Network Coding for Resource Redistribution in a Unicast Network

Jennifer Price and Tara Javidi
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
jenn-price@ucsd.edu, tara@ece.ucsd.edu

***Abstract*— In this paper, we explore the use of network coding as method for redistributing resources to achieve capacity in two-user unicast networks. We introduce a simple network example, which we call the *extended butterfly network*, whose structure illustrates the idea of resource swapping, and demonstrates possible issues related to user incentives for network coding. We show that for such a network, it is possible to construct a combination network coding/routing scheme that 1) achieves capacity of the network by employing network coding as a way to redistribute misplaced network resources, and 2) facilitates cooperation among users by creating a stable rate allocation in which users have no incentive to deviate from the network coding scheme.**

I. Introduction

Since the seminal work in [1], network coding has received a great deal of attention as a way to help bridge the gap between multi-commodity flow capacity and information theoretic capacity of networks [1], [2]. This has spawned an extensive and rich literature on network coding for both multicast [1]-[7] and unicast [8]-[12] networks. Although the multicast network coding problem is well understood, the problem of network coding for unicast networks is fundamentally different from that of network coding for multicast networks. Since information in a unicast network is sent between specific source-destination pairs, it necessarily requires inter-session network coding in which destinations are only interested in their own information.

The benefit of network coding for unicast comes about when a source has the ability to transmit information to *the other source's destination*. This resource is essentially misplaced in a multi-commodity flow context, but with network coding it can be used to transmit information for use in decoding network coded flows. If both sources have this type of misplaced resource, network coding allows the users to "swap" the misplaced resources. In other words, we can redistribute misplaced resources to be used in facilitating communication from a source to its own destination. This idea of using network coding to swap resources raises some interesting issues related to user incentives, however. Cooperation among users is required whenever network coding is used. In the more traditional multi-cast setting, users do not lose anything by sending redundant side information since this information needs to be broadcast to all users in the network. In a unicast network, however, a user gains nothing by sending side information unless the other users send side information as well. The question of what a user will do when faced with the choice between sending information to its own destination, and sending information for purposes of decoding at other users is an interesting one.

In this paper, we explore the use of network coding as method for swapping resources by introducing a simple network example which we call the extended butterfly. The structure of this network exactly illustrates the idea of resource swapping, as well as the notion of incentives for network coding. We show that for such a network, it is possible to construct a combination network coding/routing scheme that 1) achieves capacity of the network by employing network coding as a way to redistribute network resources, and 2) facilitates cooperation among users by creating a stable rate allocation in which users have no incentive to deviate from the network coding scheme.

The remainder of this paper is organized as follows. Section II introduces the extended butterfly network, and illustrates the idea of resource swapping and incentives. Section III introduces notations, definitions, and relevant results used throughout the paper. Section IV gives a procedure for finding an outer bound on capacity, while Section V gives a rate assignment scheme that achieves this bound. Finally, Section VI gives our conclusions and areas of future work.

II. A Simple Example: The Extended Butterfly Network

In this paper, we work with a simple example of a unicast network in order to illustrate the ideas of cooperation and resource swapping through network coding. Consider the extended butterfly network shown in Figure 1, consisting of two unicast sessions: one from S(1) to T(1), and one from S(2) to T(2). Link capacities are as indicated on the graph. Notice that in addition to the usual butterfly structure, there is a direct path from S(1) to T(1) that shares a link with a path from S(1) to T(2), and a direct path from S(2) to T(2) that shares a link with a path from S(2) to T(1).

Figure 2(a) shows one possible solution to the multi-commodity flow problem. Each source sends as much information as the link capacities allow to its own destination - here

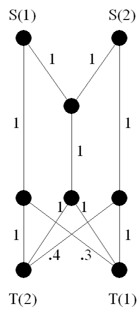

Fig. 1. The Extended Butterfly Network

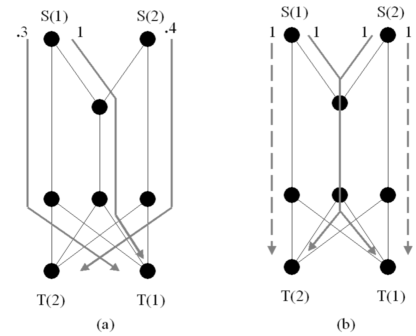

Fig. 2. Multi-Commodity Flow Solution (a) and Multi-Cast Network Coding Solution (b)

we have given all of capacity on the shared central link to user 1. The total rate achieved by this solution is $.3+1+.4 = 1.7$. We see here that the capacity along paths from S(1) to T(2) and S(2) to T(1) is wasted, since there is no benefit for sources to send information to the other source's destination in a multi-commodity situation.

