Pairwise Intersession Network Coding on Directed Networks

Chih-Chun Wang, *Member, IEEE,* Ness B. Shroff, *Fellow, IEEE,*

***Abstract*—When there exists only a single multicast session in a *directed acyclic/cyclic network*, the existence of a network coding solution is characterized by the classic min-cut/max-flow theorem. For the case of more than one coexisting sessions, network coding also demonstrates throughput improvement over non-coded solutions. This paper proposes *pairwise intersession network coding*, which allows for arbitrary directed networks but restricts the coding operations to being between two symbols (for acyclic networks) or between two strings of symbols (for cyclic networks). A graph-theoretic characterization of pairwise intersession network coding is proven based on *paths with controlled edge-overlap*. This new characterization generalizes the *edge-disjoint path* characterization of non-coded network communication and includes the well-studied butterfly graph as a special case. Based on this new characterization, various aspects of pairwise intersession network coding are studied, including the sufficiency of linear codes, the complexity of identifying coding opportunities, its topological analysis, and bandwidth- and coding-efficiency.**

***Index Terms*—Controlled edge-overlap, edge-disjoint paths, intersession network coding, intrasession network coding, the min-cut/max-flow theorem.**

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. In this work, we model the networks by *directed graphs.* Traditionally, each information packet is regarded as an unsplittable commodity [8] 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 for not only information relaying but also information mixing at the intermediate nodes [1], [16], [19], [20], which has been shown to achieve significant throughput advantages over simple non-coded solutions.

For a *single multicast session* problem, namely, when all sinks are interested in the same set of information flowing along the network, network coding is performed on packets of the same session and is thus termed *intrasession network coding*. The characterization of the corresponding capacity region is well understood. For directed cyclic/acyclic graphs, network coding is able to achieve the long standing min-cut/max-flow bound, and one can further restrict the choice of allowed coding operations and use only *linear network codes* without loss of efficiency [4], [16], [19]. For undirected graphs, network coding is able to approximate a similar cut bound within a constant factor of two [20].

The setting that different sinks are interested in different subsets of information is equivalent to the coexistence of multiple sessions competing for network resources. The benefit of network coding across multiple multicast sessions, generally termed *intersession network coding*, is clearly demonstrated in the butterfly graph of [1], [14], [19]. Nonetheless, with multiple coexisting sessions, a network coding solution has to balance the cooperative coding efforts of the intermediate nodes with the conflicting objectives of maximizing the throughput of each individual session, which significantly complicates the problem. For example, linear network coding is no longer throughput optimal for multiple multicast sessions [5]. Even for simple *directed acyclic graphs*, deciding the existence of a linear intersession network coding solution can be NP-complete [18] rather than polynomial time [22]. Until now, no coding gain has been observed in any undirected graph [21].

The capacity region or equivalently the feasibility of intersession network coding is also less understood for multiple multicast sessions. Only for very special graphs can the capacity region be characterized, such as directed cycles [10], degree-2 three-layer directed acyclic networks [29], and special bipartite undirected graphs [10]. For general arbitrary graphs, the intrinsic hardness of the problem hampers a full understanding of intersession network coding. The capacity characterization has been studied from an information-theoretic perspective, including the fundamental regions in the entropy space [24], entropy calculus [13], and an entropy-based description of the achievable region [30]. Several graph-theoretic outer bounds on the capacity region have been devised based on the *generalized edge cut* conditions: the network-sharing bound for acyclic networks [29], the information dominance condition for acyclic networks [10], and the edge-cut bounds for directed networks [17]. The achievability results, i.e., the inner bound on the capacity region, is determined generally by linear programming in a fashion similar to that of solving fractional multi-commodity flow, including the butterfly-based construction [26], the pollution-treatment with powerset-based flow division [28], and the analysis of a practical intersession network coding protocol [23]. The capacity region of a special

Manuscript received ?? ??, 2008; revised ?? ??, 2008.

This work was supported in parts by the NSF Awards CNS-0721236, CCF-0845968, CNS-0905408, and CNS-0905331, and ARO award W911NF-08-1-0238. Part of this paper was presented at the IEEE International Symposium on Information Theory, Nice, France, 2007 and at the 45th Annual Allerton Conference on Communications, Control, and Computing, Monticello, IL, 2007.

C.-C. Wang is with the Center for Wireless Systems and Applications (CWSA), School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA. E-mail: chihw@purdue.edu.

N. B. Shroff is with the Departments of ECE and CSE, The Ohio State University, Columbus, OH 43210, USA. E-mail: shroff@ece.osu.edu.

class of *conservative* network coding is studied in [9].

