Network Coding Games with Unicast Flows

Jennifer Price, *Member, IEEE,* and Tara Javidi, *Member, IEEE*

***Abstract*—To implement network coding, users need to coordinate and cooperate with respect to their strategies in terms of duplicating and transmitting side information across specific parts of the network. In unicast applications where users have no inherent interest in providing (or concealing) their information to (or from) any destinations except for their unique one, this assumption becomes critical in the face of users' autonomy. This paper addresses the issue of cooperation in unicast network coding via a game theoretic approach. Implementation of a given network coding scheme induces a *network coding game* among source-destination pairs (users). In a network with autonomous and rational unicast flows, the equilibrium properties (as well as efficiency) of a network coding scheme is shown to be related to the properties of the corresponding network coding game. In a simple generalization of butterfly networks with two users, we propose a network coding scheme whose capacity achieving operation coincides with users' dominant strategies.**

***Index Terms*—Network coding, game theory, dominant strategies, network capacity**

I. Introduction

Since its inception in the seminal work of Ahlswede, et. al, [1], network coding literature has, for the most part, assumed cooperation among users. In unicast applications where users have no inherent interest in providing (or concealing) their information to (or from) any destinations except for their unique one, the assumption that users will voluntarily participate in a cooperative network coding scheme cannot easily be justified. In this paper, we re-examine the issue of cooperation in networks with selfish source-destination pairs. More precisely, we ask whether a given network coding solution can rationally emerge as an equilibrium among selfish users.

For instance, in a unicast networking scenario an autonomous user may be faced with a choice between using network resources (link capacities) to send new information to its own destination and using these resources to send redundant information for use in decoding at other destinations. In this paper, we show that such choices are induced by the particular network coding scheme that is implemented. These choices, in turn, induce a non-cooperative game among users. In this *network coding game*, the actions and strategy spaces of users are roughly related to the amount of network resources that are used to transfer a given user's information bits to its own destination versus those that enable transfer of redundant/side information (and hence determine the rate of correctly decoded flow at other destinations). As we will show, the network coding scheme implemented by the network significantly impacts the users' transmission strategies at the equilibrium operation of the network. In other words, not only does the network coding scheme determine whether or not the capacity of a network is achieved, but it also determines the equilibrium properties of the network coding game in terms of existence, efficiency, and Pareto optimality!

A common approach to unicast network coding is for intermediate nodes to perform network coding whenever possible [2]. Via a simple generalization of the two user butterfly network, we show that in general this approach is not desirable; it not only fails to achieve capacity (which maybe acceptable in the absence of known/simple capacity achieving schemes), but it may also induce a network coding game with undesirable equilibria. In contrast, we construct a network coding scheme for our example that not only achieves capacity (implying a Pareto optimal solution), but does so in a dominant strategy sense. In other words, the proposed network coding scheme ensures that the desirable capacity achieving solution emerges as a dominant strategy equilibrium point of the game. The main contribution of our paper is to generalize this simple example to show that in some class of two-user networks, which we call *generalized butterfly networks*, there does exist a network coding scheme with a simple strategy space which not only achieves capacity of the network, but also ensures the emergence of the solution at the equilibrium. More precisely, we show that there exists a network coding scheme implemented by the network which guarantees that participating in a capacity achieving network coding strategy is a dominant strategy for all users when they are given autonomy in choosing their transmission strategies.

The remainder of the paper is organized as follows. In Section II, we briefly contextualize our work in the larger body of literature on network coding. In Section III, we provide an example that illustrates the main contribution of the pape, and in Section IV, we generalize our example to a class of *generalized butterfly networks*. In Section V, we conclude the paper: viewing our current paper as only a (small) first step towards understanding the issues of cooperation in network coding, we suggest numerous areas of future research.

II. Related Work

In recent years, network coding has become a major focus of research in network information theory due to its potential applications in communication networks. For brevity we do not list many important works on network coding and game theory, but mention those, closely related to this work.

The material in this paper was presented in part at the Allerton Conference on Communication, Control and Computing, Monticello, IL, September 2007. This work was supported in part by the National Science Foundation CAREER Award No. CNS-0347961, the ARO MURI Grant No. W911NF-04-1-0224, the AFOSR Grant No. FA9550-05-01-0430, and the Center for Wireless Communications at UCSD.

Jennifer Price is with the Department of Electrical and Computer Engineering at the University of Colorado at Colorado Springs, CO 80918, USA (email: jprice@eas.uccs.edu).

Tara Javidi is with the Department of Electrical and Computer Engineering at the University of California San Diego, La Jolla, CA 92093, USA (email: tara@ece.ucsd.edu).

The issue of cooperation and competition in the context of multicast with network coding has recently received attention [3]-[8]. The single-source min-cost multicast problem, framed as a convex optimization problem with the use of network codes and convex increasing edge costs is considered in [3], [4]. A decentralized approach to this problem is presented for the case where all users cooperate to reach the global minimum. Further, the cost for the scenario where each of the multicast receivers greedily routes its flows is analyzed and the existence of a Nash equilibrium is proved. This was later extended in [8], where pricing of edge resources is used in order to construct a selfish flow-steering algorithm for each receiver. The allocation rule by which edge cost at each edge is allocated to flows through that edge is shown to provide an equilibrium that is globally optimal. In all of these papers, the impact of the network coding solution is abstracted in order to replace classical network capacity inequalities on sum of flow rates with constraints on maximum flow rates. In doing so, these works do not address how the particular choice of network coding scheme can create incentive (or disincentive) for flow sources to participate in a given network coding stream. The notions of game and distributed resource allocation in these works focus on more traditional aspects of flow control, route selection, and cost optimization, but assume that users' participation in the network coding scheme is a given.

The construction of our robust capacity achieving network coding scheme is especially motivated by [9] and [10]. In [9], the construction of unicast network coding solutions where coding is restricted to XOR coding between pairs of flows is considered and addressed. In [10], an outer bound for the unicast capacity region of a three-layer network is derived. This outer bound construction relies on certain flows of information in the network to play the role of *side information*, as used in its distributed source coding context. Combining these approaches, our proposed algorithm divides any unicast session into side, dedicated, and network coded sub-flows while restricting the network coding operation to XOR coding of network-coded sub-flows. The outer-bound derived in Lemma 8 can be viewed as a generalization of outer bounds suggested in [10] and its construction follows the techniques introduced in [11].

III. A SIMPLE EXAMPLE: THE EXTENDED BUTTERFLY NETWORK

In this section, we work with a simple unicast network example in order to illustrate the idea of a non-cooperative network coding game. It is our goal to capture the users' adherence to network coding in the context of non-cooperative, self-interested and rational users. Before providing our example, however, we first note the following important property of the classical *butterfly* network that was first introduced in [1] and is shown in Figure 1. In the classical butterfly network (and its generalization, degree-2 three layer networks [10]) with two unicast flows, the use and utility of side links $(S(1), D_1)$ and $(S(2), D_2)$ is limited to the possible carrying of so-called *side* or redundant information that can only be used in decoding of information[1] from S(2) at T(2) and from S(1) at T(1) (see [10]). As a result, in the classical butterfly network there is little contention over the use of links $(S(1), D_1)$ and $(S(2), D_2)$. In other words, since the network coding scheme is utilizing otherwise unusable side links $(S(1), D_1)$ and $(S(2), D_2)$, there is no disincentive even for noncooperative users to participate in the given network coding scheme. It is exactly this property that we refer to as *robustness to non-cooperation*. Next, we use a simple generalization of the butterfly network to show that not all network coding schemes provide such an incentive for participation (i.e. are robust to non-cooperation).

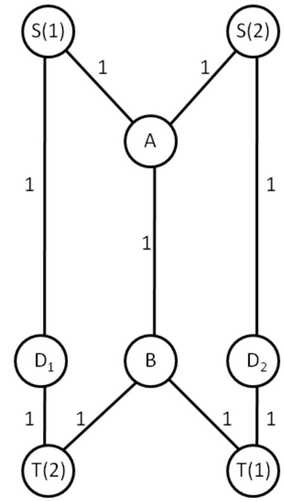


Fig. 1. The Classical Butterfly Network

Figure 2 illustrates a simple generalization of the classical butterfly network consisting of the same two unicast sessions over a network, but in which link $(S(1), D_1)$ $[(S(2), D_2)]$ can be used to carry part of the unicast flow from S(1) to T(1) [from S(2) to T(2)] or to provide side information from S(1) to T(2) [from S(2) to T(1)]. It turns out that whether a rational source $S(i)$ finds it advantageous to use link $(S(i), D_i)$ for transfer of fresh or side information is determined by the network coding scheme as a whole. In other words, the particular network coding scheme employed by node A effects the rational use of side link $(S(1), D_1)$ $[(S(2), D_2)]$ by the unicast source interested in maximizing unicast flow rate between $S(1)$ and $T(1)$ $[S(2)$ and $T(2)]$ - a condition best captured by a *non-cooperative, non-zero-sum game among rational users with common knowledge of the game* [12]. As such, we consider a non-cooperative game in which users make decisions about whether to use network resources for sending new information to their own destination or for sending side information to other destinations.

We formulate the non-cooperative network coding game as a simultaneous move game in which the source nodes are players. The network is modeled as a benevolent manager

[1]The role of side information as it relates to the problem of distributed source coding in a network context was critical in deriving capacity achieving schemes for degree-2 three layer networks in [10].

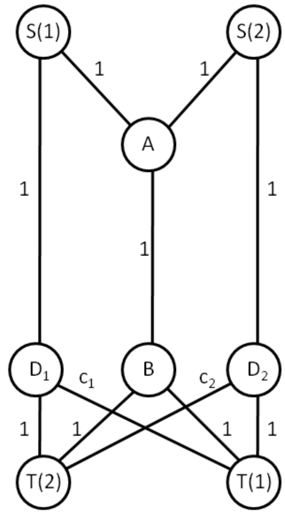


Fig. 2. The Extended Butterfly Network

whose only interest is in achieving capacity of the network. The network's "control knob" is its ability to choose the network coding scheme employed by intermediate nodes. The source nodes are selfish players interested in maximizing their own rate. Here we assume that the network coding scheme employed is fixed and known to the users.