In contrast, Figure 2(b) shows the solution to the multi-cast network flow problem, where information from each source is broadcast to both destinations[1]. Each source sends information along its path through the central link, and an equal amount of redundant information along its path to the other destination. Here, the use of network coding allows *both* users to send rate '1' over the central link. In other words, network coding allows users to share this link rather than compete for it. The total rate achieved by this solution is $1 + 1 = 2$, and is an improvement over the multi-commodity flow solution.

Notice that the resources used by each user to send the redundant information for decoding purposes could have been used to send information to its own destination instead. This raises two concerns. The first concern is that this creates a prisoner's dilemma-type situation. If both users network code, they are both better off. If one user decides not to network code, however, the other user is better off by not network coding. The second concern is, of course, that there still seems to be misplaced capacity in the network. Hence, it would seem that the solution to the multi-cast network flow solution may not be optimal. In fact, we will see in the following sections that the multi-cast network flow solution does *not* achieve capacity of the network.

In the remaining sections of this paper, we give an outer bound on the capacity of the extended butterfly network, and present a combination network coding/routing scheme that achieves this outer bound. We show that in addition to achieving capacity, this scheme does not result in a prisoner's dilemma-type scenario. In other words, we present a scheme that is both capacity-achieving, and stable in terms of incentives for network coding!

[1]We can think of this as the situation where we add a pseudo source node, that sends information from both sources. The resulting problem is an intra-session multi-cast network coding problem.

III. Notation, Definitions, and Relevant Background Information

Throughout this paper, we use the following notation. Denote by $G = (V, E)$ the directed unicast network given in Figure 1, where $\mathcal{N} = \{1, 2\}$ is the set of source-destination pairs. Following the convention introduced in [2] (and without loss of generality), we assume that each source has a single outgoing link $S(i)$ with infinite capacity, and each destination has a single incoming link $T(i)$ with infinite capacity. Let $\rho = \{S(1), S(2)\}$ and $\varphi = \{T(1), T(2)\}$. Let $\mathcal{L} = \{1, \ldots, L\}$ be the set of links (edges), each with capacity c_l and let $\mathcal{K} = \{1, \ldots, K\}$ be the set of paths from any source $S(i)$ to any destination $T(j)$.

We now introduce the following definitions that will aid us in our description of the network capacity and coding scheme.

Definition 1: A *cut* A is a set of links. We define the capacity of cut A to be $C(A) = \sum_{l \in A} c_l$.

Definition 2: A path is a *shared* path if it goes from $S(i)$ to $T(i)$, and it shares at least one link with a path that goes from $S(j)$ to $T(j)$. A path is a *dedicated* path if it goes from $S(i)$ to $T(i)$ and it is not shared. A path is a *redundant* path if it goes from $S(i)$ to $T(j)$.

In the following sections, we derive the capacity of the extended butterfly network by first constructing an outer bound on the capacity region, then showing achievability of the outer bound. In order to do so, we draw heavily from definitions and theorems given in [2]. This paper uses the concept of informational dominance to relate the structure of a graph to information-theoretic notions of capacity. In order to facilitate our discussion of network capacity and achievable rate regions, we summarize here the relevant facts, definitions, and theorems from this paper.

Definition 3: An edge set A *informationally dominates* edge set B if the information transmitted on edges in A determines the information transmitted on edges in B for all network coding solutions, regardless of the rate of the solution. We denote by $Dom(A)$ the set of all edges that are informationally dominated by A.

Fact 1: If $B \subseteq Dom(A)$, then $H(B) \leq H(A)$.

