able to reconstruct M_i from the information it receives on incoming edges. Each source generates a new message at each time step. A *solution* is a choice of a finite alphabet Σ from which the messages are drawn. The alphabet is assumed to be a finite set, possibly with additional structure such as field operations on its elements. The solution also specifies a function $f_{(u,v)} : \Sigma^k \to \Sigma$ for each edge $(u, v) \in E$ such that

- For each edge $(u, v) \in E$ the function $f_{(u,v)} : \Sigma^k \to \Sigma$ is computable from the messages for which u is a source and the information u receives on incoming edges.
- Each sink t_i can compute message M_i from the information it receives on incoming edges.

Any integral multicommodity flow solution can be transformed into a network information flow solution that performs only trivial coding operations. More generally, a fractional multicommodity flow solution can be transformed into a network information flow solution over a sufficiently large alphabet: the alphabet is treated as a vector space, and thus components of a symbol can be routed independently. The key distinction between network coding solutions and multicommodity flow solutions is that a network code allows messages M_i and M_j to "share" capacity on an edge (u, v) by letting $f_{(u,v)}$ be a function that depends on both M_i and M_j. For example, a network code could set $f_{(u,v)}(M_1, M_2, \ldots, M_k) = M_i + M_j$.

2 Related Work

This papers relates to two well studied problems: network information flow problems and multicommodity flow problems.

Network coding and the network information flow model were introduced by Ahlswede et al. [1] and have been primarily studied in the multicast scenario on directed acyclic graphs. An important special case is the *multicast problem*, in which a single source transmits the same k messages to each sink (i.e., k rate 1 connections must be established to each sink). Multicast problems have a network coding solution if and only if the min-cut between the source and each sink has capacity at least k [1, 6, 3, 2]. Solving the multicast problem without coding is equivalent to fractionally packing the maximum number of Steiner trees rooted at the source. Jaggi et al. [2] showed that the best achievable rate to all sinks using network coding can be a factor $\Omega(\log |V|)$ larger than the maximum fractional Steiner packing value. Recently, the problem of multicast communication on an undirected graph was considered [7, 8]. Their work shows two important results: the gap (i.e., ratio) between the maximum achievable network coding rate and the fractional Steiner packing value is at most 2 and can be as large as 9/8. Furthermore, the breakthrough work of Lau [4] proves that the gap between the maximum *integer* Steiner packing value and the edge-connectivity of the terminals is at most 26; this immediately implies that the best achievable rate with network coding is at most 26 times the maximum integer Steiner packing value. These results suggest that the advantage of network coding diminishes on undirected graphs. In experimental work, Wu et al. [10] tested the advantage of network coding over Steiner tree packing on the network graphs of six internet service providers and found that network coding did not provide significantly improved communication rates.

Whereas network information flow is a fairly recent area, multicommodity flow is a well-studied problem, having been considered by Ford and Fulkerson as early as 1954 [9, Chapter 70]. It was observed by Ford and Fulkerson that their famous max-flow min-cut theorem does not extend to multicommodity flow scenarios: the existence of s_i-t_i flows of value $dem(i)$ does not imply the existence of a multicommodity flow solution since those flows may not be disjoint. Another necessary condition for the existence of a multicommodity flow is the so-called *cut condition*, which states that the capacity of each cut must be at least the total demand of the source-sink pairs separated by that cut. The cut condition was conjectured to be sufficient, but a counterexample was found in 1963. A much stronger result was shown in the seminal work of Leighton and Rao [5]: there exist graphs with n vertices in which the minimum "sparsity" of a cut can be larger than the maximum concurrent multicommodity flow value by a factor of $\Omega(\log n)$. Section 5 discusses this result in more detail and generalizes it to the network information flow model.

3 Directed Acyclic k-Pairs Communication

In this section, we describe a k-pairs communication instance $I_{\mathtt{DAG}}$ in a directed graph where network coding provides a significant benefit over traditional multicommodity flow. More precisely, network coding allows a factor of $\Theta(n)$ more messages to be transmitted to their sinks, where n is the number of vertices in the graph.