Let us restrict our attention to a simple class of network coding schemes that have the following properties:

1. Network coding is performed only at Node A. All of the other nodes in the network simply forward the information received.
2. Node A splits information flow from S(1) and S(2) into a *network-codable* (NC) segment and a *to-be-routed* (R) segment.
3. Information bits from S(1) and S(2) that are designated as network-codable are coded via a simple XOR scheme at Node A, while information bits designated as to-be-routed are routed according to a conventional (priority-based) routing scheme.

A network coding scheme, then, is the collection of rules according to which 1) substreams are designated as NC or R, and 2) the to-be-routed traffic is prioritized and served.

When the sources are autonomous and rational in their transmission strategies, the network coding scheme is known to the sources, and the scheme is a fixed and deterministic function of the sources' transmission/duplication strategies, then the interaction of the sources is best captured by a *network coding game*. In this game, source nodes select their transmission strategy rationally in anticipation of how the network implements a combination of routing and coding at the intermediate nodes. As we will see, the choice of network coding scheme at the intermediate nodes not only determines the total throughput of the network, but also has a significant impact on the behavior of rational and self-interested users.

A. Game Formulation

Formally, we denote by $\mathcal{F} = [\mathcal{N}, \{\Sigma_i\}_{i=1,2}, \{U_i(\cdot)\}_{i=1,2}]$ the non-cooperative network coding game. The set of players is denoted by the set of flows $\mathcal{N} = \{1, 2\}$, where $S(i)$ and $T(i)$ denote the source and destination nodes for unicast flows $i = 1, 2$. The strategy of player i consists of how it utilizes the capacity of link $(S(i), D_i)$. Mathematically, the strategy (action) of player i is a vector $\sigma_i = (r_i^D, r_i^S) \in \mathbb{R}^2$, which specifies the rate of two flows from node i on its side tree: r_i^D is the flow rate for those bits which are not duplicated in other parts of the network and are routed toward destination $T(i)$ over link $(D_i, T(i))$, while r_i^S is the flow rate for those bits that are routed towards $T(-i)$[2] and are duplicate copies of bits that were sent via link (A,B). Hence, the strategy space of player i is given by[3]

$$\Sigma_i = \{(r_i^D, r_i^S) : r_i^D \leq \min(1, c_i), r_i^S \leq 1, r_i^D + r_i^S \leq 1\}.$$

Since user i is only interested in increasing the rate of reliable/decodable information transfer between $S(i)$ and $T(i)$ (i.e. its unicast flow rate), the utility of user i is defined as:

$$U_i(\sigma_i, \sigma_{-i}) = R_i,$$

where R_i is the total rate of flow of bits from $S(i)$ that can be decoded reliably at $T(i)$. Note that R_i is a function of both users' strategies and the network coding scheme.

Let us consider a simple class of network coding schemes in which coding is limited to XOR coding at Node A. Such a network coding scheme is denoted by a deterministic function $\sigma_0 : \Sigma_1 \times \Sigma_2 \rightarrow \mathbb{R}^3$ such that $r_1^R + r_2^R + r^{NC} \leq 1$. In other words, Node A outputs a network-coded flow of rate $r^{NC} \leq \min\left(r_1^S, r_2^S, 1\right)$ *and* routes bits of information sent from $S(1)$ and $S(2)$ at rates r_1^R and r_2^R, respectively. The capacity constraint of link (A, B) is what imposes the restriction $r_1^R + r_2^R + r^{NC} \leq 1$.

In order to determine the strategic form of the game and the corresponding von Neumann-Morgenstern utility levels [12], we need to specify the deterministic function associated with our network coding scheme. It is our goal in the remainder of this section to illustrate how the choice of network coding scheme σ_0 ultimately determines the rational outcomes of the game. As such, we introduce the following two network coding schemes which we denote by σ_0 and σ_0'.

σ_0: Node A network codes (XOR's) all of the information it receives, regardless of users' strategy. In other words,

$$(r^{NC}, r_1^R, r_2^R) = \sigma_0(r_1^D, r_1^S, r_2^D, r_2^S) = (1, 0, 0).$$

σ_0': Node A network codes (XOR's) the minimum amount of information that is guaranteed to be decodable at the destination (subject to the capacity constraint of link (A, B)). The remaining capacity on link (A, B) i.e. $(1 - r^{NC})$, is split between the

[2]Consistent with notation commonly used in game-theory, we use $-i$ to denote the user(s) that is not i. Hence, if $i = 1$ then $-i = 2$, and vice versa.

[3]Notice that the constraints on r_i^D and r_i^S are a result of link capacity constraints for links in the side-trees.

two users according to a predetermined ratio α. In other words,

$$(r'^{NC}, r_1'^{R}, r_2'^{R}) = \sigma_0'(r_1^D, r_1^S, r_2^D, r_2^S)$$
$$= \left(\min(r_1^S, r_2^S), \alpha(1 - r^{NC}), (1-\alpha)(1 - r^{NC})\right)$$

Intuitively, we can think of the first scheme, σ_0, as a blind network coding scheme; Node A performs network coding whenever it has information bits from both sources. This scheme is motivated by the proposed opportunistic coding schemes in [2]. On the other hand, the second scheme, σ_0', combines routing and coding in a manner such that the output flow consists of both XOR bits as well as original bits that remained uncoded[4]. Additionally, this scheme guarantees some level of fairness in utilizing the bandwidth between incoming flows.

Under each of these schemes, the utility of user i, i.e. the flow rate of bits originated at $S(i)$ and decoded reliably at $T(i)$, is given as:

$$U_i(\sigma_i, \sigma_{-i}) = r_i^D + r_i^R + \min(r_{-i}^s, r^{NC}) \quad \forall\ i = 1, 2.$$

B. Equilibrium Properties of the Non-Cooperative Network Coding Game

We consider the simultaneous non-cooperative network coding game with common knowledge about the network coding scheme. The user strategies are selected simultaneously. The strategy employed by each user is observable by the network coding manager, who then makes network coding decisions according to one of the two schemes described above. As described above, the utility of users is simply the amount of unicast flow rate they receive, while the network manager is modeled as a benevolent manager.

Figure 3 shows the strategic form representation of the game induced by each of the network coding schemes described above when $c_1 = 0.3$, $c_2 = 0.4$, and $\alpha = 0.625$. For illustration purposes, we consider two discrete actions of user i: $(r_i^D, r_i^S) \in \{(c_i, 1 - c_i), (0, 1)\}$ (similar results can be obtained for the continuous space of strategies). We denote action $(r_i^D, r_i^S) = (c_i, 1 - c_i)$ by *Uncoded Bits First* (UBF) and action $(r_i^D, r_i^S) = (0, 1)$ by *Duplicate Bits First* (DBF).

Figure 3 (a) shows the strategic form of the game under the first scheme, σ_0. In this setting, we see a classical prisoner's dilemma situation: the desirable operating point (DBF, DBF), where each user duplicates all of the information sent via link (A, B) to help with decoding at the other destination by selecting action $(r_i^D, r_i^S) = (0, 1)$, is not robust to non-cooperation. In other words, this point cannot emerge as an equilibrium outcome of a game between rational players. On the other hand, the Nash equilibrium point (UBF, UBF) unfairly favors user 2!

Figure 3 (b) shows the strategic form of the game under the second scheme, σ_0'. Here, the action profile (UBF, UBF), where each user i selects action $(r_i^D, r_i^S) = (c_i, 1 - c_i)$ is a dominant strategy equilibrium of the game. In [13] and [14],

[4]This scheme can be thought of a coding scheme with larger alphabet size but we choose to represent the scheme in the above form to capture its salient features vis-à-vis the corresponding network coding game.

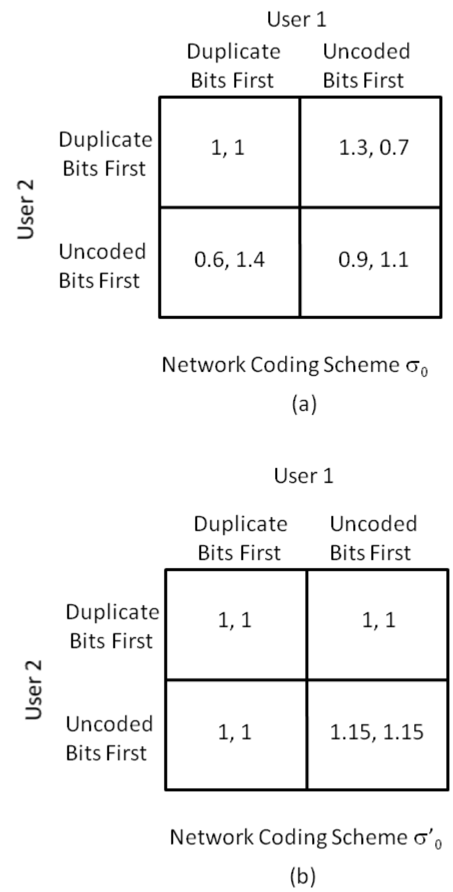


Fig. 3. Strategic Form Representation of the Non-Cooperative Network Coding Game with $c_1 = 0.3$, $c_2 = 0.4$, and $\alpha = 0.625$. For the purpose of illustration, each user is limited to a two-action strategy space, $(r_i^D, r_i^S) \in \{(c_i, 1 - c_i), (0, 1)\}$.

we have shown that the sum-rate capacity of this network is 2.3. Thus, the dominant strategy $(r_i^D, r_i^S) = (c_i, 1-c_i)$ is also Pareto optimal.

From these examples, we see that the the choice of network coding scheme impacts the existence, Pareto optimality, and fairness of the equilibrium!

