Forty-Fifth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 26-28, 2007

Intersession Network Coding for Two Simple Multicast Sessions

Chih-Chun Wang
Center for Wireless Systems and Applications (CWSA)
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47906

Ness B. Shroff
Center for Wireless Systems and Applications (CWSA)
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47906

***Abstract*— In this paper, the problem of transmitting two symbols from two sources to multiple destinations using network coding is considered for general directed acyclic graphs. When all destination nodes are interested in *both* symbols, the feasibility of this problem is characterized by the classic min-cut max-flow theorem. This work focuses on the scenarios in which each destination is interested in *one* source symbol. If no coding is allowed at intermediate nodes, the feasibility of this traditional two-session routing problem is equivalent to the existence of two edge-disjoint sets of paths. Similarly, in this paper, it is proven that the feasibility of this inter-session network coding problem is equivalent to the existence of multiple sets of paths with controlled edge overlaps. In particular, for the degenerate case of two sources and two destinations, it is proven that a binary field is sufficient and an *optimal* network coding solution must be of one of the three different forms, which includes the well-studied butterfly graph as a special case.**

I. Introduction

The goal of network communication is to exchange information packets simultaneously between different pairs of sources and sinks using the "edges/links" in the network. Traditionally, each information packet is regarded as an unsplittable commodity [1] and takes different routes from the sources to the sinks in the network. Each route occupies an exclusive share of the capacity of the edges. Network coding, on the other hand, allows not only information replicating but also information mixing at the intermediate nodes [2], [3], [4], [5], which has demonstrated significant throughput advantage over simple routing algorithms.

For a *single multicast session* problem, namely, when all sinks are interested in the same set of information flowing along the network, the characterization of the network coding capacity region is well understood. For directed graphs, network coding is able to achieve the long standing min-cut max-flow bound, and one can further limit the choice of coding schemes and use only *linear network coding* without loss of efficiency [3], [5]. For undirected graphs, network coding is able to approximate a similar cut bound within a constant factor of two [4].

The setting that different sinks are interested in different subsets of the information is equivalent to the coexistence of multiple sessions competing for the network resources and is generally termed the *multiple muticast session problem*. The benefit of network coding for multiple multicast sessions is clearly demonstrated in the butterfly graph of [2], [5]. Nonetheless, with multiple coexisting sessions, an optimal scheme has to balance the cooperative coding efforts of the intermediate nodes with the conflicting objectives of maximizing the throughput for each individual session, which perplexes the problem significantly. For example, linear network coding is no longer throughput optimal for multiple multicast sessions [6], deciding the existence of a linear network coding solution can be NP-complete [7] rather than polynomial time [8] in some scenarios, and until present, no coding gain has been observed in any undirected graph [9].

For the same reason, the capacity region or equivalently the feasibility of network coding is less understood for multiple multicast sessions, and only for very special graphs can the capacity region be characterized, such as directed cycles [10], degree-2 three-layer directed acyclic networks [11], and special bipartite undirected graphs [10]. Several outer bounds on the capacity region have been devised for general graphs based on *generalized edge cut* conditions of the underlying graph and the associated information-theoretic arguments, including fundamental regions in the entropy space [12], entropy calculus [13], the network-sharing bound [11], the information dominance condition [10], and the edge-cut bounds [14]. The achievability results, i.e., the inner bound on the capacity region, is generally determined by linear programming in a similar fashion to that of solving fractional multi-commodity flow, including the butterfly-based construction [15] and the pollution-treatment with powerset-based flow division [16].

In this paper, we continue our work in [17] and focus on the graph-theoretic characterization of the feasibility of network coding for general directed acyclic networks. The necessary and sufficient condition in [17] will be generalized from the setting of two simple *unicast* sessions to that of two simple *multicast* sessions. Namely, source nodes s_1 and s_2 would like to transmit two symbols X_1 and X_2 to two groups of receivers $\mathbf{t}_1 = \{t_{1,i}\}_i$ and $\mathbf{t}_2 = \{t_{2,j}\}_j$ in a directed acyclic network and each group is interested in one symbol respectively. Note that $\mathbf{t}_1 \cap \mathbf{t}_2$ can be empty or not. One particularly important application of this setting is multi-resolution multicast based on network coding, for which $s_1 = s_2$, X_1 and X_2 represent the low and high resolution bits respectively, and $\mathbf{t}_1 \supseteq \mathbf{t}_2$.

It is well known that for two multicast sessions, the existence of a routing scheme is equivalent to the existence of two edge-disjoint sets of paths. In this work, we first prove an analogous result that the existence of a network coding scheme is equivalent to the existence of sets of paths with *controlled edge overlaps*. This result bridges the gap between the characterization theorems of multiple-session routing and single-session network coding for the non-trivial case of two source symbols. Various implications of the new characterization theorem are discussed, including the complexity and the bandwidth requirement of a network coding solution, the sufficiency of linear network coding, and the source-sink reciprocity. The characterization theorem for the case of two unicast sessions is also strengthened and it is proven that a binary field is sufficient for network coding and an *optimal* network coding solution must be of one of the three different forms, which include the well-studied butterfly graph as a special case.

II. Formulation & Preliminary Results

A. The Setting

We consider *finite, directed, acyclic* graphs (DAGs) $G = (V, E)$. An edge $e = (u, v) \in E$ is an ordered pair of nodes, for which u and v are termed as the tail and the head of e respectively. A path P from node u to v is defined as an ordered set of edges $\{(u, w_1), (w_1, w_2), \cdots, (w_n, v)\} \subseteq E$ such that the head of the previous edge is the tail of the next one. We sometimes write $P = uw_1 \cdots w_n v$ and use the notation $P_{u,v}$ to emphasize the terminal nodes u and v. For a collection of paths $\mathcal{P} = \{P_1, \cdots, P_k\}$ and a given edge $e \in E$, the number of coinciding paths for edge e is defined as $\mathsf{ncp}_{\mathcal{P}}(e) = |\{P \in \mathcal{P} : e \in P\}|$, i.e., the number of paths that use link e. We say that an edge e_1 is an upstream edge of another edge e_2 if there exists a path starting from e_1 and ending at e_2. If e_1 is an upstream edge of e_2, then e_2 is a downstream edge of e_1.

We consider the following network coding problem with two simple multicast sessions: Given a finite DAG $G = (V, E)$, and two source nodes s_1 and s_2 and two groups of sink nodes $\mathbf{t}_1 = \{t_{1,i}\}_{i \in \mathbf{M}_1}$ and $\mathbf{t}_2 = \{t_{2,j}\}_{j \in \mathbf{M}_2}$, whether two single symbols X_1 and X_2, emanating from s_1 and s_2 respectively, can be "transmitted" simultaneously to $t_{1,i}$ and $t_{2,j}$ for all $i \in \mathbf{M}_1, j \in \mathbf{M}_2$ within *a single time slot*. It is assumed that each edge is capable of carrying one symbol per time slot, and there is no transmission delay. Parallel edges are used when the capacity between two nodes is greater than one.