Studying general intersession network coding on special classes of directed graphs has relatively limited practical applicability to real communication networks. In this paper, we study the characterization of *pairwise intersession network coding* (PINC), which allows for *arbitrary directed acyclic/cyclic networks*, but restricts the coding operations to being between two symbols (for acyclic networks) or between two strings of symbols (for cyclic networks). More explicitly, source nodes s_1 and s_2 would like to transmit two symbols X and Y (resp. two strings of symbols $X_1, \cdots, X_t$ and $Y_1, \cdots, Y_t$) to two groups of destinations $\mathbf{d}_1 = \{d_{1,i}\}_i$ and $\mathbf{d}_2 = \{d_{2,j}\}_j$ in a directed acyclic/cyclic network, and each group is interested in one symbol (or one string of symbols), respectively. Destination sets $\mathbf{d}_1$ and $\mathbf{d}_2$ may be disjoint or not. One possible application of this setting is multi-resolution video multicast by network coding, for which $s_1 = s_2 = s$, X and Y represent the low and high resolution bits respectively, and $\mathbf{d}_1 \supseteq \mathbf{d}_2$. Potentially, PINC is an appealing solution for delay-sensitive data, such as video streaming, since the larger the number of the to-be-coded sessions, the longer the delay in an asynchronous network.

It is well known that for two multicast sessions, the existence of a non-coded scheme is equivalent to the existence of two *edge-disjoint* sets of paths (or more precisely, two sets of pairwise edge-disjoint Steiner trees). In this work, we first prove an analogous result that the existence of a PINC scheme is equivalent to the existence of sets of paths with *controlled edge overlaps*, which includes the well-studied butterfly graph as a special case. This result bridges the gap between the characterization theorems of multiple-session non-coded solutions and of single-session network-coded solutions for the non-trivial case of two source symbols. Based on this new characterization, various aspects of PINC are studied, including the sufficiency of linear codes, the complexity of identifying coding opportunities, its topological analysis, and bandwidth- and coding-optimal conditions (defined and discussed in Section III.C.4). For example, for directed cyclic networks, deciding the existence of a non-coded solution, or equivalently deciding the existence of a pair of edge-disjoint paths, is an NP-complete problem. Nonetheless, we have shown that deciding the existence of a PINC solution takes only polynomial time, a dramatic complexity improvement using network coding. Another feature of the proposed new characterization is its *flow*-based form in contrast with the cut-based form of the existing outer bounds. The constructive nature of the flow-based form admits efficient design of practical PINC schemes including the *sequentially-reset scheme* used in this work when proving the achievability part of the characterization. Other practical implementations taking advantages of the flow-based form can be found in [15].

The remainder of this paper is organized as follows. Section II discusses the setting and formulations for both acyclic and cyclic directed networks and provides some basic graph-theoretic definitions and notations for completion. Section III presents the main characterization results for PINC, which will be stated for acyclic and cyclic networks separately. Several examples will be used for illustration purpose. Various implications of the new characterization will also be discussed in Section III. Section IV contains the complete proofs of the main characterization theorems. Proofs regarding the complexity and the topological analyses will be provided in Sections V and VI, respectively. Section VII concludes this paper.

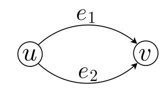


Fig. 1. Illustration of parallel edges.

II. Settings and Formulation

A. Basic Graph-Theoretic Definitions and Notations

Consider a finite-sized directed graph $G = (V, E)$, where V is the set of nodes and E is the set of directed edges. The size of G is denoted by $|G| = |V| + |E|$. Each edge $e \in E$ can be represented by $e = uv$, where $u = \mathsf{tail}(e)$ and $v = \mathsf{head}(e)$ are the tail and head of edge e, respectively. We allow the existence of *parallel edges*, namely two edges e_1 and e_2 may share the same tail and head nodes[1] (see Fig. 1). A *loop* $e = uu$ is also allowed although its impact on the network is little (one can remove all loops in G without loss of generality).

A *path* (or a *walk*) P is a sequence of edges $e_1 e_2 \cdots e_k$ such that $\mathsf{head}(e_i) = \mathsf{tail}(e_{i+1})$ for $i = 1, \cdots, k-1$ and is sometimes denoted by a sequence of nodes $u_0 u_1 \cdots u_k$. The distinction between a path and a walk is that a walk may contain *cycles* while a path is always free of any cycle, namely, $u_i \neq u_j$ for $i \neq j$. For acyclic graphs, the definition of a walk and a path is interchangeable. We use $v \in P$ (resp. $e \in P$) to indicates that a node v (resp. an edge e) is used by P. $V(P)$ denotes the vertex set used by a given path P. Similarly $E(P)$ denotes the edge set of P. For any two sets of nodes $S, T \subset V$, we say a path is from S to T if it starts from an $s \in S$ and ends in a $t \in T$.