Fix an arbitrary positive integer k and define G to be a directed graph with vertex set

$$V = \{s_1, \ldots, s_k, u, v, t_1, ..., t_k\}.$$

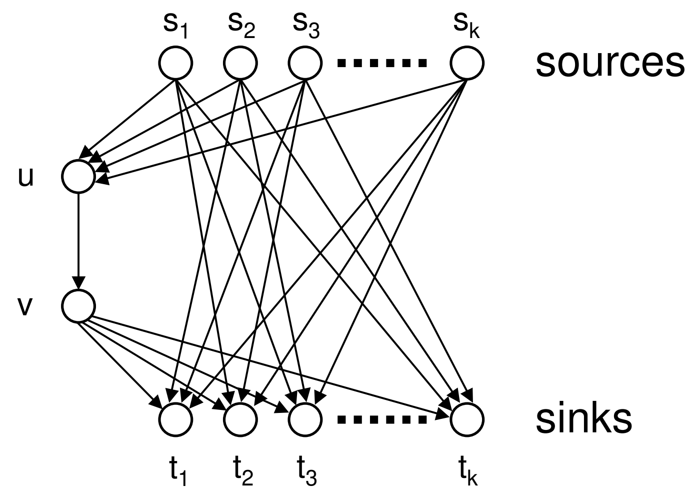

Figure 1: The graph G described in Section 3. We wish to send a message from s_i to t_i for each i. The best multicommodity flow solution can transmit at most one message to its sink, since the arc (u, v) can send at most one message. A network coding solution can overcome this limitation by sending the XOR of all the messages on the arc (u, v). A sink can reverse the effect of this XOR and extract its desired message by using the messages it receives directly from the other sources.

The communication pairs are $\{ (s_i, t_i) : 1 \le i \le k \}$. The arcs are

$$\begin{aligned} A &= E_1 \cup E_2 \cup \{(u, v)\}, \\ E_1 &= \{ (s_i, u) : 1 \le i \le k \} \cup \{ (v, t_j) : 1 \le j \le k \}, \\ E_2 &= \{ (s_i, t_j) : i \ne j \}. \end{aligned}$$

The graph G is depicted in Figure 1.

Lemma 1. *For instance $I_{\mathtt{DAG}}$, the maximum total multicommodity flow rate is* 1 *and the maximum concurrent multicommodity flow rate is $O(1/n)$.*

Proof. Note that for each i, there is only one s_i-t_i path, and this path traverses the "bottleneck" arc (u, v). Since all arcs (and in particular arc (u, v)) have unit capacity, the maximum total value of a multicommodity flow is 1. Since there are $\Theta(n)$ s_i-t_i pairs, the maximum concurrent multicommodity flow rate achievable is $\Theta(1/n)$. □

Thus without coding, at most one message can be successfully transmitted to its sink. With network coding, we now show that all k messages can be transmitted to their respective sinks.

Lemma 2. *For instance $I_{\mathtt{DAG}}$, the maximum concurrent information flow rate achievable is $\Theta(1)$ and the maximum total information flow rate is $\Theta(n)$.*

Proof. We describe a network coding solution that allows each communication pair to simultaneously communicate at rate 1. The alphabet Σ is equal to $\{0, 1\}$. Each source vertex transmits its message on all outgoing arcs. The vertex u computes the XOR of all messages that it receives and transmits the resulting value to vertex v. Vertex v forwards this message to all the sinks.

Since each sink t_j obtains the message from every source s_i ($i \neq j$) using the E_2 arcs, it can compute the XOR of all messages it receives and thereby recover the message from s_j. Thus all $k = \Omega(n)$ messages can be transmitted to their respective sinks by network coding. $\square$

Combining Lemmas 1 and 2, we get the following gap.

Theorem 3. *There exists instances of the k-pairs communication problem on directed acyclic graphs for which maximum total information flow rate is $\Omega(n)$ times the maximum total multicommodity flow rate and the maximum concurrent information flow rate is $\Omega(n)$ times the maximum concurrent multicommodity flow rate.*