Both X_1 and X_2 are drawn from a finite field $\mathsf{GF}(q)$ with sufficiently large q, see [3], [18] and the reference therein. Since the size of q is not of our primary interest, unless otherwise mentioned, the readers may safely assume that X_1 and X_2 take integer values instead, provided a sufficiently large q is adopted. Throughout the paper, we also assume that $P_{s_1,t_{1,i}}$ and $P_{s_2,t_{2,j}}$ exist for all i, j, which can be easily checked within polynomial time. Otherwise simultaneous transmission is simply impossible.

It is worth emphasizing that here we focus on achieving simultaneous transmission of two symbols within a single time slot, and therefore, time-sharing, or equivalently fractional routing/coding, is beyond the scope of this paper.

B. Preliminary Results for The 2–2 Case

In [17], we have proven the following results for a special subcase of the setting in this paper: $|\mathbf{M}_1| = |\mathbf{M}_2| = 1$, $\mathbf{t}_1 = \{t_1\}$, and $\mathbf{t}_2 = \{t_2\}$.

Theorem 1 (Characterization, [17]): A network coding solution exists if and only if one of the following two conditions holds.

- There exists a collection $\mathcal{P}$ of two paths P_{s_1,t_1} and P_{s_2,t_2}, such that $\max_{e \in E} \mathsf{ncp}_{\mathcal{P}}(e) \leq 1$.
- There exist a collection $\mathcal{P}$ of three paths P_{s_1,t_1}, P_{s_2,t_2}, and P_{s_2,t_1}, and a collection $\mathcal{Q}$ of three paths Q_{s_1,t_1}, Q_{s_2,t_2}, and Q_{s_1,t_2}, such that $\max_{e \in E} \mathsf{ncp}_{\mathcal{P}}(e) \leq 2$ and $\max_{e \in E} \mathsf{ncp}_{\mathcal{Q}}(e) \leq 2$.

Theorem 1 says that for the two-source/two-sink networking problem, the feasibility of a network coding solution is equivalent to the existence of paths with controlled edge overlaps (in contrast with the edge-disjointness requirement for routing solutions). In this work, we will generalize Theorem 1 for the two-source/m-sink problems first and provide a strengthened version of Theorem 1 for the 2–2 case later, both of which shed more insight on the multiple-session network coding problem.

Practical implications of Theorem 1 such as the distributed rate control algorithms based on the flow condition can be found in [19].

III. Main Results

A. Main Results for The 2–m Case

In this work, we consider arbitrary choices of $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{t}_1$, and $\mathbf{t}_2$. All theorems and corollaries are based on the two-multicast-session setting described in Section II-A, and it remains an open problem whether these results can be generalized for problems with three multicast sessions or more.

Theorem 2 (Characterization): The existence of a network solution is equivalent to there existing $|\mathbf{M}_1| + |\mathbf{M}_2| + 2$ collections of paths: $\mathcal{P}_i, \forall i \in \mathbf{M}_1$, $\mathcal{Q}_j, \forall j \in \mathbf{M}_2$, $\mathcal{R}$, and $\mathcal{S}$ such that

$$\forall i \in \mathbf{M}_1,\ \mathcal{P}_i = \begin{cases} \{P_{s_1,t_{1,i}}, P_{s_2,t_{1,i}}\} & \text{if } \exists \text{ a } (s_2, t_{1,i}) \text{ path} \\ \{P_{s_1,t_{1,i}}\} & \text{otherwise} \end{cases}$$

$$\forall j \in \mathbf{M}_2,\ \mathcal{Q}_j = \begin{cases} \{Q_{s_2,t_{2,j}}, Q_{s_1,t_{2,j}}\} & \text{if } \exists \text{ a } (s_1, t_{2,j}) \text{ path} \\ \{Q_{s_2,t_{2,j}}\} & \text{otherwise} \end{cases}$$

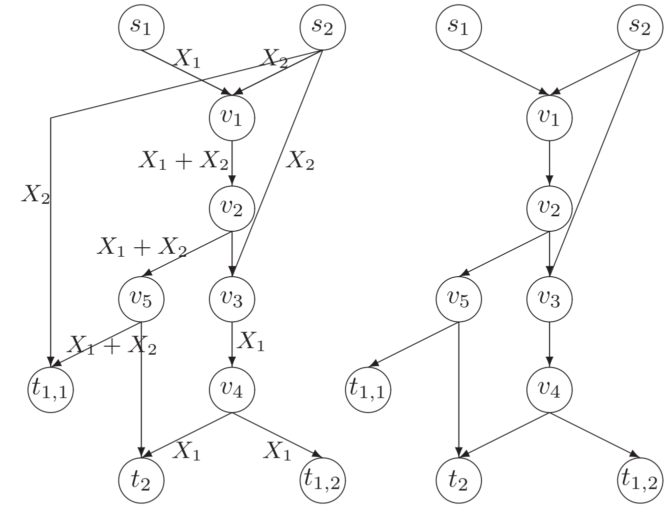
$$\mathcal{R} = \{R_{s_1,t_{1,i}} : \forall i \in \mathbf{M}_1\}$$

$$\mathcal{S} = \{S_{s_2,t_{2,j}} : \forall j \in \mathbf{M}_2\}$$

satisfy the following two conditions.

- Condition 1: $\max_{e \in E} \mathsf{ncp}_{\mathcal{P}_i \cup \{S_{s_2,t_{2,j}}\}}(e) \leq 2$ for all $i \in \mathbf{M}_1$ and $j \in \mathbf{M}_2$.
- Condition 2: $\max_{e \in E} \mathsf{ncp}_{\mathcal{Q}_j \cup \{R_{s_1,t_{1,i}}\}}(e) \leq 2$ for all $i \in \mathbf{M}_1$ and $j \in \mathbf{M}_2$.

The proofs for the necessity and sufficiency of Theorem 2 will be discussed in Sections IV and V respectively.



(a) A Feasible example. (b) An Infeasible Example.

Fig. 1. Two examples demonstrating Theorem 2. (a) satisfies Theorem 2 and the corresponding network coding solution is as described. (b) satisfies the cut-conditions but not the graph-characterization of network coding in Theorem 2. No network coding solutions exists for (b). Note if only one of $t_{1,1}$ and $t_{1,2}$ in (b) is requesting symbol X_1, then routing/network coding is possible. If both are requesting X_1, then simultaneous transmission becomes impossible.