Fact 2: For an edge set A, the set $Dom(A)$ satisfies the following conditions:

1. $A \subseteq Dom(A)$.

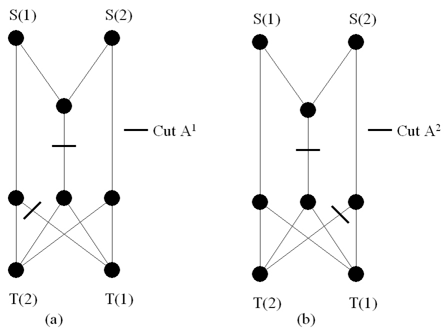

Fig. 3. Subgraphs and Cuts for Procedure P1: (a) Step 1 (b) Step 2

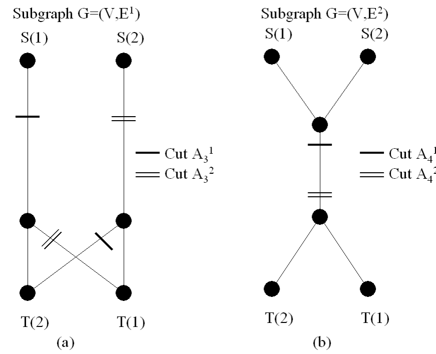

Fig. 4. Subgraphs and Cuts for Procedure P1: (a) Step 3 (b) Step 4

2. $S(i) \in Dom(A)$ if and only if $T(i) \in Dom(A)$.
3. Every edge in $E \backslash Dom(A)$ is reachable in $G \backslash Dom(A)$ from a source.
4. For every source edge $S(i)$ in $G \backslash Dom(A)$, there is an indirect walk for commodity i in $G(Dom(A), i)$.

(Theorem 10 from [2])

Fact 3: Given a directed acyclic graph G, a network coding solution with rate vector $\underline{r}$ exists if and only if there exists a constant b such that the following hold: for all edges $l \in \mathcal{L}$,

$$H(l) \leq c_l \log_2(b)$$

and for all sources $i \in \mathcal{N}$

$$H(S(i)) \geq r_i \log_2(b)$$

Again, we emphasize that Definition 3 and Facts 1-3 are relevant results from [2] that we will use in examining our combined network coding/routing scheme. The reader is encouraged to examine that paper in greater detail.

IV. Procedure to Bound Capacity of the Network

In this section, we present a procedure to find cuts in the network that give an outer bound on the network capacity for network coding. In later sections, we will see that for the extended butterfly network, this outer bound is achievable by a combination network coding/routing scheme, hence is the capacity of network.

Procedure P1

Step 1 Consider graph G. Find the smallest cut A^1 that separates $S(1)$ from $T(1)$ on G.

Step 2 Consider graph G. Find the smallest cut A^2 that separates $S(2)$ from $T(2)$ on G.

Step 3 Create subgraph $G^1 = (V, E^1)$, where E^1 is the set of links traversed by dedicated or redundant paths.

Step 3.1 Find the smallest cut A_1^3 that separates $S(1)$ from $\{T(1), T(2)\}$ and $S(2)$ from $T(2)$ on G^1.

Step 3.2 Find the smallest cut A_2^3 that separates $S(2)$ from $\{T(1), T(2)\}$ and $S(1)$ from $T(1)$ on G^1.

Step 4 Create subgraph $G^2 = (V, E^2)$, where E^2 is the set of links traversed by shared paths.

Step 4.1 Find the smallest cut A_1^4 that separates $S(1)$ from $\{T(1), T(2)\}$ and $S(2)$ from $T(2)$ on $G^2 \backslash A_1^3$.

Step 4.2 Find the smallest cut A_2^4 that separates $S(2)$ from $\{T(1), T(2)\}$ and $S(1)$ from $T(1)$ on $G \backslash A_2^3$.