C. The Impact of Anticipating Non-Cooperation

We can extend the above result to the case where users are allowed the possibility of encoding their messages in anticipation of the other user's non-cooperative strategy. In other words, r_i^S is the flow rate for bits whose conditional entropy given flow r^{NC} is zero. Consider, for example, what happens under scheme σ_0. It is not hard to see that if user i anticipates the possibility of the other user not participating in the network coding by sending insufficient side information to i's destination, i.e. $r_{-i}^S < 1$, then it can transmit its side information in the form of linearly independent combinations of bits. By transmitting carefully chosen linear combinations, some of which are broadcast to both destinations (in anticipating of missing side information bits at a ratio of c_{-i}), user i

User 1

User 2	Duplicate Bits First	Uncoded Bits First
Duplicate Bits First	1, 1	1.3, 1
Uncoded Bits First	0.9, 1.4	0.9, 1.1

Network Coding Scheme σ_0

Fig. 4. Strategic Form Representation of the Non-Cooperative Network Coding Game when Users May Encode their Messages. For the purpose of illustration, each user is limited to a two-action strategy space, $(r_i^D, r_i^S) \in \{(c_i, 1-c_i), (0, 1)\}$.

can deliver a unit rate stream of side information to node $-i$'s destination while providing some new degrees of freedom at its own destination to decode all network coded bits. In other words, the utility of users will be modified to

$$\begin{aligned} U_i(\sigma_i, \sigma_{-i}) &= r_i^D + r_i^R + \min(r_{-i}^s, r^{NC}) \\ &+ \min\left[1 - \min(r_{-i}^s, r^{NC}), \min(c_i - r_i^D, r_i^S)\right] \\ & \forall\ i = 1, 2 \end{aligned}$$

Figure 4 shows the strategic form of the game under this utility model and under scheme σ_0. This figure illustrates that block coding the side stream allows a player to protect itself against non-cooperation to some extent. However, neither of Pareto optimal operating points are rational outcomes of the game. Instead, the dominant strategy equilibrium results in a utility profile of $(.9, 1.1)$; a solution that is neither fair nor capacity achieving. This example reiterates our finding in previous sections: the properties of equilibrium points in the network coding game are strongly a function of utilities associated with certain schemes, and the space of allowable strategies. In addition, this example illustrates the difficulty in finding the optimal strategy to transfer information between a source-destination pair, given other users' and the network's (anticipated) strategies. This underlines the significance of finding network coding scheme σ_0', which ensures that a Pareto optimal capacity achieving network coding scheme emerges as the dominant strategy equilibrium of the network.

In the following section, we extend these results to a class of two user unicast network coding problems which can be viewed as the generalization of our extended butterfly network.

IV. Non-Cooperative Network Coding Game for a Class of 2-User Unicast Networks

In the previous section, we saw that first using resources to transmit their own dedicated information and then using any remaining resources for transmitting side information is a Pareto optimal dominant strategy for players in the extended butterfly network. In this section, we extend these results for a class of generalized 2-user unicast networks. In looking at these generalized networks, a fair amount of notation and technical machinery are needed. The notation and assumptions on network structure are provided below; the basic intuition and game structure, however, closely mimic the example given in Section III.

A. Notation, Definitions and Assumptions on Network Structure

Consider a directed unicast network $G = (V, E)$, where $\mathcal{N} = \{1, 2\}$ is the set of source-destination pairs. Following the convention introduced in [11] (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 all possible paths from any source $S(i)$ to any destination $T(j)$. We denote by p_k the set of links traversed by path k, and set $\psi_{kl} = \begin{cases} 1 & \text{if } l \in p_k \\ 0 & \text{else} \end{cases}$

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: Paths k_1 and k_2 are said to be *edge-connected* if $p_{k_1} \cap pk_2 - \rho - \varphi \neq \phi$

Definition 3: Paths k and l are said to *interact* iff either they are edge-connected, or there exists a sequence of pairwise edge-connected paths $k_1, k_2, \ldots, k_n$ such that k_1 and k_n are edge-connected with paths k and l, respectively.

Definition 4: The set of paths P is a *minimal interacting path group* on graph G if and only if the following conditions are met:

a. $\forall k \in P$, $\nexists \hat{k} \in \mathcal{K} - P$ such that k and $\hat{k}$ are edge-connected,

b. $\forall k_1, k_2 \in P$, k_1 and k_2 are interacting.

In other words, a minimal interacting path group is a set of paths such that 1) no path in minimal interacting path group P shares a (finite capacity) link with a path which is in $\mathcal{K}$ but not in P, and 2) if P contains more than one path, then every pair of paths in P either shares a link, or is "connected" by a sequence of paths in which adjacent paths share at least one link. Furthermore, let $E_P = \{l : l \in p_k \text{ for some } k \in P\}$ denote the set of links traversed by the paths contained in minimal interacting path group P.

Definition 5: A minimal interacting path group P is called *shared* if it includes both sources and both destinations, i.e. $\{S(1), S(2), T(1), T(2)\} \subseteq E_P$. Otherwise, it is called *dedicated.*

Finally, we introduce a lemma regarding the partitioning of paths in $\mathcal{K}$.

Lemma 1: For any network $G = (V, E)$, there exists a unique partition of paths in $\mathcal{K}$ into minimal interacting path groups.

Proof: We note that by their construction, minimal interacting path groups are equivalence classes over sets of paths (the relation defined by "interacting" is reflexive, symmetric, and transitive). We note that the equivalence classes defined by such a relation uniquely partition a set (see e.g. [15], page 3), and we are done. ■

We restrict our attention to a subset of networks which satisfy certain properties, as defined by the assumptions below.

Assumption 1: Let $\mathcal{P} = \{P_1, \ldots, P_M\}$ be the partitioning of graph G into minimal interacting path groups. Then for $\forall m = 1, \ldots, M$ where $|P_m| > 1$, $\exists k_1, k_2 \in P_m$ such that $p_{k_1} \cup p_{k_2} = E_{P_m}$.

Assumption 2: Consider P_m to be a shared minimal interacting path group. P_m does not contain undirected cycles.

From this point on, we will refer to two-user unicast networks that satisfy Assumption 1 and Assumption 2 as *generalized butterfly networks*. Notice that Assumption 1 requires that the set of links traversed by the paths in any given minimal interacting path group either belong to a single path, or can be traversed by exactly two paths. Assumption 2 requires that there are no "loop" structures on the shared path groups. These assumptions are derived from the intuition gained by considering the extended butterfly network from Section III, and allow for the construction of a linear XOR network coding scheme. Intuitively, Assumption 1 guarantees that if two paths interact on a given link, they do not interact with other paths on other links. Hence, the decision of how to utilize the capacity on a given shared link does not affect decisions about how to utilize capacity of shared links traversed by other paths in the network. Assumption 2 simply ensures that there is only one path from $S(i)$ to $T(i)$ over which network coded flows can traverse. These assumptions significantly limit/simplify the structure of the network and allow for a simple XOR/routing-based scheme to achieve capacity.

In the next section, we will show that for all generalized butterfly networks, there exists a network coding scheme which ensures the emergence of a capacity achieving scheme as the dominant strategy equilibrium of the corresponding network coding game. It turns out that in order to precisely construct such a network coding scheme, we need information about certain cuts in the network[5]. To that end, we define the following cuts on the network which will be used extensively both in the main results section below, and in the proofs contained in Appendices B-E.

Let H^1 be the subgraph induced by the set of dedicated path groups in the network, and let H^2 be the subgraph induced by the set of shared path groups in the network. We define the following cuts:

[5]This is similar to the idea that we needed to know the link capacities c_1 and c_2 in order to construct a network coding scheme for the simple generalization of the butterfly example presented in Section III. Since we are considering general two-user networks, however, we need to work with cuts in the network rather than link capacities.

A_i^D	Minimum cut separating $S(i)$ from $T(i)$ on subgraph H^1
A_i^S	Minimum cut separating $S(i)$ from $T(j)$ on subgraph H^1
A_i^T	Minimum cut separating $S(i)$ from $\{T(i), T(j)\}$ on subgraph H^1
B_i^{NC}	Minimum cut separating $S(i)$ from $T(i)$ on subgraph H^2
B^L	Minimum cut separating $S(1)$ from $T(1)$ and $S(2)$ from $T(2)$ on subgraph H^2
B_i^T	Minimum cut separating $S(i)$ from $\{T(i), T(j)\}$ and $S(j)$ from $T(j)$ on subgraph H^2

Notice that the cuts defined above can be categorized into two groups: those defined over dedicated path groups, and those defined over shared path groups. Intuitively, the dedicated path groups can be thought of as side-trees: this is where source nodes can choose to send either side information or fresh bits. The shared path groups can be thought of as butterfly structures: this is where network coding (or traditional routing) may occur. For further discussion, including intuitive explanations of these cuts and their relationship to network coding schemes, see Appendix A.

B. Description of the Network Coding Game

We formulate the non-cooperative network coding game as a simultaneous move game in which the source nodes are players. As with the example in Section III, the network is modeled as a benevolent manager whose only interest is in achieving capacity of the network. The network's "control knob" is its ability to choose the network coding scheme employed by intermediate nodes. The source nodes are selfish players interested in maximizing their own unicast flowrate.

The examples in Section III suggest that the network should employ a coding scheme similar to scheme σ_0'. For our class of generalized butterfly networks, we use a network coding scheme with the following properties:

- P1. We employ only linear XOR coding of previously uncoded information.
- P2. Network coding occurs only over shared path groups.
- P3. The amount of information network coded is equal to the minimum amount of network codable information from the two users.
- P4. Any capacity on shared path groups that remains after network coding occurs is split according to a fixed ratio of α to user 1 and $1 - \alpha$ to user 2.
- P5. These properties are assumed to be fixed and known to all players apriori.

Formally, we denote by $\mathcal{F} = [\mathcal{N}, \{\Sigma_i\}, \{U_i(\cdot)\}]$ the non-cooperative network coding game. The set of players is denoted by $\mathcal{N} = \{1, 2\}$, where $S(i)$ and $T(i)$ are the source and destination nodes for unicast flows $i = 1, 2$. The strategy space of player i consists of how it utilizes the capacity of its cut

A_i^T. In other words, the flow of information from source $S(i)$ over the dedicated path groups can be broken into a stream of duplicated bits at rate r_i^S and an independent stream of bits carrying fresh information bits towards destination $T(i)$ at rate r_i^D. Thus, as in the example in Section III, action $\sigma_i = (r_i^D, r_i^S)$ specifies the rates of flows from source $S(i)$ on its side trees (dedicated path groups): r_i^D is the total of all flow rates for those bits which are not duplicated in other parts of the network and are routed toward destination $T(i)$, while r_i^S is the flow rate for those bits that are routed toward $T(-i)$ and are duplicate copies of information sent over the shared path groups[6]. Mathematically, the strategy space of player i is

$$\begin{aligned}\Sigma_i \;=\; & \{(r_i^D, r_i^S) : r_i^D \le C(A_i^D), r_i^S \le C(A_i^S), \\ & r_i^D + r_i^S \le C(A_i^T)\}\end{aligned} \tag{1}$$

where the cuts A_i^D, A_i^D, and A_i^T are defined in Section IV-A[7].