Theorem 2 characterizes the feasibility of network coding for two simple multicast sessions for general DAGs, based on constructive flow-based conditions. Take Fig. 1 for example in which $\mathbf{t}_1 = \{t_{1,1}, t_{1,2}\}$ and $\mathbf{t}_2 = \{t_2\}$. In Fig. 1(a), the following selection of $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{S}$

$$P_{s_1,t_{1,1}} = s_1v_1v_2v_5t_{1,1} \qquad Q_{s_2,t_2} = s_2v_3v_4t_2$$
$$P_{s_2,t_{1,1}} = s_2t_{1,1} \qquad Q_{s_1,t_2} = s_1v_1v_2v_5t_2$$
$$P_{s_1,t_{1,2}} = s_1v_1v_2v_3v_4t_{1,2} \qquad R_{s_1,t_{1,1}} = s_1v_2v_5t_{1,1}$$
$$P_{s_2,t_{1,2}} = s_2v_3v_4t_{1,2} \qquad R_{s_1,t_{1,2}} = s_1v_1v_2v_3v_4t_{1,2}$$
$$S_{s_2,t_2} = s_2v_1v_2v_5t_2$$

satisfies Theorem 2 and a network coding solution is described in Fig. 1(a). On the other hand, Fig. 1(b) admits no network coding solution as explained in the following. Successful reception of X_1 in $t_{1,1}$ and $t_{1,2}$ requires both edges (v_2, v_5) and (v_3, v_4) carrying message X_1, which in turn prevents t_2 from receiving/decoding X_2 successfully. By enumerating all possible path combinations, one can also prove that there is no path selection satisfying Conditions 1 and 2, as predicted by Theorem 2. This generalization from the 2–2 case to the 2–m case is not trivial since Fig. 1(b) satisfies all cut-conditions. Moreover, if only one of $t_{1,1}$ and $t_{1,2}$ is requesting X_1 and the other one is dormant, the degenerate two-unicast problem becomes feasible. Only when both $t_{1,1}$ and $t_{1,2}$ are requesting symbol X_1 simultaneously, does the two-multicast problem become infeasible.

In general, when $|\mathbf{M}_1| = |\mathbf{M}_2| = 1$, Theorem 2 collapses to Theorem 1 by the nature of its statements. Another interesting degenerate case is when $\mathbf{t}_1 = \mathbf{t}_2$, i.e. all sinks are interested in both symbols. Condition 1 guarantees the min-cut between $t_{1,i}$ and any one of the sources is no less than one and the min-cut between $t_{1,i}$ and $\{s_1, s_2\}$ is no less than two. By symmetry, similar statements can be made for $t_{2,j}$. Therefore, Theorem 2 collapses to the classic min-cut max-flow characterization for the single-session problem.

Several implications of Theorem 2 are discussed as follows.

Corollary 1 (Complexity): Deciding the existence of a network coding solution is a polynomial-time problem.

Proof: Due to space limitation, we provide only a sketch of the proof for interested readers, which is based heavily on the pebbling game approach in [20].

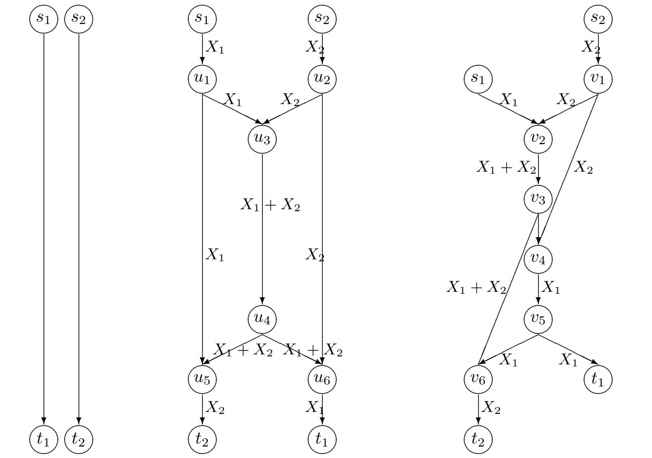
Step 1: Augment the graph G by converting each edge into two parallel edges. Step 2: Convert the augmented graph to its line graph counterpart. Step 3.1: To decide the existence of $\mathcal{P}_i, \forall i \in \mathbf{M}_1$ and $\mathcal{S}$ satisfying Condition 1, we will follow closely the pebbling game approach in Lemma 4 and Theorem 3 of [20] for which we have pebbles labelled as $\pi_{1,i}$, $\pi_{2,i}$ and σ_j, totally $2|\mathbf{M}_1|+|\mathbf{M}_2|$ pebbles, corresponding to the $2|\mathbf{M}_1| + |\mathbf{M}_2|$ paths of interest. Step 3.2: Modify condition (2b) in [20] to incorporate the three cases: (i) if the to-be-moved pebble is σ_{j_0} for some j_0, then m cannot contain $\pi_{1,i}$ or $\pi_{2,i}$ pebbles for any i, (ii) if the to-be-moved pebble is π_{1,i_0} for some i_0, then m cannot contain π_{2,i_0} or σ_j for any j, and (iii) if the to-be-moved pebble is π_{2,i_0} for some i_0, then m cannot contain π_{1,i_0} or σ_j for any j. Step 3.3: with the above modification, the winning strategy of a pebbling game is equivalent to the the existence of $\mathcal{P}_i, \forall i \in \mathbf{M}_1$ and $\mathcal{S}$ satisfying Condition 1, and therefore can be determined in polynomial time. Step 4: Repeat Step 3 for $\mathcal{Q}_j, \forall j \in \mathbf{M}_2$, $\mathcal{R}$, and Condition 2. Step 5: If both Step 3 and Step 4 confirm the existence of $\mathcal{P}_i, \forall i \in \mathbf{M}_1$, $\mathcal{S}$, $\mathcal{Q}_j, \forall j \in \mathbf{M}_2$, and $\mathcal{R}$ respectively, then a network coding solution must exist by Theorem 2. Otherwise, it is impossible to transmit X_1 and X_2 using network coding. ■

Corollary 2 (Bandwidth requirement): If there exists a network coding solution, there exists a network coding solution using at most $3|\mathbf{M}_1| + 3|\mathbf{M}_2|$ paths during transmission.

Proof: This corollary is a straightforward result of Theorem 2 since one can remove any edges not used by the $3|\mathbf{M}_1| + 3|\mathbf{M}_2|$ paths in $\mathcal{P}_i, \forall i \in \mathbf{M}_1$, $\mathcal{Q}_j, \forall j \in \mathbf{M}_2$, $\mathcal{R}$ and $\mathcal{S}$. By Theorem 2, a network coding solution still exists for the trimmed graph. ■

Although the above corollary discusses how to "trim" the network for a more bandwidth-efficient network coding solution, it is worth mentioning that the main contribution of Theorem 2 is the graph-theoretic characterization that is parallel to the min-cut max-flow theorem. Corollary 2 presents a simple way of trimming the network for two multicast sessions, which is the first of its kind but is unlikely to be the most efficient. A more delicate information-decomposition-based simplification (similar to that for single-session network coding [21]) will be built upon the basis of this new characterization.

Corollary 3 (Sufficiency of Linear Network Codes): The existence of a non-linear network coding solution implies the existence of a linear network coding solution.