4 Network Coding on Undirected Graphs

This section defines our information flow model for the k-pairs communication problem on an undirected graph. This model necessarily differs from the model for directed acyclic graphs since we must eliminate the possibility of feedback in the coding functions. In order to achieve this, we consider a solution over time and assume that all edges have latency 1. To make this precise we consider a representation of the information flow in G between time 0 and some time T using a layered graph $G(T) = (V(T), E(T))$. For each time $0 \leq t \leq T$, there is a set of nodes V_t which is a copy of the vertex set V of G. For $v \in V$, let v_t denote the copy of v that is in V_t. For each edge $\{u, v\} \in G$ and each time $t \geq 1$, there is an arc (u_{t-1}, v_t) and an arc (v_{t-1}, u_t). Let E_t be the set of arcs between nodes in V_{t-1} and nodes in V_t. For the purposes of this paper it suffices to allow each directed arc to have the same capacity as the associated edge, i.e., $c(u_{t-1}, v_t) = c(v_{t-1}, u_t) = c(u, v)$. Thus, for some finite alphabet Σ we assume that each directed arc transmits one symbol from Σ at each time step. In the information flow formulation, as before, each vertex $v_t \in V_t$ is allowed to transmit on its outgoing edge (v_t, u_{t+1}) at time t a function $f_{(v_t, u_{t+1})}$ of the information v_t received from nodes in V_{t-1}.

Having specified a procedure for transforming an undirected graph G into a directed acyclic graph $G(T)$ which models the flow of information through G over time, we must now specify how we model the production and reception of messages by the senders and receivers over time. For these purposes, we assume that for each sender-receiver pair $(s_i, t_i) = (u, v)$ we have $k(u, v) = dem(i) \cdot r \cdot (T - L)$ messages $M_1(u, v), M_2(u, v), \ldots, M_{k(u,v)}(u, v)$, each of which consists of a single symbol from the alphabet, Σ. Here r is a parameter specifying the rate of the solution, and $L \ll T$ is a parameter which may be thought of as specifying the length of a "warm-up" (resp. "cool-down") period during which messages need not be received (resp. sent). This is to model the fact that messages which must travel many hops through the network cannot possibly be received during the first few steps of the timeline, nor can they be sent during the last few steps of the timeline. By defining $k(u, v)$ to be equal to $dem(i) \cdot r \cdot (T - L)$ we model a communication pair which must send/receive $dem(i) \cdot r$ messages during each period except the warm-up/cool-down periods.

Definition 1. *Given a communication problem represented by an undirected graph $G = (V, E)$, communication pairs (s_i, t_i), demands $dem(i)$, and capacities $c(e)$, we say that rate r is achievable if as $T \to \infty$, there is a function $L = L(T) = o(T)$ such that the following holds. First, for every communication pair $(s_i, t_i) = (u, v)$, let*

$$k(u, v) = dem(i) \cdot r \cdot (T - L).$$

Now let

$$\mathcal{M}(u,v) = \{M_1(u,v), M_2(u,v), \ldots, M_{k(u,v)}(u,v)\}$$

be a set of $k(u,v)$ messages. (The set $\mathcal{M}(u,v)$ may be empty for some (u,v) pairs.) Finally let

$$\mathcal{M} = \bigcup_{(u,v)\in V\times V} \mathcal{M}(u,v)$$

be the set of all such messages. There must exist two mappings $\sigma, \tau : \mathcal{M} \to V(G(T))$ satisfying the following criteria:

1. *For all $m = M_j(u,v) \in \mathcal{M}$, $\sigma(m) = u_t$ for some t and $\tau(m) = v_t$ for some t.*
2. *There exists a network code for $G(T)$ which solves the network information flow problem in $G(T)$ with message set $\mathcal{M}$, in which the source and sink for message m are $\sigma(m), \tau(m)$ respectively.*

Note that one restriction of our model is that we restrict node v to only transmit a function of the information it received in the previous time step. Thus, node v is not allowed to use any information in its memory when computing the messages to be transmitted on its outgoing edges. Moreover, the assumption that messages must be sent and received at vertices in $V(G(T))$ is essentially tantamount to requiring that the messages must be transmitted and decoded in real-time. A more relaxed model would feature a "super-source" node u^* for each vertex $u \in V(G)$, attached to the nodes u_t of $G(T)$ by infinite-capacity edges, and a "super-sink" node v^* for each $v \in V(G)$, attached to the nodes v_t of $G(T)$ by infinite-capacity edges The sender and receiver for message $m = M_j(u,v)$ would be u^*, v^*, respectively. This more relaxed model would correspond to a scenario in which information flows through the graph over time, but the receivers are not required to decode *any* messages until the end of the time-line when all information has been received. Another possible model could allow "memory edges", with bounded or potentially infinite capacity, between any vertices v_t and v_{t+1}. These memory edges model the notion that a vertex of G may be able to store information and use it at a later time.