For all $i = 1, 2$, let $U_i(\sigma_i, \sigma_{-i})$ be the total rate of information originated at $S(i)$ and decoded at destination node $T(i)$. Since user i is only interested in its unicast flow rate, the utility of user i is defined as $U(\sigma_i, \sigma_{-i}) = R_i$, where R_i is the total rate of information that originates at source node $S(i)$ and is decoded at destination node $T(i)$.

As in the example in Section III, the total rate achieved by each user (and hence the rational behavior of each user) depends on how the network (acting as a benevolent manager) implements the network coding scheme. Here, a network coding scheme is parameterized by three quantities:

- Total network coding (XOR) rate, r^{NC},
- Routing capacity of the shared path groups reserved for the uncoded traffic from $S(1)$ to $T(1)$, denoted by r_1^R,
- Routing capacity of the shared path groups reserved for uncoded traffic from $S(2)$ to $T(2)$, denoted by r_2^R.

In other words, the network coding scheme can be described by a mapping:

$$\begin{array}{rccc}\sigma_0 : & \Sigma_1 \times \Sigma_2 & \longrightarrow & \mathbb{R}^3 \\ & (r_1^D, r_1^S, r_2^D, r_2^S) & \longrightarrow & (r^{NC}, r_1^R, r_2^R).\end{array}$$

In particular,

$$r^{NC} = \min(r_1^S + r_1^E, r_2^S + r_2^E, C(B_1^{NC}), C(B_2^{NC})) \tag{2}$$

where

$$r_i^E = C(B_i^T) - C(B^L) \tag{3}$$

and

$$(r_1^R, r_2^R) = \begin{cases} (C(B_1^{NC}) - r^{NC}, C(B^L) - C(B_1^{NC})) \\ \quad \text{if } \alpha\left[C(B^L) - r^{NC}\right] \ge C(B_1^{NC}) - r^{NC} \\ \\ (C(B^L) - C(B_2^{NC}), C(B_2^{NC}) - r^{NC}) \\ \quad \text{if } (1-\alpha)\left[C(B^L) - r^{NC}\right] \ge C(B_2^{NC}) - r^{NC} \\ \\ (\alpha\left(C(B^L) - r^{NC}\right), (1-\alpha)\left(C(B^L) - r^{NC}\right)) \\ \quad \text{Otherwise} \end{cases} \tag{4}$$

Note that

$$r^{NC} \le \min(C(B_1^{NC}), C(B_2^{NC})) \tag{5}$$
$$r^{NC} + r_1^R \le C(B_1^{NC}) \tag{6}$$
$$r^{NC} + r_2^R \le C(B_2^{NC}) \tag{7}$$
$$r^{NC} + r_1^R + r_2^R \le C(B^L) \tag{8}$$
$$r^{NC} + r_1^R + r_2^R + r_i^E \le C(B_i^T) \tag{9}$$

where the inequalities are derived from the cut-set bounds for these rates. For additional explanation of this network coding scheme, the rates involved, and their relationship to the cut-set bounds see Appendix A.

Of course, the obvious questions that arise are 1) can we always construct such a coding scheme, and 2) what are the rates achieved by such a scheme? The next section provides answers to these questions.

C. Main Results: Existence of a Pareto Optimal Dominant Strategy Equlibrium

In this section we address the question of existence and construction of a coding scheme satisfying P1-P5, and the resulting equilibrium properties. It turns out that in the case of generalized butterfly networks, not only can we always construct a network coding solution that satisfies P1-P5 and achieves capacity, but we can do so with a combination of routing and linear XOR codes! To that end, we introduce the following theorem.

Theorem 1: Given a generalized butterfly network, for any pair of user strategies $(\sigma_1, \sigma_2) \in \{\Sigma_1 \times \Sigma_2\}$ there exists a network coding scheme $\sigma_0(\cdot)$ that satisfies Properties P1-P5 and that achieves the unicast rate vector (R_1, R_2) where $R_i = r^{NC} + r_i^D + r_i^R$.

Proof: See Appendix B. ■

Once we know that it is possible to construct a feasible network coding scheme as above, we turn our attention to the incentive structure of the game. If users know that the network will follow the strategy given by $\sigma_0(\cdot)$, how will this impact their choice of transmission strategy (r_i^D, r_i^S)? In order to address this, we introduce the following theorem.

Theorem 2: Let the vector

$$\sigma_i^* := (r_1^{*D}, r_i^{*S}) = (C(A_i^D), C(A_i^T) - C(A_i^D)) \;\forall\; i = 1, 2$$

denote user strategies. Then given a generalized butterfly network, σ_i^* is a dominant strategy for users $i = 1, 2$.

Proof: See Appendix C. ■

Finally, we present a theorem showing that if users follow their dominant strategies, the resulting rate assignment vector actually achieves capacity of the network (hence is Pareto optimal).

Theorem 3: Let σ_i^* be the dominant strategy for $i = 1, 2$ given in Theorem 2, and let $\sigma_0^* = \sigma_0(\sigma_1^*, \sigma_2^*)$. Then given a generalized butterfly network, the rate assignment (R_1, R_2) achieved by the collective actions $(\sigma_0^*, \sigma_1^*, \sigma_2^*)$ achieves capacity of the network, hence is Pareto optimal.

Proof: See Appendix D. ■

The collection of Theorems 1-3 provide a complete picture of a network coding scheme for generalized butterfly networks.

[6]The transmission strategies in this paper are two dimensional vectors describing the rates of flow of freshly generated bits and duplicated bits over dedicated paths. The work can be extended, with cumbersome notations, to the case where the transmission strategies of users are allowed to be the source nodes' coding (in time and space) scheme.

[7]Note that the restriction of transmission rate vectors to $r_i^D \le C(A_i^D)$ and $r_i^S \le C(A_i^S)$ comes from the max-flow min-cut theorem, and does not incur any loss of generality.

They not only provide a capacity achieving network code for generalized butterfly networks, but do so in a manner where the desirable (capacity achieving) solution emerges as a rational (dominant strategy) outcome of the network coding game between the two users. The difficulty in generalizing the above result to the more general network setting involves not only the construction of a capacity achieving solution, but also in the investigation of the space of strategies for users and their incentives in manipulating or participating in the scheme. Next, viewing our strong results above as a first step, we catalogue and motivate the range of interesting problems and future areas of investigation.

V. Future Work and Conclusion

While many simple and opportunistic XOR coding schemes have been advocated in the literature, the (critically practical) issue of robustness of such schemes to selfish behavior has, for the most part, been left unaddressed. In this work, we examined the robustness of given network coding solutions to selfish behavior. More precisely, we considered a context when sources of unicast flow have autonomy over the scheme according to which bits of information are generated and transmitted over different parts of the network. Using a non-cooperative game theoretic formulation, we asked whether or not the desirable solutions emerge as the outcome of the interaction among autonomous and rational unicast flows. From this perspective, a given network coding solution is robust to non-cooperation only if the desirable solution can be sustained, in a dominant strategy or Nash sense, at the equilibrium operation of the corresponding network coding game. In other words, we have shown that efficiency as well as the robustness to the non-cooperative nature of network users for any given network coding scheme is strongly related to the equilibrium properties of the corresponding network coding game. Furthermore, we have shown that for a class of two-user unicast networks, it is possible to construct a network coding scheme for which participation in the network coding scheme is a dominant strategy for all users, and for which the resulting rate assignment is capacity achieving, hence Pareto-optimal. Below, we discuss several other possible extensions of this work.

First, the results presented in this paper are for a class of two-user unicast networks for which it is possible to construct a capacity achieving network coding scheme. In other words, we consider a class of networks for which the utilities of users are known, hence the dominant strategy property can be shown. In [10], a generalized outer bound is shown to characterize the capacity of a degree-two, three-layer networks - a special class of networks that satisfy Assumptions 1 and 2. The capacity achieving scheme for this degree-2 three layer network for two unicast flows is a generalization of the capacity achieving scheme in a classical butterfly network, hence, is a dominant strategy by Theorem 2. An interesting area of future research is an extension of our work to three-layer networks of higher degrees or with more than two unicast flows.

Second, the analysis presented in this paper is based on the notion that the actions of users are observable by the network manager who is capable of making centralized network coding decisions. In a realistic system with distributed implementation, however, this assumption might not hold. In such a case, users may have incentive to declare that they are participating in the network coding scheme without actually doing so. More precisely, since users have no direct use for side information sent to other destinations, what would prevent a user from declaring its packets "network codable" without actually sending the requisite side information for decoding at other destinations? For instance, consider this question in the context of the generalization of the classical butterfly network in Figure 2. In case of scheme σ_0', user 2 has an incentive to "pretend" that it has a side capacity of .4 to transfer side information in order to increase the network coding rate r^{NC} from .6 to .7. This will increase user 2's rate (utility) while decreasing his opponent's utility by the same amount!

Intuitively, this phenomenon can be explained as follows. The capacity achieving network coding schemes in generalized butterfly networks with unicast flows, in effect, provide a means for redistributing and swapping resources (link capacities) which would be left underutilized by a traditional routing solution. In the context of incomplete information, however, these trading schemes are sensitive to users' withholding of information about the traded object(s). As such, designing network coding mechanisms that enforce truth telling as well as cooperation in the network coding scheme is another important topic of future research.

Appendix A
Further Insights and Intuition

In Section IV-A, we introduced a set of cuts over the network. These cuts are chosen specifically for the construction of the proposed capacity achieving network coding scheme and are used to prove Theorems 1-3 in Appendix B. Here, we offer additional details and insights regarding these cuts and their relationship to the network coding scheme. As such, this appendix is solely designed to provide intuition and does not provide any further technical material.