(a) 2-EDPs (b) The full butterfly (c) The full grail

Fig. 2. Three classes of graphs for which simultaneous transmission of X_1 and X_2 along (s_1, t_1) and (s_2, t_2) is feasible via routing/network coding.

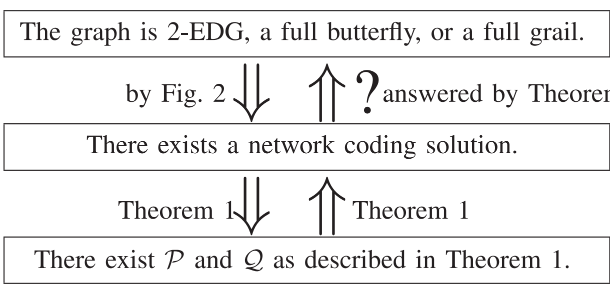


Fig. 3. Comparison of Theorems 1 and 3

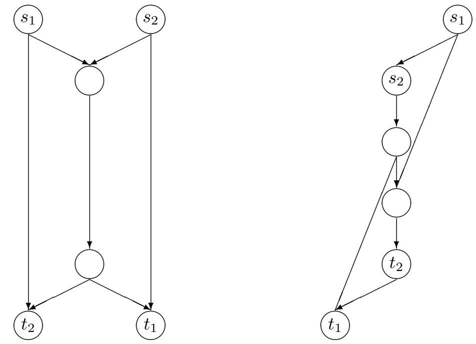
The above corollary is a byproduct of the proof of Theorem 2 and will be discussed in Section IV.

B. Strengthened Characterization Theorem for The 2–2 Case

Consider three simple graphs: the 2 edge-disjoint paths (2-EDPs) (see Fig. 2(a)), the full butterfly (Fig. 2(b)), and the full grail (Fig. 2(c)). For all these three graphs, it is feasible to send simultaneously X_1 and X_2 from s_1 and s_2 to t_1 and t_2 respectively via routing/network coding. (The corresponding network coding solutions are also described in Fig. 2.)

We can then make the statement that any graph being one of the three types in Fig. 2 must admit a network coding solution. In addition, Theorem 1 proves the equivalency between the feasibility of network coding and the existence of two special path collections $\mathcal{P}$ and $\mathcal{Q}$. Fig. 3 summarizes the current state-of-the-art knowledge for the two-unicast-session problem, in which three arrows have been established, except the upper left one. We will complete the picture by showing that if a network coding solution exists, then the graph must "be" one of the three cases: 2-EDPs, the butterfly, and the grail. This statement is made rigorous in the following theorem. See [22] for the graph-theoretic definitions of "edge-contraction," "subdivison," and "independent paths."

Theorem 3: Define two sets of graphs, $\mathcal{G}_b$ and $\mathcal{G}_g$, as follows.



(a) A degenerate butterfly (b) A degenerate grail

Fig. 4. Two examples of graphs that can be obtained from the full butterfly and the full grail by edge contraction.

- $\mathcal{G}_b$ contains the full butterfly as described in Fig. 2(b), and all graphs obtained from the full butterfly via edge contraction, e.g. , Fig. 4(a).
- $\mathcal{G}_g$ contains the full grail as described in Fig. 2(c), and all graphs obtained from the full grail via edge contraction,[1] e.g. Fig. 4(b).

Suppose there exists a network coding solution to the two-unicast-session problem. Then one of the following two conditions must hold.

1) There exist two EDPs connecting (s_1, t_1) and (s_2, t_2).
2) G contains a subgraph $G' = (V', E')$ such that (i) $\{s_1, s_2, t_1, t_2\} \subseteq V'$ and (ii) there exists a $G_r \in \mathcal{G}_b \cup \mathcal{G}_g$ such that G' is a subdivision of G_r. Namely, G' can be obtained from G_r by replacing each edge of G_r with an interior-vertex-disjoint path, also known as an *independent path*.

Theorem 3 says that if a network coding solution exists, then it is either the case that a routing solution also exists (with 2-EDPs) or the case in which the graph contains a subgraph that is topologically similar to a butterfly or a grail. The closest counterpart of Theorem 3 is the information-decomposition arguments in [21], which specifies the minimal coding-based topology. There are at least three fundamental differences between Theorem 3 and the results in [21]. First, the subject of the former is the two-unicast-session problem while the latter focuses on the single-multicast-session problem. Second, Theorem 3 is obtained from the pure graph-theoretic argument rather than coding-based simplification. Third, Theorem 3 focuses directly on the graph G rather than its line graph, which in turn provides a tighter description of the graph structure. For example, from the coding perspective [21], the question of whether two paths share a vertex is irrelevant as long as these two paths are edge disjoint. For comparison, the second condition in Theorem 3 makes a stronger statement on

[1] Unlike the butterfly, the grail graph is asymmetric for source-sink pairs (s_1, t_1) and (s_2, t_2). For simplicity, we do not use a different class to represent the symmetric version of Fig. 2(c). To be more precise, swapping the indices/subscript of Fig. 2(c) is a legitimate method of creating $\mathcal{G}_g$. For example, Fig. 4(b) is obtained from Fig. 2(c) by edge contraction and index/subscript swapping.

the vertex-disjointness between different paths.

Although Theorem 3 focuses on the graph-theoretic statement that G contains a G' that is topologically similar to either a butterfly or a grail, it can be easily translated to the corresponding network coding statement. Namely, the network coding solutions for Figs. 2(b) and 2(c) can be applied straightforward to the graph G by letting the independent paths in $G' \subseteq G$ carrying the same messages as the corresponding edges in G_r.

From the above observation, we have the following corollaries.

Corollary 4: If there exists any network coding solution for the two-unicast problem, there exists a binary linear network coding solution for the same problem. Furthermore, either a routing-based solution exists (with no encoding/decoding) or one needs exactly three binary exclusive OR (XOR) operations for the entire network encoding/decoding solution.

Proof: This corollary is proven straightforwardly by Theorem 3 and by observing that there are exactly three XOR binary operations in the cases of a butterfly and a grail. ■

Theorem 3 also implies the following results on the bandwidth and coding efficiency of network coding solutions.

Definition 1: Consider two network coding solutions NC_1 and NC_2 for the same two-unicast-session problem based on finite fields $\mathsf{GF}(q_1)$ and $\mathsf{GF}(q_2)$, respectively. Use $G_1 = (V_1, E_1) \subseteq G$ to denote the active nodes and edges used in NC_1 and similarly does $G_2 = (V_2, E_2) \subseteq G$ for NC_2. We say that NC_1 is more bandwidth-efficient than NC_2 if $E_1 \subseteq E_2$. NC_1 is more coding-efficient than NC_2 if $q_1 \leq q_2$.