5 Undirected k-Pairs Communication Problem

In this section we consider the relationship between the capacity of cuts in the graph G and the maximum concurrent information flow between all pairs of vertices in G. For this discussion we will need to define cuts in undirected and directed acyclic graphs.

5.1 Cuts in Undirected Graphs

In an undirected graph a cut can be defined in two ways. First, for a subset $U \subseteq V$, the cut defined by U is the set of edges with one endpoint in U and one endpoint in V. Second, and more generally, a set $A \subseteq E$ is a cut if it separates at least one source-sink pair. In general, we can define the capacity of the cut, the demand crossing the cut and the sparseness of the cut for either definition of a cut.

Let U be a subset of the vertices of G and let $A \subseteq E$ be the set of edges with one endpoint in U and one endpoint in $(U, V-U)$. The capacity $C(U, V-U) = C(A)$ of the cut defined by U is the sum of the capacities of the edges in A. The demand $D(U, V-U) = D(A)$ across the cut defined by U is equal to the total demand between source-sink pairs separated by A. The ratio of the capacity of the cut A to the demand between source-sink pairs separated by A is called the *sparseness* of A.

Definition 2 (Sparseness). *Let $U \subset V$ be a subset of the vertices and let $A \subset E$ be the set of edges with one endpoint in U and one endpoint in $V - U$. The sparseness of the cut defined by U is*

$$\mathcal{S}(U, V - U) = \mathcal{S}(A) = \frac{C(A)}{D(A)}.$$

For a graph G, the value of the sparsest cut is defined as:

$$\mathcal{S}_G = \min_{A \subseteq E} \mathcal{S}(A).$$

5.2 Cuts in Directed Graphs

If G is a directed graph, it is possible to define the sparseness of a cut as above, but for our purposes we will be more concerned with a different parameter, which we will call "meagerness," whose definition is very similar to the definition of sparseness.

Definition 3 (isolation, meagerness). *Given an edge set $A \subset E$ and a subset of source-sink pairs $P = \{(s_i, t_i) : i \in I\}$, we say A* isolates *P if for $i, j \in I$, every path from s_i to t_j intersects A. We denote $\sum_{i \in I} dem(i)$ by $D(P)$. The* meagerness *of the cut A, denoted by $\mathcal{M}(A)$, is defined to be ∞ if A does not separate any source-sink pair (s_i, t_i), and is otherwise defined by:*

$$\mathcal{M}(A) = \min \left\{ \frac{C(A)}{D(P)} \ : \ A \text{ isolates } P \right\},$$

where the minimum is taken over all sets of source-sink pairs. The value of the most meager cut in G is denoted by

$$M_G = \min_{A \subseteq E(G)} \mathcal{M}(A).$$

Note that meagerness is a weaker notion than sparsity, i.e., $M_G \geq S_G$ for any graph G.

5.3 Sparse Cuts and Concurrent Flows

The value of the sparsest cut is an upper bound on the rate of the maximum concurrent multicommodity flow, and the value of the most meager cut is an upper bound on the rate of the maximum concurrent information flow. We state this as a lemma for reference later.

Lemma 4. *Given an undirected graph G, the rate of the maximum concurrent multicommodity flow is at most S_G. Given a directed acyclic graph H, the rate of the maximum concurrent information flow is at most M_H.*

An important result of Leighton and Rao [5] shows that the rate δ of the maximum concurrent multicommodity flow can be significantly smaller than the value of the sparsest cut. In particular, there exists a graph G for which

$$\delta = O\left(\frac{1}{\log n}\right) \mathcal{S}_G.$$

In this section we extend this result to the network coding setting. First we describe the undirected k-commodity flow instance of Leighton and Rao [5] on the graph $G = (V, E)$. Second, we modify this instance slightly so that the sparsest cut value is 1. Next we consider the directed acyclic graph $G(T)$ that represents the information flow over time for this instance. We demonstrate that the maximum rate of network information flow achievable is $O\left(\frac{1}{\log |V|}\right)$.