The cuts defined in Section IV-A can be broken into two categories: those that are defined over the subgraph induced by the set of dedicated path groups, and those that are defined over the subgraph induced by the set of shared path groups. Intuitively, the subgraph induced by the set of dedicated path groups can be thought of as a graph of side-trees, over which users have the choice to send either side information or fresh bits. As we will see, the network coding scheme constructed in the next appendix does not perform any network coding over this subgraph. The subgraph induced by the set of shared path groups can be thought of as disjoint butterfly-like structures, over which network coding may occur.

Using this idea of side-trees and butterfly-like structures, the rate assignments for each user can be broken into several components, summarized in the following table.

Dedicated Rate	r_i^D	Amount of information sent from $S(i)$ to $T(i)$ over dedicated path groups
Side Rate	r_i^S	Amount of information sent from $S(i)$ to $T(j)$ over dedicated path groups
Network Coded Rate	r^{NC}	Amount of information sent over shared path groups which is network coded
Routing Rate	r_i^R	Amount of information sent from $S(i)$ to $T(i)$ over shared path groups which is *not* network coded
Extra Side Rate	r_i^E	Amount of information sent from $S(i)$ to $T(j)$ over shared path groups

The rates r_i^D and r_i^S are the rate of information sent over dedicated path groups. Notice that the capacity of the cut A_i^D, defined as the minimum cut separating $S(i)$ from $T(i)$ on subgraph H^1 (the subgraph induced by the dedicated path groups), is the maximum amount of dedicated rate that can be sent by user i. Similarly, the capacity of the cut A_i^S, defined as the minimum cut separating $S(i)$ from $T(j)$ on subgraph H^1, is the maximum amount of side information that can be sent by user i to user j. Finally, the capacity of the cut A_i^T, defined as the minimum cut separating $S(i)$ from $\{T(i), T(j)\}$ on the subgraph H^1, is the maximum amount of information that can be sent from $S(i)$ to either destination. In the case where users first use their available dedicated path group capacity for sending fresh information bits, we have $r_i^D = C(A_i^D)$ and $r_i^S = C(A_i^T) - C(A_i^D)$.

The rates r^{NC}, r_i^R and r_i^E are the rates of information sent over shared path groups[8]. Notice that the capacity of the cut B_i^{NC}, defined as the minimum cut separating $S(i)$ from $T(i)$ on subgraph H^2 (the subgraph induced by the shared path groups), is the maximum amount of information that user i can send either as network coded or routed information.

Notice that the table contains a component called "extra side rate," which is used to distinguished the flow of "side" information to the destination over the shared path groups from the flow of "side" information to the destination over the dedicated path groups. In the simple generalization of butterfly network in Section III, the users were able to fully bottleneck all paths in the shared path groups with either network-coded or routing rate. In some cases, however, it may be possible to send additional side information over a shared path group. Consider, for example, the network shown in Figure 5(a). If network coding occurs over the middle link, each user can send information at rate 1 to its own destination. If we remove this capacity, we are left with the network shown in in Figure 5(b). We see that although there is no remaining path from a source to its own destination, there is a path from user 1's source to user 2's destination! In other words, user 1 can send extra side information at a rate of .5 over this shared path group without changing the amount of network-coded or routing rate that the subgraph can sustain.

[8]Since we employ a network coding scheme in which the amount of information network coded is the minimum of the amount of network coded information of the users, the network coded rate is the same for each user.

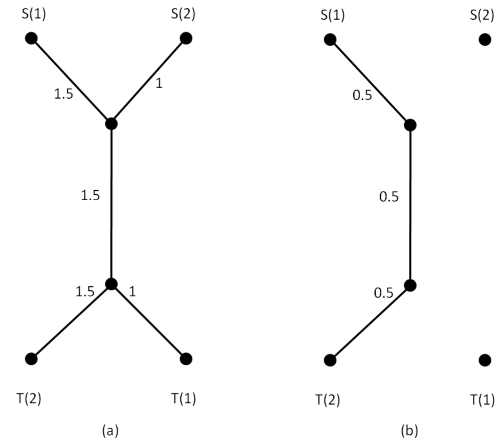


Fig. 5. Illustration of Rate of "Extra Side Information"

Notice that the capacity of cut B^L, defined as the minimum cut separating $S(1)$ from $T(1)$ and $S(2)$ from $T(2)$ on subgraph H^2, is the maximum amount of physical flow (as opposed to information rate) that can be sent from either source to its own destination. The capacity of cut B_i^T, defined as the minimum cut separating $S(i)$ from $\{T(i), T(j)\}$ and $S(j)$ from $T(j)$, is the total amount of information that can be sent from user $S(i)$ to either destination, and from $S(j)$ to $T(j)$. Thus, the amount of extra side rate that user i can transmit is given by $r_i^E = C(B_i^T) - C(B^L)$.

Appendix B
Proof of Theorem 1

The proof of Theorem 1 is constructive; that is, we provide a procedure for assigning rates and network coding operations along individual paths in the network. To do so, we require an alternate enumeration of paths in the network that captures the salient features of the 2-user generalized butterfly network as defined by Assumption 1 and Assumption 2. This alternate enumeration of paths is constructed as follows:

Procedure P1

Let $\mathcal{P} = \{P_1, \ldots, P_M\}$ be the set of minimal interacting path groups for network G, and let $\mathcal{K}_i^D$, $\mathcal{K}_i^S$, $\mathcal{K}_i^{NC}$, $\mathcal{K}_i^E$ $\forall\ i = 1, 2$ be initially empty sets of paths. For each minimal interacting path group $m = 1, \ldots, M$ in $\mathcal{P}$, do the following:

Step P1-1 When $|P_m| = 1$, there is a single path $k \in P_m$. If path k traverses from $S(1)$ to $T(1)$ (from $S(2)$ to $T(2)$), add it to $\mathcal{K}_1^D$ (to $\mathcal{K}_2^D$). If path k traverses from $S(1)$ to $T(2)$ (from $S(2)$ to $T(1)$), add it to $\mathcal{K}_1^S$ (to $\mathcal{K}_2^S$).

Step P1-2 When $|P_m| > 1$ and P_m is a *dedicated* minimal interacting path group, Assumption 1 guarantees that there exists at least one pair of paths k_1 and k_2 such that $p_{k_1} \cup p_{k_2} = E_{P_m}$ [9]. If k_1 traverses from $S(1)$ to $T(1)$ (from $S(2)$ to $T(2)$), add it to $\mathcal{K}_1^D$ (to $\mathcal{K}_2^D$). If path k_1 traverses from $S(1)$ to $T(2)$ (from $S(2)$ to $T(1)$), add it to $\mathcal{K}_1^S$ (to $\mathcal{K}_2^S$). Do the same for path k_2.

Step P1-3 When $|P_m| > 1$ and P_m is a *shared* minimal interacting path group, from Assumption 1 and Assumption 2 we know that there exists exactly one pair of paths k_1 and k_2 such that path k_1 goes from S(1) to T(1), path k_2 goes from S(2) to T(2), and $p_{k_1} \cup p_{k_2} = E_{P_i}$. Add k_1 to $\mathcal{K}_1^{NC}$ and k_2 to $\mathcal{K}_2^{NC}$. Similarly, there is exactly one pair of paths $\hat{k}_1$ and $\hat{k}_2$ such that path $\hat{k}_1$ goes from $S(1)$ to $T(2)$ and $\hat{k}_2$ goes from $S(2)$ to $T(1)$. Add $\hat{k}_1$ to $\mathcal{K}_1^E$ and $\hat{k}_2$ to $\mathcal{K}_2^E$.

Step P1-4 Set $\mathcal{K}_1 = \{\mathcal{K}_1^D\ \mathcal{K}_1^S\ \mathcal{K}_1^{NC}\ \mathcal{K}_1^E\}$ and $\mathcal{K}_2 = \{\mathcal{K}_2^D\ \mathcal{K}_2^S\ \mathcal{K}_2^{NC}\ \mathcal{K}_2^E\}$ and $F^1 = \{p_k : k \in (\mathcal{K}_1^D \cup \mathcal{K}_2^D \cup \mathcal{K}_1^S \cup \mathcal{K}_2^S)\}$ and $F^2 = \{p_k : k \in (\mathcal{K}_1^{NC} \cup \mathcal{K}_2^{NC})\}$.

We now introduce the following lemma, which delineates important properties of this enumeration of links. We note that it is exactly these properties that Assumption 1 and Assumption 2 are designed to ensure.

Lemma 2: Let $\mathcal{K}_i^D$, $\mathcal{K}_i^S$, $\mathcal{K}_i^{NC}$, $\mathcal{K}_i^E$, F^1, and F^2 be as constructed above. Then these sets satisfy the following properties:

$$F^1 \cup F^2 = E, \tag{10}$$

$$F^1 \cap F^2 = \rho \cup \varphi, \tag{11}$$

$$\left(\cup_{k \in \mathcal{K}_1^D} p_k\right) \cap \left(\cup_{k \in \mathcal{K}_2^D} p_k\right) \subseteq \rho \cup \varphi, \tag{12}$$

$$\left(\cup_{k \in \mathcal{K}_1^S} p_k\right) \cap \left(\cup_{k \in \mathcal{K}_2^S} p_k\right) \subseteq \rho \cup \varphi, \tag{13}$$

$$\forall k_1 \in \mathcal{K}_1^{NC}\ \exists \text{ exactly one } k_2 \in \mathcal{K}_2^{NC} \text{ such that } (p_{k_1} \cap p_{k_2}) \supset (\rho \cup \varphi). \tag{14}$$

Proof: Procedure P2 is a constructive proof of this theorem for networks satisfying Assumptions 1-2. ■

We now turn to the construction of a network-coding/rate-assignment scheme. Notice that not only does the enumeration of paths given by Procedure P2 capture all of the relevant information about network G, but it has a natural structure in terms of our desired network-coding/rate-assignment scheme. In particular, the sets $\mathcal{K}_i^D$ contain paths along which users transmit "dedicated" information to their own destination, and along which network coding does not occur. The sets $\mathcal{K}_i^S$ contain paths along which users can at best transmit "side" information to other destinations for use in decoding. The sets $\mathcal{K}_i^{NC}$ contain paths along which users transmit information to their own destination, but along which network coding may occur. Finally, the sets $\mathcal{K}_i^E$ contain additional paths along which side information can be transmitted for use in decoding.