Corollary 5: For any network coding solution NC of a form different than those in Fig. 2(a–c), there exists another solution $\mathrm{NC}_{\mathrm{opt}}$ of the form in Fig. 2(a–c) that is both more bandwidth-efficient and more coding-efficient than NC. Therefore, the three cases in Fig. 2 are the only solutions that are *jointly bandwidth-&-alphabet optimal*.

We close this section by stating the *source-sink reciprocity* of two-multicast-session network coding.

Corollary 6 (Source-sink Reciprocity): Consider the general setting of two simple multicast sessions with arbitrary $\mathbf{t}_1$ and $\mathbf{t}_2$. Suppose a network coding solution exists for source-sink pairs $(s_1, \mathbf{t}_1)$ and $(s_2, \mathbf{t}_2)$ in G. Fix the values of i_0, j_0 and construct G_R by reversing the orientation of all edges in G. Then there exists a network coding solution for the two-simple-unicast problem of transmitting two symbols Y_1 and Y_2 for the reverse source-sink pairs (t_{1,i_0}, s_1) and (t_{2,j_0}, s_2) in G_R. In other words, the feasibility in one orientation implies the feasibility in the reverse orientation.

Proof: Since reciprocity holds for all three cases in Fig. 2, it is a straightforward implication of Theorem 3. ■

IV. Necessary Condition of Network Coding

In this section, we provide a proof of the necessary condition of Theorem 2. Since the condition on $\mathcal{P}_i, \forall i \in \mathbf{M}_1$ and $\mathcal{S}$ is completely decoupled from that of $\mathcal{Q}_j, \forall j \in \mathbf{M}_2$ and $\mathcal{R}$, we focus only on constructing $\mathcal{P}_i$ and $\mathcal{S}$ satisfying Condition 1, given the existence of a network coding solution. The other case can be obtained by symmetry.

Suppose that a linear network coding solution exists. (The case in which only a non-linear network solution exists will be addressed in the end of this proof.) Let $S_{s_2,t_{2,j}}$ denote the path from s_2 to $t_{2,j}$ such that all its edges are carrying messages with a non-zero coefficient for X_2. Since $t_{2,j}$ is able to recover X_2, this type of path $S_{s_2,t_{2,j}}$ always exists. This special path selection $\mathcal{S}$ is fixed throughout the entire proof.

For any i, we can assume $P_{s_2,t_{1,i}}$ exists. Otherwise, we just select an arbitrary $P_{s_1,t_{1,i}}$. Then $\mathcal{P}_i \cup \{S_{s_2,t_{2,j}}\} = \{P_{s_1,t_{1,i}}, S_{s_2,t_{2,j}}\}$ contains only two paths and $\max_{e\in E} \mathsf{ncp}_{\mathcal{P}_i\cup\{S_{s_2,t_{2,j}}\}}(e) \leq 2$ holds for all $j \in \mathbf{M}_2$.

Arbitrarily select two paths $P^{(1)}_{s_1,t_{1,i}}$ and $P^{(1)}_{s_2,t_{1,i}}$. We will then iteratively construct $P^{(k+1)}_{s_1,t_{1,i}}$ and $P^{(k+1)}_{s_2,t_{1,i}}$ from $P^{(k)}_{s_1,t_{1,i}}$ and $P^{(k)}_{s_2,t_{1,i}}$ and show that within a finite number of iterations K, the selections $\mathcal{P}_i = \{P^{(K)}_{s_1,t_{1,i}}, P^{(K)}_{s_2,t_{1,i}}\}$ and $\mathcal{S}$ satisfy Condition 1 for all $j \in \mathbf{M}_2$.

If $\{P^{(k)}_{s_1,t_{1,i}}, P^{(k)}_{s_2,t_{1,i}}\}$ and $\mathcal{S}$ satisfy Condition 1, the iteration stops and we set $K = k$. If not, then there exists at least one violating edge e such that all three paths $\{P^{(k)}_{s_1,t_{1,i}}, P^{(k)}_{s_2,t_{1,i}}, S_{s_2,t_{2,j}}\}$ share e for some j. Note, that all these violating edges are strictly ordered since they participate in the paths $P^{(k)}_{s_1,t_{1,i}}$ and $P^{(k)}_{s_2,t_{1,i}}$. Among all violating edges, let $e^{(k)}$ denote the edge that is the closest to $t_{1,i}$. We will construct another pair $P^{(k+1)}_{s_1,t_1}$ and $P^{(k+1)}_{s_2,t_1}$ based on $P^{(k)}_{s_1,t_1}$ and $P^{(k)}_{s_2,t_1}$ such that the closest violating edge $e^{(k+1)}$ (if exists) is a strictly upstream edge of $e^{(k)}$. By repeatedly applying the aforementioned construction and by the finiteness of the graph, we can find a pair $P^{(K)}_{s_1,t_{1,i}}$ and $P^{(K)}_{s_2,t_{1,i}}$ within a finite number of iterations such that Condition 1 is satisfied.

Before we proceed, we need the following terminology. For two paths P and Q and three nodes x, y, and z, xPy denotes the path segment connecting nodes x and y using path P. Similarly, $xPyQz$ denotes the path segment connecting nodes x, y, and z while P and Q are used during the x–y and y–z segments respectively.

The construction of $P^{(k+1)}_{s_1,t_{1,i}}$ and $P^{(k+1)}_{s_2,t_{1,i}}$ from $P^{(k)}_{s_1,t_{1,i}}$ and $P^{(k)}_{s_2,t_{1,i}}$ is as follows. Suppose $e^{(k)} = (u, v)$. Due to the special construction of $\mathcal{S}$, the message along $e^{(k)}$ must have a non-zero coefficient of X_2. Therefore, there must be another path from either s_1 or s_2 that reaches $t_{1,i}$ without using $e^{(k)}$. Otherwise, the linear network coding is short of new input, and $t_{1,i}$ can only receive *linearly dependent* messages all corrupted by X_2. Let A denote the new path reaching $t_{1,i}$ without using $e^{(k)}$ and w be the first node that A touches either $v\left(P^{(k)}_{s_1,t_{1,i}}\right)t_{1,i}$ or $v\left(P^{(k)}_{s_2,t_{1,i}}\right)t_{1,i}$.

There are four different cases depending on whether the new path is from s_1 or s_2 and on whether w belongs to $v\left(P^{(k)}_{s_1,t_{1,i}}\right)t_{1,i}$ and $v\left(P^{(k)}_{s_2,t_{1,i}}\right)t_{1,i}$. Multiple cases may be active at the same time, and the argument for any one of the active cases is sufficient for this proof.