Let $G = (V, E)$ be a 3-regular expander. For each pair of vertices in G there is a demand of rate 1. Since G is an expander, there exists a $\rho = \Theta(1)$ such that for all $U \subseteq V$ the number of edges crossing the cut between U and $V - U$ is at least $\rho \cdot \min\{|U|, |V - U|\}$.

We scale the capacity of each edge of G in order to make the sparsest cut value equal to 1. Let the capacity of each edge $(u, v) \in E$ be $c(u, v) = \frac{n-1}{\rho}$ where $n = |V|$.

Lemma 5 (Leighton-Rao [5]). *The value of the sparsest cut $\mathcal{S}_G$ in G is* 1.

Proof. The value of the sparsest cut in G can be bounded from below as follows.

$$
\begin{aligned}
\mathcal{S}_G &= \min_{A \subseteq E} \frac{C(A)}{D(A)} \\
&= \min_{U \subseteq V} \frac{C(U, V-U)}{D(U, V-U)} \\
&\geq \min_{U \subseteq V} \frac{\big((n-1)/\rho\big) \cdot \big(\rho \cdot \min\{|U|, |V-U|\}\big)}{|U| \cdot |V-U|} \\
&= \min_{U \subseteq V} \frac{(n-1)}{\max\{|U|, |V-U|\}}
\end{aligned}
$$

This quantity is minimized when $|U| = 1$ and $|V - U| = n - 1$.

$$\mathcal{S}_G \geq \frac{n-1}{n-1} = 1$$

□

Theorem 6. *The maximum rate of network information flow achievable in G (in the sense of Definition 1) is $O\left(\frac{1}{\log n}\right)$.*

Proof. For some rate $r > 0$, let $T, L, k(u, v), \mathcal{M}(u, v)$, and $\mathcal{M}$ be as in Definition 1, and assume that rate r is achievable and that this is verified by two functions σ, τ satisfying the two criteria in that definition. Let T_0 denote a random variable uniformly distributed in the set $\{1, 2, \ldots, T\}$. Let $V^- \subset V(G(T))$ denote the set of vertices w_t where $1 \leq t < T_0$, and let $V^+ = V(G(T)) \setminus V^-$. Finally let

$$\mathcal{M}_0 = \left\{ m \in \mathcal{M} \,:\, \sigma(m) \in V^-, \tau(m) \in V^+ \right\}.$$

We begin by computing the expected cardinality of $\mathcal{M}_0$. This is the sum, over all messages $m \in \mathcal{M}$, of the probability that $m \in \mathcal{M}_0$. Now let $m = M_j(u, v)$, and let $d(u, v)$ be the length of the shortest path joining u and v in G. Let $u_t = \sigma(m), v_{t'} = \tau(m)$. We must have $t' - t \geq d(u, v)$, as otherwise there would be no path in $G(T)$ from $\sigma(m)$ to $\tau(m)$, violating the fact that there exists a network code for $G(T)$ which solves the network information flow problem with message set $\mathcal{M}$, sources $\sigma(m)$, and sinks $\tau(m)$. Thus

$$\Pr(m \in \mathcal{M}_0) = \Pr(t < T_0 \leq t') = (t' - t)/T \geq d(u, v)/T,$$

and summing over all messages $m \in \mathcal{M}$ we obtain

$$\mathbf{E}(|\mathcal{M}_0|) = r \cdot (T - L) \cdot \sum_{u,v \in V(G)} \frac{d(u,v)}{T} = r \cdot (1 - o(1)) \cdot \sum_{u,v \in V(G)} d(u,v). \tag{1}$$

We claim that the sum on the right side of (1) is $\Omega(n^2 \log(n))$. To prove this, it suffices to show that for each $u \in V(G)$, the number of vertices within distance $\log_2(n/6)$ of u is at most $n/2$. This follows directly from the Moore bound: since G is 3-regular, the number of vertices at distance *exactly* d from u is at most $3 \cdot 2^{d-1}$, for all $d \geq 1$.