The procedure for finding the rate assignments consists of several steps in which rates are allocated to individual paths in the network. We denote by r_{ik} the rate sent from source i along path $k \in \mathcal{K}_i$. The fact that the strategy spaces defined in Section IV are constrained by certain cut set bounds is crucial to being able to construct these rate assignments along individual paths.

[9]If there exists more than one such pair, arbitrarily choose any single pair.

Rate Assignment Procedure R

Step R 1: Assign Dedicated Rates

Assign rates r_{ik} along every path $k \in \mathcal{K}_i^D$ such that

$$\sum_{k \in \mathcal{K}_i^D} r_{ik} \psi_{kl} \le C_l,\ \forall l \in \mathcal{L} \tag{15}$$

and

$$\sum_{k \in \mathcal{K}_i^D} r_{ik} = r_i^D,\ \forall i = 1, 2. \tag{16}$$

Notice that (12) combined with the strategy space restriction given by (1) ensures it is possible to assign dedicated rates in a way that satisfies (15).

Step R 2: Assign Side Information Rates

Assign rates r_{ik} along every path $k \in \mathcal{K}_i^S$ such that

$$\sum_{i=1}^{N} \left[\sum_{k \in \mathcal{K}_i^D} r_{ik} \psi_{kl} + \sum_{k \in \mathcal{K}_i^S} r_{ik} \psi_{kl} \right] \le C_l \tag{17}$$

and

$$\sum_{k \in \mathcal{K}_i^S} r_{ik} = r_i^S. \tag{18}$$

Notice that (12) and (13) combined with the strategy space restriction given by (1) ensures it is possible to assign side rates in a way that satisfies (17).

Step R 3: Assign Network Coded Rates

Recall from (14) that for each path $k_1 \in \mathcal{K}_1^{NC}$, there exists exactly one path $k_2 \in \mathcal{K}_2^{NC}$ such that k_1 and k_2 share at least one link. For each such pair, assign network coding rates rates $r'_{1k_1} = r'_{2k_2}$ such that $\forall l \in \mathcal{L}$

$$\max\left(\sum_{k \in \mathcal{K}_1^{NC}} r'_{1k} \psi_{kl}, \sum_{k \in \mathcal{K}_2^{NC}} r'_{2k} \psi_{kl} \right) \le C_l \tag{19}$$

and

$$\sum_{k \in \mathcal{K}_1^{NC}} r'_{1k} = \sum_{k \in \mathcal{K}_2^{NC}} r'_{2k} = r^{NC}, \tag{20}$$

where the value of r^{NC} is chosen according to (2). Note that, due to XOR coding at the intermediate nodes, it is the *physical flow*, given by $\max\left(\sum_{k \in \mathcal{K}_1^{NC}} r'_{1k}\psi_{kl}, \sum_{k \in \mathcal{K}_2^{NC}} r'_{2k}\psi_{kl}\right)$ rather than the *information flow*, given by $\sum_{k \in \mathcal{K}_1^{NC}} r'_{1k}\psi_{kl} + \sum_{k \in \mathcal{K}_2^{NC}} r'_{2k}\psi_{kl}$ that must satisfy the link capacity constraints. Furthermore, since $r^{NC} \le \min\left[C(B_1^{NC}), C(B_2^{NC})\right]$, it is always possible to construct rates that satisfy (19).

Step R 4: Assign Routing Rates

Assign rates r_{ik} along every path $k \in \mathcal{K}_1^{NC}$ for each user $i = 1, 2$ such that

$$\sum_{i=1}^{2} \left(\sum_{k \in \mathcal{K}_i^{NC}} r_{ik}\psi_{kl} \right) + \max \left(\sum_{k \in \mathcal{K}_1^{NC}} r'_{1k}\psi_{kl}, \sum_{k \in \mathcal{K}_2^{NC}} r'_{2k}\psi_{kl} \right) \leq C_l, \ \forall l \in \mathcal{L}, \quad (21)$$

and

$$\sum_{k \in \mathcal{K}_i^{NC}} r_{ik} = r_i^R, \ \forall \ i = 1, 2, \quad (22)$$

where r_i^R is chosen according to (4). Since $r^{NC} + r_i^R \leq C(B_i^{NC}) \ \forall \ i = 1, 2$ and $r^{NC} + r_1^R + r_2^R \leq C(B^L)$, it is always possible to construct rates that satisfy (22).

Step R 5: Assign Extra Rates

Assign rates r_{ik} along every path $k \in \mathcal{K}_i^E$ such that

$$\sum_{i=1}^{2} \left[\sum_{k \in \mathcal{K}_i^{NC} \cup \mathcal{K}_i^E} r_{ik}\psi_{kl} \right] + \max \left(\sum_{k \in \mathcal{K}_1^{NC}} r'_{1k}\psi_{kl}, \sum_{k \in \mathcal{K}_2^{NC}} r'_{2k}\psi_{kl} \right) \leq C_l, \ \forall l \in \mathcal{L}, \quad (23)$$

and

$$\sum_{k \in \mathcal{K}_i^E} r_{ik} = r_i^E. \quad (24)$$

Note that the construction of $r_i^E = C(B_i^T) - C(B^L)$ ensures that it is always possible to assign rates that satisfy (24).

By construction, the rate assignment procedure R satisfies the capacity constraints of the network. All that is left to prove Theorem 1 is to show that the information received at the destination nodes can be correctly decoded. Since we are only using linear XOR codes on previously uncoded information, the only requirement for successful decoding at the destination node is that the amount of side information received is at least as much as the amount of information that was coded. But this is guaranteed by the choice of r^{NC} given by (2). Hence, each source receives total rate $R_i = r_i^D + r_i^{NC} + r_i^R$, and we are done.

APPENDIX C
PROOF OF THEOREM 2

Without loss of generality, we fix the strategy of user 2 and consider what happens as user 1 varies its strategy. We break the proof of Theorem 2 into the following lemmas:

- User 1 cannot increase its utility by setting $r_1^D > C(A_1^D)$.
- User 1 cannot increase its utility by setting $r_1^S > C(A_1^T) - C(A_1^D)$.
- User 1 cannot increase its utility by setting $r_1^D < C(A_1^D)$.
- User 1 cannot increase its utility by setting $r_1^S < C(A_1^T) - C(A_1^D)$.

Lemma 3: User 1 cannot increase its utility by setting $r_1^D > C(A_1^D)$.

Proof: This comes from the definition of the strategy space of users, in which $r_i^D \leq C(A_i^D)$. ■

Lemma 4: User 1 cannot increase its utility by setting $r_1^S > C(A_1^T) - C(A_1^D)$.

Proof: Let $r_1^{S*} = C(A_1^T) - C(A_1^D)$, and let $\hat{r_1^S} = r_1^{S*} + \Delta$ where $\Delta > 0$. From (2), we know that $r^{\hat{NC}} \leq r^{NC*} + \Delta$. But we also know that the side and dedicated rates must satisfy the cut set bound $r_1^D + r_1^S \leq C(A_1^T)$. Hence, $\hat{r_1^D} \leq r_1^{D*} - \Delta$. This gives:

$$\hat{r_1^D} + r^{\hat{NC}} \leq r_1^{D*} - \Delta + r^{NC*} + \Delta = r_1^{*D} + r^{NC*}$$

We only have left to show that a user cannot improve its routing rate $\hat{r_1^R}$ above rate r_1^{R*}. To do, recall from (4) that the formula for r_1^R can take on one of three forms.

First, we consider the case when $\alpha \left[C(B^L) - r^{NC*} \right] \leq C(B_1^{NC*}) - r^{NC*}$ and $(1 - \alpha) \left[C(B^L) - r^{NC*} \right] \leq C(B_2^{NC*}) - r^{NC*}$. Since $\hat{r_1^S} > r_1^{S*}$, then $r^{\hat{NC}} \geq r^{NC*}$, which in turn means $C(B^L) - r^{\hat{NC}} \leq C(B^L) - r^{NC*}$. In other words, if $\underline{r}^{R*}$ is assigned according to (4), then so will $\hat{\underline{r}^R}$. But

$$\hat{r_1^R} = \alpha(C(B^L) - r^{\hat{NC}}) \leq \alpha(C(B^L) - r^{NC*}) = r_1^{R*}$$

Next, consider the case when $\alpha \left[C(B^L) - r^{NC*} \right] \geq C(B_1^{NC*}) - r^{NC*}$. Here $\underline{r}^{R*}$ is assigned according to (4). Notice, however, that $r_1^{R*} = C(B_1^{NC}) - r^{NC*}$ satisfies the cut set bound $r_1^R + r^{NC} \leq C(B_1^{NC})$. Thus

$$\hat{r_1^R} \leq C(B_1^{NC}) - r^{\hat{NC}} = r_1^{R*} + r^{NC*} - r^{NC*} < r_1^{R*}$$

Finally, consider the case when $(1 - \alpha) \left[C(B^L) - r^{NC*} \right] \geq C(B_2^{NC*}) - r^{NC*}$, hence routing rates are assigned according to (4). If $(1 - \alpha) \left[C(B^L) - r^{NC} \right] \geq C(B_2^{NC}) - r^{\hat{NC}}$ - i.e. the change in network coding rate does not change the fact that user 2 cannot use its designated portion of the leftover capacity, then the routing rate of user 1 stays exactly the same. If, however, the change in network coding rate means that user 2 can now use its designated portion of the leftover capacity - i.e. since there is less leftover capacity the routing rates are now assigned according to (4) - then we have the following:

$$\hat{r_1^R} = \alpha(C(B^L) - r^{\hat{NC}}) \leq \alpha(C(B^L) - r^{NC*}) \quad (25)$$

But, from the assumption that $\underline{r}^{R*}$ was assigned according to (4), we have

$$(1 - \alpha) \left[C(B^L) - r^{NC*} \right] \geq C(B_2^{NC*}) - r^{NC*}$$

Rearranging gives

$$\alpha(C(B^L) - r^{NC*}) \leq C(B^L) - C(B_2^{NC}) = r_1^{R*}$$

Substituting into (25) gives $\hat{r_1^R} \leq r_1^{R*}$, and we are done. ■

Lemma 5: User 1 cannot improve its utility by setting $r_i^D < C(A_i^D)$.