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 $\mathsf{ncp}_{\mathcal{P}}(e) \triangleq |\{P \in \mathcal{P} : e \in P\}|$, i.e., the number of paths that use edge e. We often use $P_{u,v}$ (or $Q_{u,v}$) to denote a path starting from node u and ending at node v. A node v is *reachable* from u if there exists a $P_{u,v}$ path. An edge e_2 is *reachable* from e_1 if $\mathsf{tail}(e_2)$ is reachable from $\mathsf{head}(e_1)$. For acyclic networks, we sometimes say that v is a downstream node of u (or u is an upstream node of v) if v is reachable from u. The downstream/upstream edges can be defined similarly. A *k-edge cut* separating node sets $\mathbf{u} \subseteq V$ and $\mathbf{v} \subseteq V$ is a collection of k edges such that after the removal of those k edges, there is no path connecting $u \in \mathbf{u}$ and $v \in \mathbf{v}$ for any u and v. Two paths are *edge-disjoint* if they share no common edge. Two paths are *independent* if they are interior-vertex-disjoint.

For two paths P and Q and three nodes x, y, and z, xPy denotes the path segment connecting nodes x and y using

[1]To rigorously represent parallel edges, one has to use the incidence mapping $A : E \mapsto V^2$. For simplicity, we describe an edge e by its head and tail unless when we have to distinguish parallel edges.

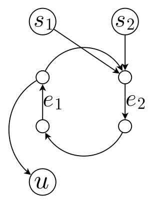


Fig. 2. Illustration of the critical 1-edge cut. Both e_1 and e_2 are 1-edge cuts while e_2 is the unique *critical 1-edge cut.*

path P. Similarly, $xPyQz$ denotes a walk connecting nodes x, y, and z while P and Q are used during the x–y and y–z segments, respectively. For cyclic graphs, $xPyQz$ may contain cycles and thus only be a walk even when both P and Q are valid paths. The above notation for path concatenation can be generalized for concatenating multiple paths such as $x_0P_1x_1 \cdots x_{k-1}P_kx_k$.

We also define the *critical 1-edge cut* as follows.

Definition 1: For any node u in a directed acyclic / cyclic network, the critical 1-edge cut e^* is a 1-edge cut separating $\{s_1, s_2\}$ and u that is the *farthest* from u. That is, any path from $\{s_1, s_2\}$ to u will meet cut e^* before meeting any other 1-edge cut.

Fig. 2 illustrates the concept of the critical 1-edge cut. Both e_1 and e_2 are 1-edge cuts while e_2 is the unique critical 1-edge cut. Note that for acyclic networks, the concept of being the *farthest* is unambiguous as one can use the topological order of the edges to define the farthest edge. The following lemma proves the existence and the uniqueness of the critical 1-edge cut even in a cyclic network.

Lemma 1: If there exists a 1-edge cut separating $\{s_1, s_2\}$ and u, then e^* always exists and is unique.

Proof: For cyclic networks, one needs to show that the *farthest* 1-edge cut is well-defined and does not change when different paths are considered.

Choose an arbitrary path P connecting $\{s_1, s_2\}$ and u. By definition, all 1-edge cuts are in P. Choose e^* that is the farthest from u according to their order in the path P. Suppose that there exists another path P' from $\{s_1, s_2\}$ to u that meets another 1-edge cut e before e^*. Since $e \in P$, $P'eP$ is a walk from $\{s_1, s_2\}$ to u without using e^*, which contradicts the construction that e^* is a 1-edge cut. Therefore, such e^* must be unique. ■

B. PINC on Directed Acyclic Networks

We consider the following network coding problem with two simple multicast sessions. Given a finite directed acyclic graph $G = (V, E)$, two source nodes s_1 and s_2, and two groups of destination nodes $\mathbf{d}_1 = \{d_{1,i}\}_i$ and $\mathbf{d}_2 = \{d_{2,j}\}_j$, *whether two symbols X and Y, emanated from s_1 and s_2 respectively, can be transmitted simultaneously to $d_{1,i}$ and $d_{2,j}$ for all $d_{1,i}$ and $d_{2,j}$ within a single time slot*, say a second, *using network coding*? It is assumed that *each edge is capable of carrying one symbol per second*, and there is no propagation delay. High-rate links are modelled by parallel edges. For comparison, if network coding is prohibited, the existence of a non-coded solution is equivalent to the existence of two edge-disjoint paths (for two unicast sessions) and of two edge-disjoint Steiner trees (for two multicast sessions). This work studies the characterization when network coding is allowed.