Now let a fixed value of T_0 be chosen, such that $|\mathcal{M}_0|$ is greater than or equal to its expected value, i.e.

$$|\mathcal{M}_0| = \Omega(rn^2 \log(n)).$$

The set of edges separating V^- from V^+ is $E_{T_0} = \{(u_{T_0-1}, v_{T_0}) : (u,v) \in E(G)\}$. It must be possible to reconstruct every message in $\mathcal{M}_0$ from the set of messages traversing the edge set E_{T_0}; thus the combined capacity of these edges must be $\Omega(rn^2 \log(n))$. However, we know that

$$C(E_{T_0}) = 2C(E(G)) = 3n(n-1)/\rho = O(n^2).$$

Hence it must be the case that $r = O\left(\frac{1}{\log n}\right)$. □

6 Maximum Concurrent Information Flow Rate ≠ Meagerness

Recall Lemma 4, which presents upper bounds on the rates of the maximum concurrent multicommodity flow and the maximum concurrent information flow in a graph G. In the remarks immediately following Lemma 4, we presented an example due to Leighton and Rao which demonstrates that this upper bound is not tight, in the multicommodity flow setting. In this section, we will present an example illustrating that Lemma 4 also fails to give a tight bound in the information flow setting. Specifically, we will give an example of a directed acyclic graph G in which the meagerness of every cut is at least 1, yet the maximum concurrent information flow rate is 2/3. The graph is illustrated in Figure 2. All edges have unit capacity, and all three commodities have unit demand.

Lemma 7. *The value of the most meager cut in G is 1.*

Proof. Suppose A is a cut which isolates a set P of source-sink pairs. If $|P| = 1$, then $|A| \geq 1$ since each source s_i is joined to the corresponding sink t_i by at least one path in G. If $|P| = 2$ and $(s_3, t_3) \in P$ then $|A| \geq 2$ since s_3 is joined to t_3 by two edge-disjoint paths. If $|P| = 2$ and $(s_3, t_3) \notin P$, then $P = \{(s_1, t_1), (s_2, t_2)\}$ and $|A| \geq 2$ because A must contain the edges (s_1, t_2) and (s_2, t_1). Finally, if $|P| = 3$, then P consists of all source-sink pairs in G. In this case, $|A| \geq 4$ because A must contain the edges (s_1, t_2) and (s_2, t_1) as well as an edge cut separating s_3 from t_3. In all cases, $|A| \geq |P|$ which verifies that the value of the most meager cut in G is at least 1. In fact it is equal to 1, for instance because the cut $A = \{(s_3, v)\}$ isolates the pair $P = \{(s_1, t_1)\}$. □

Theorem 8. *There does not exist a network code for G which achieves rate 1.*

Proof. Let $f : \Sigma^3 \to \Sigma$ be the function computed by the edge $e = (s_3, u) \in E(G)$. In other words, $f(a, b, c)$ is the message sent on edge e when a, b, c are the messages originating at s_1, s_2, s_3,

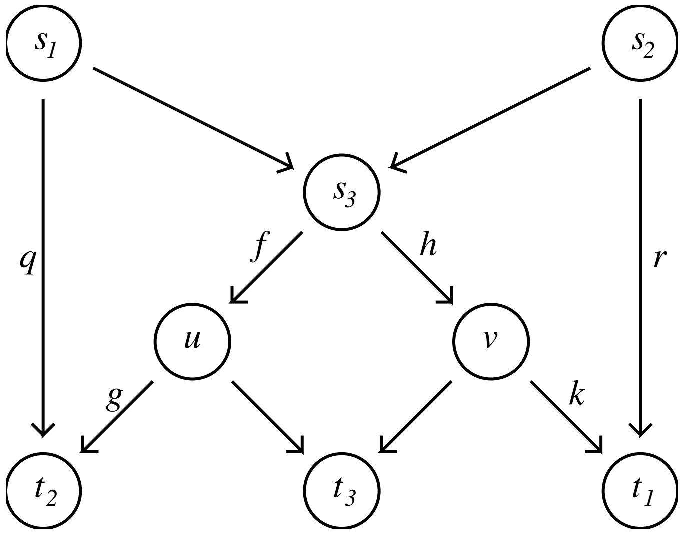

Figure 2: The graph G.