Proof: Since the value of r_i^D only plays a role in the network coding scheme insofar as it determines how much side information can be sent, decreasing the value of r_i^D can only improve the utility of a user if increasing the side information beyond $C(A_i^T) - C(A_i^D)$ can improve the users' utility. From Lemma 4, we know that this is not true, hence we are done. ■

Lemma 6: User 1 cannot increase its utility by setting $r_1^S < C(A_1^T) - C(A_1^D)$.

Proof: Since user 1 can send its min-cut max-flow value $r_i^D = C(A_1^D)$ while also sending $r_1^S = C(A_1^T) - C(A_1^D)$, the user cannot send any additional dedicated rate even if it decreases its side rate. On the other hand, if $\hat{r_1^S} < r_1^{S*}$, then $r^{\hat{NC}} \leq r^{NC*}$. If the network coding rate is decreased, then the leftover capacity that can be split between the users is increased by that same amount. At best, user 1 can acquire all of that capacity. Even if user 1 acquires *all* of the additional leftover capacity (which it may not), that still only exactly offsets the loss in network coding rate. Thus, we have:

$$r^{\hat{NC}} + \hat{r_1^R} \leq r^{NC*} + \hat{r_1^R}$$

Hence, the rate of user 1 cannot be improved. ■

Since the above lemmas delineate all of the ways in which a user could deviate from the strategy σ_i^*, and by symmetry of users 1 and 2, we have the proof for Theorem 2.

Appendix D
Proof of Theorem 3

The proof of Theorem 3 consists of two main components. First, we develop an outer bound for the capacity region of two-user unicast networks using cut-set bounds. Then, we show that the rate assignment scheme constructed in Appendix B achieves the outer bound for generalized butterfly networks by relating these cut-sets to the the cut-sets introduced in Section IV-A. The proofs use the notion of informational dominance in graphs, introduced in [11]. For completeness, we have included useful results from [11] in Appendix E.

In order to facilitate the proof of Theorem 3, we first introduce the following procedure for finding cuts in the network that will give the outer bound on the network capacity for network coding.

Procedure P2

- **Step P2-1** Consider graph G. Find the smallest cut A^1 that separates $S(1)$ from $T(1)$ on G.
- **Step P2-2** Consider graph G. Find the smallest cut A^2 that separates $S(2)$ from $T(2)$ on G.
- **Step P2-3** Create subgraph $G^1 = (V, E^1)$, where $E^1 = \bigcup_{P_m \in P^D} (E_{P_m})$.
  - **Step P2-3.1** 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 P2-3.2** 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 P2-4** Create subgraph $G^2 = (V, E^2)$, where $E^2 = \bigcup_{P_m \in P^S} (E_{P_m})$.
  - **Step P2-4.1** 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 P2-4.2** Find the smallest cut A_2^4 that separates $S(2)$ from $\{T(1), T(2)\}$ and $S(1)$ from $T(1)$ on $G^2 \backslash A_2^3$.
- **Step P2-5** Set $A = \begin{cases} A_1^3 \cup A_1^4 & \text{if } C(A_1^3 \cup A_1^4) < C(A_2^3 \cup A_2^4) \\ A_2^3 \cup A_2^4 & \text{else} \end{cases}$

We now introduce a lemma to show that these cuts can be used to outer bound the capacity of the network. The proof is based on the informational dominance of cut sets A_1 , A_2, and A as it is introduced in [11] and stated in Appendix E.

Lemma 7: 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. Rates R_1, R_2 are achievable only if they satisfy the following conditions:

$$R_1 \leq C(A^1) \tag{26}$$

$$R_2 \leq C(A^2) \tag{27}$$

$$R_1 + R_2 \leq C(A) \tag{28}$$

where A_1, A_2, and A are as defined above. Then the capacity region of the network is a subset of $\Delta := \{(R_1, R_2) :$ rates R_1, R_2 satisfy (26)-(28)$\}$.

Proof: First, consider (26) and (27). From the max-flow min-cut condition (see e.g. [16], 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 3 in Appendix E, $T(2) \in Dom(A_1^3 \cup A_1^4)$, 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 3 in Appendix E, 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:

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

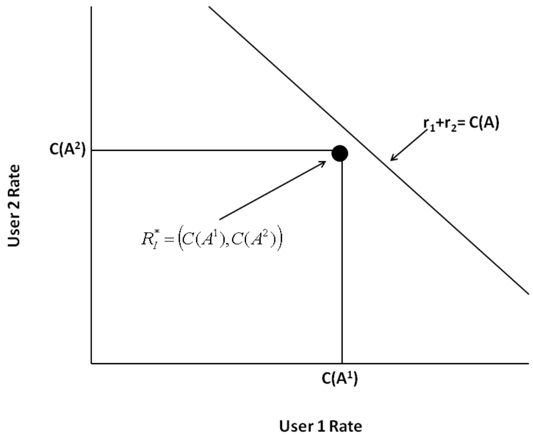


Fig. 6. Upper Bound on Network Capacity: Version I

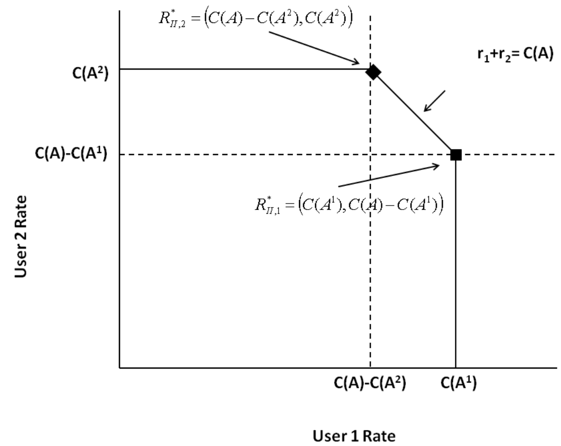


Fig. 7. Upper Bound on Network Capacity: Version II

where the first inequality comes from Fact 2 in Appendix E, the equality comes from an assumption of independent sources, and the last inequality comes from Fact 1 in Appendix E. But we also have

$$H(A_1^3 \cup A_1^4) \leq \sum_{l \in (A_1^3 \cup A_1^4)} H(l) \leq \sum_{l \in (A_1^3 \cup A_1^4)} c_l \log_2(b),$$

where the first inequality comes from the definition of entropy, and the second inequality comes from Fact 1 in Appendix E. 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 $C(A) \geq R_1 + R_2$ and we are done. ■

Figures 6 and 7 show the two possible shapes the region given by Δ can take. Figure 6 shows the case when the line $R_1 + R_2 = C(A)$ does not intersect the lines $R_1 = C(A^1)$ and $R_2 = C(A^2)$. This shape results, for example, if user i has a set of paths along which it can only send information from $S(i)$ to $T(j)$, and which have much larger capacities than the paths along which user i can send information from $S(i)$ to $T(i)$. Figure 7 shows the case when the line $R_1 + R_2 = C(A)$ intersects the lines $R_1 = C(A^1)$ and $R_2 = C(A^2)$. If Δ takes on this shape, there may be a multi-user gain that can be achieved with network coding.

Next, we introduce the following lemmas to show that the corner points of the region Δ is achievable. From standard time-sharing argument, the achievability of corner points implies the achievability of the whole Δ region.

Lemma 8: Let (R_1^A, R_2^A) be the rate assignment given by the rate assignment procedure in Appendix B when $\alpha = 1$. Then

$$R_1^A = C(A^1),$$

and

$$R_2^A = \begin{cases} C(A^2) & \text{if } C(A) \geq C(A^1) + C(A^2) \\ C(A) - C(A^1) & \text{if } C(A) < C(A^1) + C(A^2) \end{cases}$$

where A^1 and A^2 are the cuts defined in Procedure P2.

Proof: Each user is assigned four types of rates: dedicated, side, network coded, and routed. The side information assigned to each user is only useful for decoding at the other users' destination, hence each users' achieved rate is $r_i^D + r^{NC} + r_i^R$. We break the proof into two cases: the case when the network coding rate is limited by the shared capacity, and the case when the network coding rate is limited by the side capacity. As we will see, these two cases correspond to the two possible shapes of the feasibility regions discussed earlier.

<u>Case 1</u>: $\min(r_1^T, r_2^T, r_1^S + r_1^E, r_2^S + r_2^E) \in \{r_1^T, r_2^T\}$

First, assume that $\min(r_1^T, r_2^T, r_1^S + r_1^E, r_2^S + r_2^E) = r_1^T$, giving each user a network coding rate of r_1^T. User 1 receives no additional routing rate since its network coding rate r_1^T bottle-necks the paths from $S(1)$ to $T(1)$. This gives

$$R_1^A = r_1^D + r^{NC} + r_1^R = r_1^D + r_1^T = C(B_1^{1D}) + C(B_1^{2T}).$$

Recall that B_1^{1D} is the minimum cut that separates $S(1)$ from $T(1)$ on subgraph $H^1 = (V, F^1)$, and B_1^{2T} is the minimum cut that separates $S(1)$ from $T(1)$ on subgraph $H^2 = (V, F^2)$. This implies that $C(B_1^{1D}) + C(B_1^{2T})$ is the capacity of the minimum cut that separates $S(1)$ from $T(1)$ on graph $G = (V, E)$. This is exactly the definition of cut A^1 from Step P1-1 of Procedure P2. Hence, we have $R_1^A = C(A^1)$.