Without loss of generality, we assume that each s_1 and s_2 has only one outgoing edge and no incoming edges. If not, say s_1 has more than one outgoing edge or has at least one incoming edge, we add an additional node s_1' and an additional edge $e' = s_1's_1$ to the graph G. After the addition, use s_1' as the new source node and treat the old s_1 as an ordinary intermediate node. Any network coding solution for the old graph can be mapped bijectively to a network coding solution for the new graph. Similarly, without loss of generality, we assume that each $d_{1,i}$ and $d_{2,j}$ has only one incoming edge and no outgoing edge. By the same technique of adding auxiliary nodes and edges, we can further assume that $s_1 \neq s_2$ and all $d_{1,i}$ and $d_{2,j}$ are distinct without loss of generality. For notational simplicity, we sometimes use interchangeably s_1 as the source *node* or as the unique outgoing *edge* of the source. Similarly, s_2, $d_{1,i}$, and $d_{2,j}$ denote interchangeably the nodes or their uniquely associated edges.

Both symbols X and Y are drawn from a finite field $\mathsf{GF}(q)$ with sufficiently large q (see [11], [16] 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 and Y take integer values instead, provided a sufficiently large q is adopted. Throughout the paper, we also assume that $P_{s_1,d_{1,i}}$ and $P_{s_2,d_{2,j}}$ exist for all $d_{1,i} \in \mathbf{d}_1$ and $d_{2,j} \in \mathbf{d}_2$, which can be checked within polynomial time. Otherwise network communication is simply impossible.

C. PINC on Directed Cyclic Networks

For cyclic networks, we follow the basic two-multicast setting $(s_1, \mathbf{d}_1)$ and $(s_2, \mathbf{d}_2)$ in the previous subsection. However, to be consistent with the law of causality, propagation delay in a cyclic network has to be handled with care and we adopt the setting of *slotted transmission*. For a given directed cyclic network G, we assume that each edge is capable of sending one packet per second and the propagation delay is also one second. High-rate links are modelled by parallel edges. Large-delay links are modelled by longer paths with added auxiliary interior nodes. We consider the following PINC problem on cyclic networks.

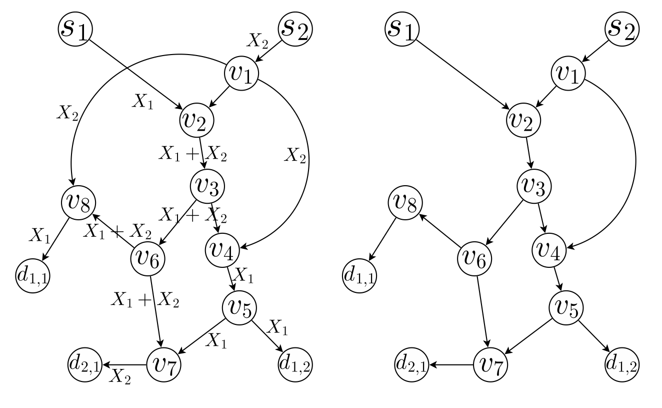
Source s_1 sends a string of T i.i.d. packets $X_1, \cdots, X_T$ to destination $d_{1,i} \in \mathbf{d}_1$ in a duration of T seconds. Source s_2 sends i.i.d. packets $Y_1, \cdots, Y_T$ to $d_{2,j} \in \mathbf{d}_2$. Each packet is in $\mathsf{GF}(q)$ for some q. Let $M_{e,t} \in \mathsf{GF}(q)$ denote the coded symbol sent along edge e in the t-th second, and we use $[M_e]_1^t \triangleq \{M_{e,\tau} : \tau = 1, \cdots, t\}$ to denote the collection of coded symbols in the first t seconds. Destinations $\{d_{1,i}\}_i$ and $\{d_{2,j}\}_j$ receive $[M_{d_{1,i}}]_1^T$ and $[M_{d_{2,j}}]_1^T$ in the entire duration. We say that *a cyclic network G admits a network coding solution with asymptotic rate 1 if for any $\epsilon > 0$, there exists a sufficiently large T such that there exists a PINC scheme satisfying the following inequalities for all $d_{1,i} \in \mathbf{d}_1$ and*

$d_{2,j} \in \mathbf{d}_2$:

$$\frac{1}{T} I([X]_1^T; [M_{d_{1,i}}]_1^T) > (1-\epsilon)\log(q)$$
$$\text{and } \frac{1}{T} I([Y]_1^T; [M_{d_{2,j}}]_1^T) > (1-\epsilon)\log(q),$$

where $I(\cdot;\cdot)$ is the *mutual information*. It is worth noting that when each edge induces a delay, it is generally impossible to achieve *an exact rate-1* transmission (unless the source and destination are only 1-hop away) and the ϵ-term has to be included for the asymptotic analysis. For the following, we sometimes drop the adjective "asymptotic" for simplicity.