Case 1: $A = A_{s_1,t_{1,i}}$ and $w \in v\left(P^{(k)}_{s_1,t_{1,i}}\right)t_{1,i}$. Construct

$P^{(2)}_{s_1,t_{1,i}}$ and $P^{(2)}_{s_2,t_{1,i}}$ as follows.

$$P^{(k+1)}_{s_1,t_{1,i}} \leftarrow s_1\left(A_{s_1,t_{1,i}}\right) w \left(P^{(k)}_{s_1,t_{1,i}}\right) t_{1,i}$$
$$P^{(k+1)}_{s_2,t_{1,i}} \leftarrow P^{(k)}_{s_2,t_{1,i}}$$

Since $e^{(k)}$ is the violating edge closest to $t_{1,i}$ along paths $P^{(k)}_{s_1,t_{1,i}}$ and $P^{(k)}_{s_2,t_{1,i}}$, the new closest violating edge $e^{(k+1)}$ under the new selection $P^{(k+1)}_{s_1,t_{1,i}}$ and $P^{(k+1)}_{s_2,t_{1,i}}$ cannot be a strictly downstream edge of $e^{(k)}$. Furthermore, $s_1\left(A_{s_1,t_{1,i}}\right) w$ does not use $e^{(k)}$ so the new violating edge $e^{(k+1)} \neq e^{(k)}$. Since $e^{(k)}$ still participates in $P^{(k+1)}_{s_2,t_{1,i}}$, $e^{(k+1)}$ must be a strictly upstream edge of $e^{(k)}$. The construction is complete.

Case 2: $A = A_{s_2,t_{1,i}}$ and $w \in v\left(P^{(k)}_{s_1,t_{1,i}}\right) t_{1,i}$. Construct $P^{(k+1)}_{s_1,t_{1,i}}$ and $P^{(k+1)}_{s_2,t_{1,i}}$ as follows.

$$P^{(k+1)}_{s_1,t_{1,i}} \leftarrow s_1\left(P^{(k)}_{s_1,t_{1,i}}\right) uv \left(P^{(k)}_{s_2,t_{1,i}}\right) t_{1,i}$$
$$P^{(k+1)}_{s_2,t_{1,i}} \leftarrow s_2\left(A_{s_2,t_{1,i}}\right) w \left(P^{(k)}_{s_1,t_{1,i}}\right) t_{1,i}$$

By the same argument as in Case 1, $e^{(k+1)}$ cannot be a strictly downstream edge of $e^{(k)}$ and $e^{(k+1)} \neq e^{(k)}$. Since $e^{(k)} = (u,v)$ still participates in $P^{(k+1)}_{s_1,t_{1,i}}$, $e^{(k+1)}$ must be a strictly upstream edge of $e^{(k)}$. The construction is complete.

Cases 3 and 4 can be obtained by symmetry.

The proof for linear network coding is thus complete. For non-linear network coding, we simply modify the construction of $S_{s_2,t_{2,j}}$ by selecting the path from s_2 to $t_{2,j}$ such that all its edges are carrying messages $f_e(X_1,X_2)$ with a non-zero conditional mutual information $I(f_e(X_1,X_2);X_2|X_1)$. The rest of the proof follows verbatim.

V. Sufficient Condition of Network Coding

For any i if there exists no path connecting s_2 and $t_{1,i}$, then the messages sent from s_1 to $t_{1,i}$ cannot be corrupted by symbol X_2. A pure routing solution is feasible for the s_1 to $t_{1,i}$ transmission, and has zero impact on the rest of the network. Therefore, those i can be considered separately from the rest of the network. Without loss of generality, we can thus assume that for all $i \in \mathbf{M}_1$, there exists a path connecting s_2 and $t_{1,i}$. Similarly, we assume that there exists a path connecting s_1 and $t_{2,j}$ for all $j \in \mathbf{M}_2$.

In this section, we will show that if Conditions 1 and 2 of Theorem 2 are satisfied, then there exists a linear, *add-up-&-reset* network coding solution. Before proceeding, we will digress from our proof and explain the add-up-&-reset scheme.

A. The Add-Up-&-Rest Scheme

For any linear network coding scheme, the message M along any edge is a linear combination $M = c_1X_1 + c_2X_2$ of the symbols X_1 and X_2, and we use the corresponding coding vector $M = (c_1, c_2)$ as shorthand. To ensure the computability of network coding, the outgoing message, as a two-dimensional vector, must be in the span of all incoming messages. We use M_e to denote the message along a specific edge e.

The "add-up stage" of the proposed add-up-&-reset scheme uses the following mixing rules for an edge e.

1) If all the incoming messages of e are identical (including the case when there is only one incoming message), then M_e, the message for edge e, is set to be identical to the incoming messages.
2) Suppose one of the incoming messages (denoted by $M_1, \cdots, M_m$) is different. Choose M_e such that $M_e = a_1M_1 + \cdots + a_mM_m$ for some *strictly positive* integer coefficients a_1 to a_m, such that M_e is linearly independent of any other messages $M_{e'}$ for those e' not in the downstream of e.

This add-up stage can always be completed for any DAG, since one can assign M_e sequentially for different e's according to the topological order of e. The messages for the upstream edges are assigned earlier while the the messages for the downstream edges are chosen later according to the second rule of the add-up stage.

Several properties of the add-up stage can be proved and are stated as follows.

- If two messages M_1 and M_2 are not identical, they are linearly independent.
- For a message along an edge e, its *originating edge* is defined as the furthest upstream edge e' of e such that $M_e = M_{e''}$ for all e'' along any path from e' to e.[2] If two messages M_1 and M_2 are identical, they must have the same originating edge.
- Along any edge that is reachable from s_i, the i-th component of the corresponding message must be a strictly positive integer.

The add-up stage can be complemented with the "reset-to-X_i" operation described as follows. Take $i = 1$ for example and consider again the sequential construction according to the topological order of the edges. For any edge $e = (u,v)$ that has two independent incoming messages, the designer of the network code can choose whether to apply the reset-to-X_1 operation or to use the regular "add-up" operation. If the reset-to-X_1 operation is chosen, then the add-up construction $M_e = a_1M_1 + \cdots + a_mM_m$ in Rule 2 is replaced by $M_e = (1,0)$. The message M_e is indeed "reset" to X_1. The messages of the remaining edges will be assigned sequentially following the two add-up rules until the designer decides to apply another reset-to-X_i operation.

B. The Proof

We first perform the add-up scheme on the active subgraph $G' \triangleq \bigcup_{i\in\mathbf{M}_1} \mathcal{P}_i \bigcup_{j\in\mathbf{M}_2} \mathcal{Q}_j \cup \mathcal{R} \cup \mathcal{S}$.

After the add-up stage, suppose there exist some $t_{1,i}$ and $t_{2,j}$ who are not able to recover their intended symbols successfully, which means those $t_{1,i}$ (or $t_{2,j}$ resp.) must receive identical messages that are contaminated by the unwanted symbol X_2 (or X_1 resp.). Let I denote the collection of all such i and J denote the collection of all such j. For

[2] The "originating edge" is similar to the root of the coding subtrees in the line graph discussed in [21].

each $i \in I$, let (u_i, v_i) denote the edge that originates all the incoming identical messages of $t_{1,i}$. Similarly, for each $j \in J$, (u_j, v_j) denote the edge that originates all the incoming identical messages of $t_{2,j}$.