User 2 also receives network coding rate r_1^T. The additional routing rate that user 2 receives is equal to $r_2^T - r_1^T$ (the difference between the total rate that can be sent from $S(2)$ to $T(2)$ and the rate that user 2 is already sending via network coding)[10]. This gives

$$\begin{aligned} R_2^A &= r_1^D + r^{NC} + r_2^R \\ &= r_1^D + r_1^T + r_2^T - r_1^T = r_2^D + r_2^T \\ &= C(B_2^{1D}) + C(B_2^{2T}). \end{aligned}$$

Again, recall that B_2^{1D} is the minimum cut that separates $S(2)$ from $T(2)$ on subgraph $H^1 = (V, F^1)$, and B_2^{2T} is the minimum cut that separates $S(2)$ from $T(2)$ on subgraph $H^2 = (V, F^2)$. This means that $C(B_2^{1D}) + C(B_2^{2T})$ is the capacity of the minimum cut that separates $S(2)$ from $T(2)$ on graph $G = (V, E)$. This is exactly the definition of cut A^2 from Step P1-2 of Procedure P1. Hence, we have $R_2^A = C(A^2)$, and hence the assertion of the lemma. By symmetry, we have the same result when $\min(r_1^T, r_2^T, r_1^S + r_1^E, r_2^S + r_2^E) = r_2^T$.

Case 2: $\min(r_1^T, r_2^T, r_1^S + r_1^E, r_2^S + r_2^E) \in \{r_1^S + r_1^E, r_2^S + r_2^E\}$

Recall that $C(B_1^{1S}) = r_1^D + r_1^S$, where B_1^{1S} is the minimum cut that separates $S(1)$ from $\{T(1), T(2)\}$ on subgraph $H^1 = (V, F^1)$, and $C(B_2^{1D}) = r_2^D$. However, $C(B_1^{1S}) + C(B_2^{1D})$ is the capacity of the minimum cut that separates $S(1)$ from $\{T(1), T(2)\}$ and $S(2)$ from $T(2)$ on subgraph $G^1 = (V, E^1)$. This is exactly the definition of cut A_1^3 from Step P1-3.1 of Procedure P1, hence

$$C(A_1^3) = r_1^D + r_2^D + r_1^S. \tag{29}$$

Furthermore, recall that $C(B_1^{2E}) = r_1^T + r_2^L + r_1^E$, where B_1^{2E} is the minimum cut that separates $S(1)$ from $\{T(1), T(2)\}$ and $S(2)$ from $T(2)$ on subgraph $H^2 = (V, F^2)$. So, again, w$C(B_1^{2E})$ is the capacity of the minimum cut separating $S(1)$ from $\{T(1), T(2)\}$ and $S(2)$ from $T(2)$ on subgraph $G^2 = (V, E^2)$. This is exactly the definition of cut A_1^4 from Step P1-4.1 of Procedure P1, hence

$$C(A_1^4) = r_1^T + r_2^L + r_1^E. \tag{30}$$

By symmetry, we also have

$$C(A_2^3) = r_1^D + r_2^D + r_2^S. \tag{31}$$

$$C(A_2^4) = r_2^T + r_1^L + r_2^E. \tag{32}$$

Notice that the quantities $r_1^T + r_2^L$ and $r_2^T + r_1^L$ are the maximum amount of information that can be sent from $S(1)$ to $T(1)$ and from $S(2)$ to $T(2)$ on subgraph H^2. In other words, the points $r_1^T + r_2^L$ and $r_2^T + r_1^L$ are different points on the same line with sum-rate equal to the multi-commodity flow capacity of subgraph H^2. This gives

$$r_1^T + r_2^L = r_2^T + r_1^L.$$

Now, evaluating $C(A)$, we have

$$\begin{aligned} C(A) &= \min(C(A_1^3) + C(A_1^4), C(A_2^3) + C(A_2^4)) \\ &= \min(r_1^D + r_2^D + r_1^S + r_1^T + r_2^L + r_1^E) \\ &= \min(r_1^D + r_2^D + r_2^S + r_2^T + r_1^L + r_2^E) \\ &= r_1^D + r_2^D + r_1^T + r_2^L + \min(r_1^S + r_1^E, r_2^S + r_2^E) \\ &= r_1^D + r_2^D + r_1^T + r_2^L + r^{NC} \end{aligned}$$

[10]Since all of the rate sent by user 1 is network coded, it does not interfere with information sent by user 2. Hence user 2 is able to achieve the max-cut min-flow rate corresponding to r_2^T.

All that is left is to relate the rates (R_1^A, R_2^A) to the capacity of cuts A^1 and A. To do so, recall that, when $\alpha = 1$, in Step RA-3, user 1 receives as much additional routing rate as it takes to bottle-neck the paths from $S(1)$ to $T(1)$: hence $r_1^R = r_1^T - r^{NC}$. This gives

$$\begin{aligned} R_1^A &= r_1^D + r^{NC} + r_1^R \\ &= r_1^D + r^{NC} + r_1^T - r^{NC} \\ &= r_1^D + r_1^T = C(A^1). \end{aligned}$$

Note that the last equality was proven in Case 1.

User 2 also receives as much additional routing rate as it takes to bottle-neck the paths from $S(2)$ to $T(2)$, defined as r_2^L. This gives

$$\begin{aligned} R_2^A &= r_2^D + r^{NC} + r_2^R \\ &= r_2^D + r^{NC} + r_2^L \\ &= C(A) - r_1^D - r_1^T = C(A) - C(A^1). \end{aligned}$$

■

So far, we have shown that if $C(A) \geq C(A^1) + C(A^2)$, then (R_1^A, R_2^A) coincides with the corner point R_I^* in Figure 6. When $C(A) < C(A^1) + C(A^2)$, (R_1^A, R_2^A) coincides with the corner point $R_{II,1}^*$ in Figure 7; we still need to show that the other corner point is achievable in order to establish the achievability of Δ. Thus, we introduce our next lemma.

Lemma 9: Let (R_{12}, R_{22}) be the rate assignment given by the rate assignment procedure in Appendix B when $\alpha = 0$. Then

$$R_1^A = \begin{cases} C(A^1) & \text{if } C(A) \geq C(A^1) + C(A^2) \\ C(A) - C(A^2) & \text{if } C(A) < C(A^1) + C(A^2) \end{cases}$$

and

$$R_2^A = C(A^2)$$

where A^1 and A^2 are the cuts defined in Procedure P2.

Proof: The proof of this lemma is identical to the proof of Lemma 8, with the exception that $\alpha = 0$ corresponds to User 2 first being given all of the available capacity for routing on the shared path groups. ■

Lemmas 8 and 9 provide schemes to achieve the corner points of region Δ. Time sharing of the two schemes results in the achievability of region Δ as a whole. This, together with the outer-bound result in Lemma 7, implies that Δ is the capacity region of the network.

APPENDIX E
SUPPORTING FACTS AND DEFINITIONS

In the proof of Theorem 3, we drew heavily on definitions and theorems given in [11]. The authors in [11] use the concept of informational dominance to relate the structure of a graph to information-theoretic notions of capacity. In the interest of completeness, we summarize here the relevant facts and definitions from this paper.

Fact 1: 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 $i \in \mathcal{N}$, $H(S(i)) \geq r_i \log_2(b)$.

Definition 6: 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 2: If $B \subseteq Dom(A)$, then $H(B) \leq H(A)$.

Fact 3 (Theorem 10, [11]): For an edge set A, $Dom(A)$ satisfies the following conditions:

1. $A \subseteq Dom(A)$,
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 $G(Dom(A), i)$.

Again, we emphasize that Definition 6 and Facts 1-3 are relevant results from [11] that we will use in examining our combined network coding/routing scheme. The reader is encouraged to examine that paper in greater detail.

ACKNOWLEDGMENT

The authors would like to thank Prof T. Ho, R. Appuswamy and Prof K. Zeger for their thoughtful discussions.

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, July 2000.

[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2006, pp. 243–254.

[3] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: a decentralized approach based on network coding," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, 2005.

[4] D. Traskov, N. Ratnakar, D. S. 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.

[5] Y. E. Sagduyu and A. Ephremides, "Joint optimization of MAC and network coding for cooperative and competitive wireless multicasting," in *Proceedings of Workshop on Modelling and Design of Multi-hop Wireless Networks*, 2006.

[6] ——, "Network coding in wireless queueing networks: Tandem network case," in *Proceedings of International Symposium on Information Theory (ISIT)*, 2006.

[7] J. Yuan, Z. Li, and B. L. W. Yu, "A cross-layer optimization framework for multihop multicast in wireless mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, November 2006.

[8] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," *Information Theory, IEEE Transactions on*, vol. 52, no. 11, pp. 5077–5087, 2006.

[9] 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*, September 2006.

[10] X. Yan, Y. Yang, and Z. Zhang, "An outer bound for multisource multisink network coding with minimum cost consideration," *IEEE Transactions on Information Theory*, 2006.

[11] 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.

[12] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[13] J. Price, "Resource allocation in multi-user communication systems," *Ph.D. Thesis*, 2007.

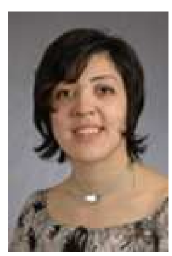
[14] J. Price and T. Javidi, "Network coding for resource redistribution in a unicast network," in *Proccedings of Forty-fifth Allerton Conference on Communications, Control, and Computing*, 2007.

[15] G. Folland, *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, Inc., 1999.

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

Jennifer Price (S'02-M'08) received her B.S. and M.S. degrees from the University of Oklahoma and the University of Washington in 2002 and 2004, respectively. She received her Ph.D. in Communications Theory and Systems from the University of California, San Diego in 2007.

She is currently an Assistant Professor at the University of Colorado at Colorado Springs. Her research interests include communication networks, optimization, distributed resource allocation, network pricing and economics, and game theory.



Tara Javidi (S'96-M'02) studied electrical engineering at Sharif University of Technology, Tehran, Iran from 1992 to 1996, where she received the B.S. degree in electrical engineering. She received the M.S. degrees in electrical engineering (systems), and in applied mathematics (stochastics) in 1998 and 1999, respectively, and the Ph.D. degree in electrical engineering and computer science in 2002, all from the University of Michigan, Ann Arbor.

From 2002 to 2004, Professor Javidi was an Assistant Professor with the Electrical Engineering Department, University of Washington, Seattle. She joined University of California, San Diego, in 2005, where she is currently an Assistant Professor of electrical and computer engineering. She was a Barbour Scholar during 1999-2000 academic year and received an NSF CAREER Award in 2004.

Her research interests are in communication networks, stochastic resource allocation, stochastic control theory, and wireless communications.