respectively. The functions $g, h, k, q, r : \Sigma^3 \to \Sigma$ are associated to their corresponding edges as shown in Figure 2. Let $\tilde{g} : \Sigma \to \Sigma$ denote the function mapping the message $f(a, b, c)$, which u receives from s_3, to the message $g(a, b, c)$, which it sends to t_2. For any given message $a \in \Sigma$, let $\tilde{d}_a : \Sigma \to \Sigma$ denote the function specified as follows: $\tilde{d}_a(x)$ is the output at node t_2 when it receives message $q(a)$ from s_1 and message x from u. (Here we are writing $q(a)$ rather than $q(a, b, c)$ since the function $q : \Sigma^3 \to \Sigma$ is only allowed to depend on its first argument.)

We claim that each of the functions $\tilde{d}_a$ and $\tilde{g}$ is a bijection. Since each of these functions maps Σ to Σ, it is sufficient to prove that both $\tilde{d}_a$ and $\tilde{g}$ are surjective. But considering the fact that

$$\tilde{d}_a(\tilde{g}(f(a, b, c))) = b, \tag{2}$$

we see that the composition $\tilde{d}_a \circ \tilde{g}$ is surjective, which implies that each of the maps individually must be surjective.

Next we claim that for any elements $a, b, c_0, c_1 \in \Sigma$, it must be the case that $f(a, b, c_0) = f(a, b, c_1)$. Indeed, using equation (2) together with the fact that both $\tilde{d}_a$ and $\tilde{g}$ are bijective, we obtain

$$f(a, b, c_0) = f(a, b, c_1) = \tilde{g}^{-1}(\tilde{d}_a^{-1}(b)).$$

Arguing symmetrically, we may also conclude that $h(a, b, c_0) = h(a, b, c_1)$ for all elements $a, b, c_0, c_1 \in \Sigma$. Fixing a and b to be any pair of elements of Σ, this means that the values of $f(a, b, c)$ and $h(a, b, c)$ do not vary as c runs through all the elements of Σ; consequently, the output of node t_3 doesn't vary as c runs through Σ. This contradicts the fact that the output of node t_3 is equal to c. $\square$

Given that Theorem 8 shows that network coding cannot achieve rate 1 in G, one may wish to know the maximum achievable rate. Observe that this rate is at least equal to 2/3, the rate of the maximum concurrent multicommodity flow in G. (This flow sends 2/3 units of flow along the unique path from s_1 to t_1 and along the unique path from s_2 to t_2, and it sends 1/3 units of flow along each of the two paths from s_3 to t_3.) The following theorem demonstrates that this lower bound of 2/3 is actually tight, i.e. no network code can achieve a higher rate of information flow. In the theorem, we assume that the messages are drawn from an alphabet Σ and that the values transmitted along the edges are drawn from a possibly different alphabet Σ'.

Theorem 9. *If $|\Sigma| > |\Sigma'|^{2/3}$, there does not exist a network code for G that solves the corresponding network information flow problem.*

Proof. Define $\tilde{d}_a : \Sigma' \to \Sigma$ and $\tilde{g} : \Sigma' \to \Sigma'$ as before. It is no longer the case that $\tilde{d}_a$ and $\tilde{g}$ must be bijections, since it is possible that $|\Sigma'| > |\Sigma|$. However, equation (2) is still valid. Consequently, if a, b_0, c_0, b_1, c_1 are any elements of Σ satisfying $f(a, b_0, c_0) = f(a, b_1, c_1)$ we must have

$$b_0 = \tilde{d}_a(\tilde{g}(f(a, b_0, c_0))) = \tilde{d}_a(\tilde{g}(f(a, b_1, c_1))) = b_1.$$

Given $a, b \in \Sigma$, let us define a set

$$F(a, b) = \{f(a, b, c) : c \in \Sigma\}.$$

We have seen that for any elements $a, b_0, c_0, b_1, c_1 \in \Sigma$,

$$f(a, b_0, c_0) = f(a, b_1, c_1) \quad \Longrightarrow \quad b_0 = b_1,$$

from which we may conclude that $F(a, b_0), F(a, b_1)$ are disjoint subsets of Σ' when $b_0 \neq b_1$. Thus

$$\sum_{b \in \Sigma} |F(a, b)| \leq |\Sigma'|$$

for each $a \in \Sigma$, and hence

$$\sum_{a,b \in \Sigma} |F(a, b)| \leq |\Sigma||\Sigma'|.$$

Dividing both sides by $|\Sigma|^2$, we obtain

$$\frac{\sum_{a,b} |F(a, b)|}{|\Sigma|^2} \leq \frac{|\Sigma'|}{|\Sigma|} < |\Sigma|^{1/2}.$$