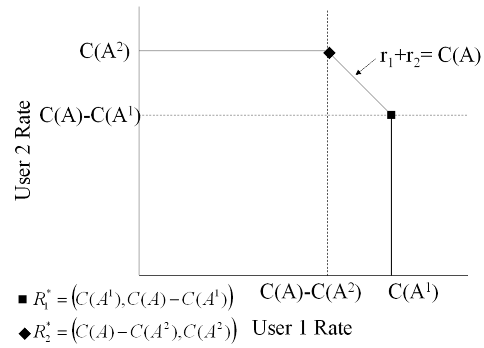

Fig. 5. Outer Bound on Network Capacity

Step 5 Set $A = \begin{cases} A_1^3 \cup A_1^4 & \text{if } C(A_1^3 \cup A_1^4) < C(A_2^3 \cup A_2^4) \\ A_2^3 \cup A_2^4 & \text{else} \end{cases}$.

Figures 3 and 4 show the subgraphs and cuts obtained when using Procedure P1 to find the capacity of the extended butterfly network. From Procedure P1, we have $C(A^1) = 1.3$, $C(A^2) = 1.4$, and $C(A) = \min(2.3, 2.4) = 2.3$.

We now present a theorem showing that this procedure can be used to outer-bound the capacity not only of the extended butterfly network, but of general two-user unicast networks.

Theorem 1: Let R_1 and R_2 be the total information rate conveyed from $S(1)$ to $T(1)$ and from $S(2)$ to $T(2)$, respectively. We denote by Δ the set of rates R_1, R_2 that satisfy the following conditions:

$$R_1 \leq C(A^1) \tag{1}$$

$$R_2 \leq C(A^2) \tag{2}$$

$$R_1 + R_2 \leq C(A) \tag{3}$$

where A^1, A^2, and A are as defined in Procedure P1. Then the capacity region of graph G is a subset of Δ.

Proof: First, consider R_1. From the max-flow min-cut condition (see e.g. [13], Chapter 3), we know that the information rate between a single source and a single sink in a network is outer bounded by the capacity of the smallest cut that separates the source from the sink. Since this is exactly how we have defined A^1, we have $R_1 \leq C(A^1)$. Similarly for the condition $R_2 \leq C(A^2)$.

Next, consider the condition $R_1 + R_2 \leq C(A)$. There are two main steps needed to show that this is an outer bound on the total information rate in a network that employs network coding. First, we show that the set A informationally dominates the source nodes $S(1)$ and $S(2)$. Then we show that this informational dominance implies that the total information rate in the network is outer bounded by $C(A)$.

To see that A informationally dominates the source nodes $S(1)$ and $S(2)$, consider subgraphs G^1 and G^2, and their corresponding edge sets E^1 and E^2. We note that by definition, $E^1 \cup E^2 = E$ and $E^1 \cap E^2 = \rho \cup \varphi$. In other words, the edge sets E^1 and E^2 are disjoint except for the infinite capacity source and destination edges - hence cuts A_1^3 and A_1^4 are completely disjoint. Now, consider subgraph $G \backslash A_1^3 \backslash A_1^4$. By construction of A_1^3 and A_1^4 and using the fact that E^1 and E^2 are disjoint except at the source and destination edges, $T(2)$ is not reachable from a source node. From Fact 2, $T(2) \in Dom(A_1^3 \cup A_1^4)$, hence $S(2) \in Dom(A_1^3 \cup A_1^4)$ on G. Since $S(2) \in Dom(A_1^3 \cup A_1^4)$, and by construction of A_1^3 and A_1^4, $T(1)$ is not reachable from a source node in $G \backslash Dom(A_1^3 \cup A_1^4)$. Again from Fact 2, we have $T(1) \in Dom(A_1^3 \cup A_1^4)$, hence $S(1) \in Dom(A_1^3 \cup A_1^4)$ on G.