We first note that for each $i \in I$ and the corresponding edge (u_i, v_i), $\mathsf{ncp}_{\mathcal{P}_i \cup \{R_{s_1,t_{1,i}}\}}(u_i, v_i) = 3$. Namely, both paths in $\mathcal{P}_i$ and the path $R_{s_1,t_{1,i}}$ must use edge (u_i, v_i). Otherwise, the three paths $P_{s_1,t_{1,i}}$, $P_{s_2,t_{1,i}}$, and $R_{s_1,t_{1,i}}$ will carry at least two different messages to $t_{1,i}$, which contradicts $i \in I$. Similarly, for each $j \in J$, $\mathsf{ncp}_{\mathcal{Q}_j \cup \{S_{s_2,t_{2,j}}\}}(u_j, v_j) = 3$.

For any i, since (u_i, v_i) is reachable from s_1 along $P_{s_1,t_{1,i}}$, the message on (u_i, v_i) must have a non-zero coefficient of X_1. Therefore, the reset-to-X_1 operation is feasible on edge (u_i, v_i) and once the reset-to-X_1 operation is performed, $t_{1,i}$ is able to receive X_1 successfully. Similarly, for any j, if the reset-to-X_2 operation is performed on (u_j, v_j), then $t_{2,j}$ will receive X_2 successfully. The above reasoning shows that the reset operation is, in some sense, capable of sending uncorrupted symbols to the corresponding $t_{1,i}$ (or $t_{2,j}$).

Our goal is to apply the reset operations sequentially for all $i \in I$ and $j \in J$ according to the topological order of the corresponding (u_i, v_i) and (u_j, v_j). Namely, the reset operations will be applied to the upstream edges (u_i, v_i) (or (u_j, v_j)) earlier and then later applied to the downstream edges. Since the effect of a reset operation on any edge e will ripple through the network and change the messages carried in the downstream edges of e, we have to prove that when Conditions 1 and 2 are satisfied, (i) it is feasible to continue the reset operation sequentially without running into any unwanted scenarios, and (ii) resetting any (u_i, v_i) (or (u_j, v_j)) will not create any new sink that fails to recover the intended symbols. Provided that (i) and (ii) hold, it can be easily seen that the resulting add-up-&-reset scheme is able to send X_1 and X_2 to the designated sinks $\mathbf{t}_1$ and $\mathbf{t}_2$ successfully.

The proof will be completed by induction. For the following, we assume several reset operations have already been performed and the next to-be-reset edge is (u_{i_0}, v_{i_0}) for some $i_0 \in I$. The case that the next to-be-reset edge is (u_{j_0}, v_{j_0}) can be obtained by symmetry.

To prove Statement (i), we made the following observation that can be proved rigorously.

- Performing reset-to-X_1 on edge (u_{i_0}, v_{i_0}) is always possible when the message carried by (u_{i_0}, v_{i_0}) is (a, b) for some non-zero X_1 coefficient $a \neq 0$.

Since the reset-to-X_1 operations performed previously do not change the set of edges carrying messages with non-zero X_1 coefficients, we only have to consider whether there is any reset-to-X_2 operation performed previously that has resulted in (u_{i_0}, v_{i_0}) carrying a message with zero X_1 coefficient. By noting that $\mathsf{ncp}_{\mathcal{Q}_j}(u_j, v_j) = 2, \forall j \in J$ and by Condition 2 in Theorem 2, $R_{s_1,t_{1,i_0}}$ does not contain any $(u_j, v_j), \forall j$. Therefore, no edge in $R_{s_1,t_{1,i_0}}$ was reset to X_2 previously. The messages along $R_{s_1,t_{1,i_0}}$ are not affected by the reset-to-X_2 operations, and thus the message along $(u_{i_0}, v_{i_0}) \in R_{s_1,t_{1,i_0}}$ must have a non-zero coefficient of X_1 (since it is reachable from s_1 and not reset to X_2 previously). From the above reasoning, performing reset-to-X_1 is always feasible and Statement (i) is proved.

To prove Statement (ii), we need the following observations, the proof of which is omitted due to lack of space.

- For any $i_1, i_2 \in I$, if (u_{i_2}, v_{i_2}) is not an upstream edge of (u_{i_1}, v_{i_1}), then t_{1,i_1} is not a downstream node of (u_{i_2}, v_{i_2}). The same statement holds when we replace $i_2 \in I$ by $j_2 \in J$. Furthermore, its symmetric version holds for $j_1, j_2 \in J$ and t_{2,j_1} as well.
- After performing reset operations, two messages are independent if they are not identical. Furthermore, for two identical messages, either they are originated from the same edge, or they are of the form $(1, 0)$ or $(0, 1)$ (namely, they contain only one symbol).

The proof of Statement (ii) contains two claims as follows.

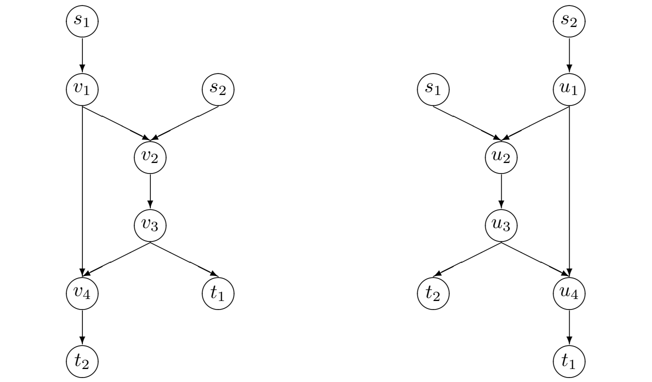
Claim 1: Performing reset-to-X_1 operation on (u_{i_0}, v_{i_0}) will not affect the successful reception for those $t_{1,i}, i \in I$ and $t_{2,j}, j \in J$ for which we have already performed reset operations. Since the reset operation is performed in the topological order, (u_{i_0}, v_{i_0}) is not an upstream edge of those (u_i, v_i) and (u_j, v_j). By the first observation, those $t_{1,i}$ and $t_{2,j}$ are not downstream nodes of (u_{i_0}, v_{i_0}) and are thus not affected by the reset-to-X_1 operation on (u_{i_0}, v_{i_0}). The successful receptions of X_1 and X_2 at those $t_{1,i}$ and $t_{2,j}$ remain unchanged.