The arithmetic-mean-geometric-mean inequality implies that

$$\prod_{a,b \in \Sigma} |F(a, b)|^{1/|\Sigma|^2} \leq \frac{\sum_{a,b} |F(a, b)|}{|\Sigma|^2},$$

and combining this with the preceding inequality we obtain

$$\prod_{a,b \in \Sigma} |F(a, b)| < |\Sigma|^{|\Sigma|^2/2}.$$

Now define $H(a, b) = \{h(a, b, c) : c \in \Sigma\}$ and argue symmetrically to derive the bound

$$\prod_{a,b \in \Sigma} |H(a, b)| < |\Sigma|^{|\Sigma|^2/2}.$$

Multiplying these two bounds we obtain

$$\prod_{a,b \in \Sigma} |F(a, b) \times H(a, b)| < |\Sigma|^{|\Sigma|^2},$$

from which it follows that at least one of the $|\Sigma|^2$ terms in the product on the left side must be less than $|\Sigma|$. Fix a pair of messages a, b such that $|F(a,b) \times H(a,b)| < |\Sigma|$. As c ranges over all elements of Σ, the ordered pair $(f(a,b,c), h(a,b,c))$ ranges over a subset of $F(a,b) \times H(a,b)$, hence it ranges over a set of values whose cardinality is strictly smaller than $|\Sigma|$. The output of node t_3 is determined by the ordered pair $(f(a,b,c), h(a,b,c))$, hence it can take on fewer than $|\Sigma|$ distinct values as c ranges over Σ. This contradicts the fact that the output of node t_3 is equal to c. □

7 Conclusions

We have considered the use of network coding and multicommodity flows for the k-pairs communication problem. For directed graphs, we have shown that a network coding solution can yield a factor of $\Theta(n)$ higher rate than the best multicommodity flow solution, in some instances.

For multicommodity flows in undirected graphs, the following relations between cuts and concurrent flows are well known: the value of the sparsest cut is an upper bound on the rate of a maximum concurrent multicommodity flow, this bound is tight to within a factor of $O(\log n)$, and the ratio $O(\log n)$ is best possible for some graphs. We have considered the question of whether analogous relations exist between cuts and network coding solutions. We have defined a notion of *meagerness*, similar to sparsity, such that the value of the most meager cut is an upper bound on the maximum rate achievable by a network coding solution, and have presented an example illustrating that this bound is not always tight. We have also demonstrated that there exist undirected graphs in which the value of the sparsest cut is equal to 1, yet no network coding solution can achieve a rate higher than $O(1/\log(n))$. This leads us to the following conjecture.

Conjecture 1. *For a k-pairs communication problem in an undirected graph, the maximum rate achievable by a network coding solution equals the maximum rate achievable by a fractional multicommodity flow.*

Acknowledgements

We thank David Karger and Muriel Médard for instigating this research and for helpful discussions. We also thank Eric Lehman and Madhu Sudan for helpful discussions on this topic.

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. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory.* Submitted July 2003.

[3] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking.* To appear.

[4] L. C. Lau. An approximate max-steiner-tree-packing min-steiner-cut theorem. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*, Oct. 2004.

[5] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, Nov. 1999.

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

[7] Z. Li and B. Li. Network coding in undirected networks. In *Proceedings of the 38th Annual Conference on Information Sciences and Systems (CISS 2004)*, Mar. 2004.

[8] Z. Li, B. Li, D. Jiang, and L. C. Lau. On achieving optimal end-to-end throughput in data networks: Theoretical and empirical studies, Feb. 2004. ECE Technical Report, University of Toronto.

[9] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, 2003.

[10] Y. Wu, P. A. Chou, and K. Jain. A comparison of network coding and tree packing. In *IEEE International Symposium on Information Theory (ISIT 2004)*, 2004.