Once we have that $\{S(1), S(2)\} \subseteq Dom(A_1^3 \cup A_1^4)$, we have:

$$\begin{aligned} H(A_1^3 \cup A_1^4) &\geq H(S(1), S(2)) \\ &= H(S(1)) + H(S(2)) \\ &\geq (r_1 + r_2) \log_2(b) \end{aligned}$$

where the first inequality comes from Fact 1, the equality comes from an assumption of independent sources, and the last inequality comes from Fact 3. But we also have

$$\begin{aligned} H(A_1^3 \cup A_1^4) &\leq \sum_{l \in (A_1^3 \cup A_1^4)} H(l) \\ &\leq \sum_{l \in (A_1^3 \cup A_1^4)} c_l \log_2(b) \end{aligned}$$

where the first inequality comes from the definition of entropy, and the second inequality comes from Fact 3. Combining these two inequalities, we get:

$$C(A_1^3 \cup A_1^4) = \sum_{l \in (A_1^3 \cup A_1^4)} c_l \geq R_1 + R_2$$

We can follow this same argument using cuts A_2^3 and A_2^4 to get $C(A_2^3 \cup A_2^4) \geq R_1 + R_2$. Since this is an outer bound on $R_1 + R_2$ and we defined A to be the $\arg\min$ of $C(A_1^3 \cup A_1^4)$ and $C(A_2^3 \cup A_2^4)$, then $C(A) \geq R_1 + R_2$ are we are done. ■

Figure 5 shows the region given by Δ for the extended butterfly network.

V. Network Coding Scheme to Achieve Capacity

In this section, we introduce a combination network coding/routing scheme that achieves capacity of the extended butterfly network using linear XOR coding. The procedure for finding the rate assignments that achieve capacity has three main components: use traditional routing to assign rates along dedicated paths (a multi-commodity flow problem), solve a multi-cast network coding problem over the shared and redundant paths, and finally use traditional routing to assign rates along shared paths if there is any "leftover" capacity after solving the multi-cast problem.

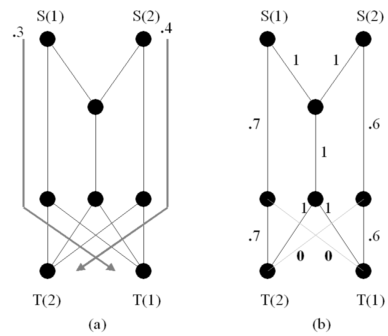

Fig. 6. Step 1: Multi-Commodity Flow Sub-Problem 1

Rate Assignment Procedure RA

Step 1: Solve a Multi-Commodity Flow Problem Construct a subgraph consisting of all dedicated paths - i.e. the paths from each source to its own destination over which network coding cannot occur. Solve the multi-commodity flow problem over this subgraph, as shown in Figure 6(a). Subtract these rates from the capacity of the original network, giving the network shown in Figure 6(b).

Step 2: Solve a Multi-Cast Network Coding Problem Take the network from Figure 6(b) and solve the multi-cast network coding problem, as shown in Figure 7(a). Again, subtract these rates from the capacity of the network from Figure 6(b), giving us the network shown in Figure 7(b).

Step 3: Solve a Second Multi-Commodity Flow Problem The last step is to take the network from Figure 7(b) and solve the multi-commodity flow problem. In this step, users are essentially competing for any capacity on the central link that is leftover after solving the multi-cast network coding problem. For example, we can assign rates as shown in Figure 8(a), where the leftover capacity is given entirely to user 1.

The rates achieved by this solution are $R_1 = .3 + .6 + .4 = 1.3$ and $R_2 = .4 + .6 = 1.0$. Not only is the total rate assignment $R_1 + R_2 = 2.3$ an improvement over both the multi-commodity flow and multi-cast network coding solutions presented in Section II, but this solution exactly achieves the corner point R_1^* from Figure 5. Ultimately, our goal is to show that the outer bound on capacity given by the region Δ from Section IV is achievable, making it the capacity of the network. Since we have shown that it is possible to achieve the first corner point R_1^*, we only need to show that the second corner point R_2^* is achievable in order to show that the entire region Δ is achievable.

In order to do this, we notice that in Step 3 of Procedure RA, user 1 was given all of the leftover capacity along the central link. Alternately, we could have given this capacity to user 2, resulting in the solution $R_1 = .3 + .6 = .9$ and $R_2 = .4 + .6 + .4 = 1.4$. This still gives total rate assignment $R_1 + R_2 = 2.3$, but achieves the corner point R_2^*.