Claim 2: Performing reset-to-X_1 operation on (u_{i_0}, v_{i_0}) will not affect the successful reception for those $t_{1,i}, i \in \mathbf{M}_1 \backslash I$ and $t_{2,j}, j \in \mathbf{M}_2 \backslash J$. Consider $t_{1,i}, i \in \mathbf{M}_1 \backslash I$ first and the $t_{2,j}$ case follows by symmetry. Since for any $i \in \mathbf{M}_1 \backslash I$, $t_{1,i}$ must receive different messages during the add-up stage, the messages entering $t_{1,i}$ are not originated from the same edge. By the second observation, after performing reset-to-X_1 on (u_{i_0}, v_{i_0}), the messages entering $t_{1,i}$ must either contain two independent messages or be identical and of the form $(1, 0)$ or $(0, 1)$. If the messages are independent, then $t_{1,i}$ can recover X_1 by solving linear equations. If the messages are identical and are $(1, 0)$, then $t_{1,i}$ receives uncontaminated X_1 successfully. The only unwanted scenario is when the identical messages are $(0, 1)$. Following the same argument as in the proof of Statement (i), for any $i \in \mathbf{M}_1 \backslash I$, no edge in $R_{s_1,t_{1,i}}$ is reset to X_2 and the message entering $t_{1,i}$ along $R_{s_1,t_{1,i}}$ must have a non-zero X_1 coefficient. As a result, the messages entering $t_{1,i}$ cannot be all identical and be $(0, 1)$. The last scenario is impossible. Claim 2 is proved.

VI. Sketch of The Proof of Theorem 3

Theorem 1 states that the existence of a network coding solution is equivalent to either the existence of 2-EDPs or the existence of path collections $\mathcal{P}$ and $\mathcal{Q}$ in Theorem 1. Theorem 3 is thus a straightforward result of the following propositions.

Proposition 1 (The half butterfly): Suppose there exists a collection of three paths $\mathcal{P} = \{P_{s_1,t_1}, P_{s_2,t_2}, P_{s_2,t_1}\}$ such that $\max_{e \in E} \mathsf{ncp}_{\mathcal{P}}(e) \leq 2$. Let $G_{\mathcal{P}} \triangleq P_{s_1,t_1} \cup P_{s_2,t_2}, \cup P_{s_2,t_1}$



(a) The left half butterfly (LHB)

(b) The right half butterfly (RHB)

Fig. 5. The half butterflies.

denote the subgraph induced by $\mathcal{P}$. Then $G_{\mathcal{P}}$ contains a subgraph $G' = (V', E')$ such that $V_{s,t} = \{s_1, s_2, t_1, t_2\} \subseteq V'$ and G' is a *subdivision* of a right half butterfly (RHB) (Fig. 5(b)) or an edge-contracted version of a RHB. Similar statements can be made for $G_{\mathcal{Q}}$ by symmetry.

Proposition 2: Suppose $G_{\mathcal{P}} \subseteq G$ is a subdivision of a RHB or a subdivision of its edge-contracted version and similarly is $G_{\mathcal{Q}} \subseteq G$ for the LHB. Then $G_{\mathcal{P}} \cup G_{\mathcal{Q}} \subseteq G$ contains a subgraph $G' = (V', E')$ such that $V_{s,t} = \{s_1, s_2, t_1, t_2\} \subseteq V'$ and G' is a subdivision of the G_r described in Theorem 3.

Proposition 1 establishes the existence of half butterflies while Proposition 2 focuses on the topological relationship between two half-butterflies. Due to the limit of space, the proofs of both propositions are omitted.

VII. Conclusion and Future Work

In this work, we have provided a graph-theoretic, flow-based, characterization theorem for network coding with two simple multicast sessions, which finds applications in multi-resolution multicast scenarios, and is a precursor to the full understanding of general multiple-multicast-session problems. Various implications have been discussed, including the complexity and the bandwidth requirement of a network coding solution, the sufficiency of linear network coding, and the source-sink reciprocity. For problems with two simple unicast sessions, we have also shown that it must contain one of the three bandwidth-optimal structures, which include the well-studied butterfly graph as a special case.

We conclude this paper by providing some future directions that we are currently investigating.

- Theorem 2 can be used to construct an inner bound of the capacity region for multiple multicast sessions based on pairwise network coding, an approach similar to [17]. A major goal of our investigation is to develop efficient distributed linear network coding schemes realizing the achievable capacity region. We have outlined preliminary work on multiple unicast sessions in [19].
- Based on the new characterization theorem, the information decomposition method for single-multicast-session problems [21] will be generalized for the two-multicast-session problems. We are interested in deciding the coding-&-bandwidth-optimal solution for intersession network coding with more than two receivers.
- A more general characterization for problems with more than two multicast sessions is currently under investigation.

References

[1] V. Guruswami, B. Shepherd, R. Rajaraman, and M. Yannakakis, "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems," in *Proc. 31st Annual ACM Symp. on Theory of Computing (STOC)*, 1999, pp. 19–28.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

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

[4] Z. Li and B. Li, "Network coding in undirected networks," in *Proc. 38th Conf. Inform. Sciences and Systems*. Princeton, NJ, USA, March 2004.

[5] S.-Y. R. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[6] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2745–2759, August 2005.

[7] A. Lehman, "Network coding," Ph.D. dissertation, MIT, 2005.

[8] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proc. 15th ACM Symp. Parallel Algorithms and Architectures*. San Diego, CA, USA, June 2003.

[9] Z. Li and B. Li, "Network coding: the case of multiple unicast sessions," in *Proc. 42nd Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, Illinois, USA, Sept. 2004.

[10] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information network," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2345–2364, June 2006.

[11] X. Yan, J. Yang, and Z. Zhang, "An outer bound for multisource multisink network codign with minimum cost consideration," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2373–2385, June 2006.

[12] L. Song, R. Yeung, and N. Cai, "Zero-error network coding for acyclic networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3129–3139, Dec. 2003.

[13] K. Jain, V. Vazirani, R. Yeung, and G. Yuval, "On the capacity of multiple unicast sessions in undirected graphs," in *Proc. Int'l Symp. Inform. Theory*. Seattle, USA, July 2006.

[14] G. Gramer and S. Savari, "Edge-cut bounds on network coding rates," *Journal of Network and Systems Management*, vol. 14, no. 1, pp. 49–67, March 2006.

[15] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. Int'l Symp. Inform. Theory*. Seattle, USA, July 2006.

[16] Y. Wu, "On constructive multi-source network coding," in *Proc. Int'l Symp. Inform. Theory*. Seattle, USA, July 2006.

[17] C.-C. Wang and N. Shroff, "Beyond the butterfly — a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proc. Int'l Symp. Inform. Theory*. Nice, France, June 2007.

[18] T. Ho, D. Karger, M. Médard, and R. Koetter, "Network coding from a network flow perspective," in *Proc. Int'l Symp. Inform. Theory*. Yokohama, Kanagawa, Japan, 2003.

[19] A. Khreishah, C.-C. Wang, and N. Shroff, "An optimization-based rate control for communication networks with inter-session network coding," in *Proc. IEEE Infocom*, 2007, submission preprint.

[20] S. Fortune, J. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *Theoretical Computer Science*, vol. 10, pp. 111–121, 1980.

[21] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 829–848, March 2006.

[22] R. Diestel, *Graph Theory*. New York: Springer-Verlag Heidelberg, 2005.