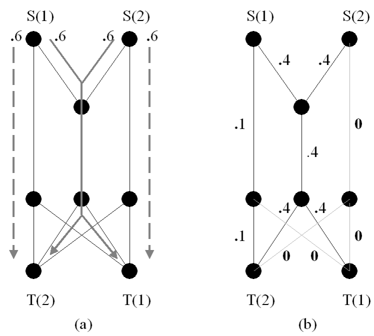

Fig. 7. Step 2: Multi-Cast Network Coding Sub-Problem

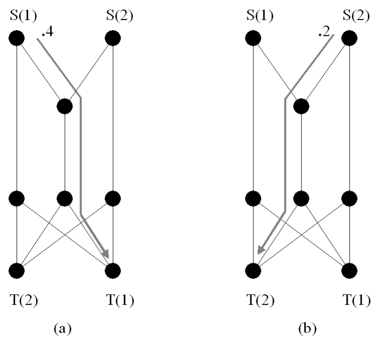

Fig. 8. Step 3: Multi-Commodity Flow Sub-Problem 2

Theorem 2: Any rate vector $(R_1, R_2) \in \Delta$ is achievable, hence Δ is the capacity of the extended butterfly network.

Proof: The rate assignment procedure RA is a constructive argument for showing that the corner points R_1^* and R_2^* are achievable. Any point in Δ can be achieved by an appropriate time-division of these two corner points, and we are done. ■

In addition to achieving capacity, the rate assignment scheme described by Procedure RA has the benefit of being stable in terms of incentives for network coding. In order to see this, recall from Section II that the multi-cast network coding solution used resources along dedicated paths to transmit redundant information for use in decoding at the other user's destination. This is not the case with the scheme achieved by Procedure RA. Since each source is first filling their resources to their own destinations, the multi-cast solution from Step 2 of Procedure RA does not waste any capacity along shared paths! Users are sending redundant information only over paths for which they have no other use. This means that users have no reason to deviate from the network coding scheme, making it a stable solution in terms of incentives for network coding.

VI. Conclusions

In this paper, we explored the use of network coding as method for redistributing resources to achieve capacity the extended butterfly network. For such a network, we show that it is possible to construct a combination network coding/routing scheme that 1) achieves capacity of the network by employing network coding as a way to redistribute network resources, and 2) facilitates cooperation among users by creating a stable rate allocation in which users have no incentive to deviate from the network coding scheme.

There are many interesting and important extensions to this work, and we discuss a few of them briefly. With the exception of the procedure to outer bound capacity of the network, the results presented here are specific to the extended butterfly example. Extension of this scheme to general two-user (and possibly multi-user) networks is an important (although difficult) first step. In addition, the rate assignment scheme presented in this paper is centralized; that is, it assumes that the desired communication rates and capacity of links in the network are fixed and known. Construction of a distributed rate control mechanism will be an important area of future work.

Acknowledgment

The authors would like to thank Prof T. Ho, R. Appuswamy and Prof K. Zeger for their thoughtful discussions. This work was supported by the NSF Career No. CNS-0347961, the ARO-MURI No. W911NF-04-1-0224, the AFOSR No. FA9550-05-01-0430, and CWC at UCSD.

References

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

[2] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2345–2364, June 2006.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct 2003.

[4] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[5] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *Submitted to IEEE Transactions on Information Theory*.

[6] L. Chen, T. Ho, S. Low, M. Chiang, and J. Doyle, "Optimization based rate control for multicast with network coding," in *Proceedings of IEEE Infocom*, May 2007.

[7] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct 2004.

[8] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Proceedings of the Forty-Second Annual Allerton Conference on Communications, Control and Computing*, Sep 2004.

[9] C. Wang and N. Shroff, "Beyond the butterfly - a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2007.

[10] T. Ho, Y. Chang, and K. Han, "On constructie network coding for multiple unicasts," in *Proceedings of the Forty-Fourth Annual Allerton Conference on Control, Communication and Computing*, Sep 2006.

[11] A. Eryilmaz and D. Lun, "Control for inter-session network coding," in *Proceedings of the Workshop on Network Coding, Theory and Applications (NetCod)*, Jan 2007.

[12] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, July 2006.

[13] R. Rockafellar, *Network Flows and Monotropic Optimization*. Athena Scientific, 1998.