For future reference, we use "2-multicast acyclic networks" to denote the problem of finding network coding (or non-coded) solutions for directed acyclic networks with two multicast sessions. Similarly, we use the following terms "2-multicast cyclic networks," "2-unicast acyclic networks," and "2-unicast cyclic networks" to refer to different settings of interest. Note that unicast is a special case of multicast. All our results regarding the case of "2-multicast" hold for the case of "2-unicast" as well.



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

Fig. 3. Two examples demonstrating Theorem 2. (a) satisfies Theorem 1 and the corresponding network coding solution is as described. (b) does not satisfy Theorem 1 and no network coding solution exists for (b). Note if only one of $d_{1,1}$ and $d_{1,2}$ in (b) is requesting symbol X, then finding a non-coded/network coding solution is possible. If both are requesting X, then simultaneous transmission becomes impossible.

III. Main Results

We first state the characterization theorems for PINC on acyclic and cyclic networks, and then discuss their corresponding implications. All proofs of the theorems and propositions are provided in Sections IV to VI.

A. Characterization for Acyclic Networks

Theorem 1 (Cut-Based Characterization): For 2-multicast acyclic networks, the existence of a network coding solution is equivalent to the following condition being satisfied: Let $e_i = u_i v_i$ denote[2] the *critical 1-edge cut* of $d_{1,i} \in \mathbf{d}_1$ and similarly let $e_j = u_j v_j$ denote the critical 1-edge cut of $d_{2,j} \in \mathbf{d}_2$. Then any $d_{1,i} \in \mathbf{d}_1$ is reachable from s_1 using only edges in $G \backslash \{e_j : d_{2,j} \in \mathbf{d}_2\}$ and any $d_{2,j} \in \mathbf{d}_2$ is reachable from s_2 using only edges in $G \backslash \{e_i : d_{1,i} \in \mathbf{d}_1\}$.

Take Fig. 3 for example in which $\mathbf{d}_1 = \{d_{1,1}, d_{1,2}\}$ and $\mathbf{d}_2 = \{d_{2,1}\}$. In Fig. 3(a), the critical 1-edge cuts for $\mathbf{d}_1$ and $\mathbf{d}_2$ are:

$$e_{1,1} = v_8 d_{1,1},\ e_{1,2} = v_4 v_5,\ \text{and } e_{2,1} = v_7 d_{2,1}.$$

It can be checked that the conditions in Theorem 1 hold for such $e_{1,1}$, $e_{1,2}$, and $e_{2,1}$. A network coding solution is illustrated in Fig. 3(a). On the other hand, in Fig. 3(b), the critical 1-edge cuts for $\mathbf{d}_1$ and $\mathbf{d}_2$ are:

$$e_{1,1} = v_2 v_3,\ e_{1,2} = v_4 v_5,\ \text{and } e_{2,1} = v_7 d_{2,1},$$

and $d_{2,1}$ is not reachable from s_2 using only edges in $G \backslash \{e_{1,1}, e_{1,2}\}$. Theorem 1 guarantees that no coding solution exists for Fig. 3(b).

It is worth noting that Fig. 3(b) is a non-trivial example and satisfies the generalized cut-conditions in [10], [29] even when no network coding solution exists. Moreover, if only one of $d_{1,1}$ and $d_{1,2}$ is requesting X and the other one is dormant, the degenerate two-unicast problem becomes feasible. Only when both $d_{1,1}$ and $d_{1,2}$ are requesting symbol X simultaneously, does the two-multicast problem become infeasible.

The above cut-based description facilitates upper bounding the achievable rates (the necessary condition). In the following, we provide an equivalent path-based characterization that is more helpful for explicit code construction (the sufficient condition) and practical implementation [15].

Theorem 2 (Flow-Based Characterization): For 2-multicast acyclic networks, the existence of a network coding solution is equivalent to the existence of $|\mathbf{d}_1| + |\mathbf{d}_2| + 2$ collections of paths: $\mathcal{P}_i, \forall d_{1,i} \in \mathbf{d}_1$, $\mathcal{Q}_j, \forall d_{2,j} \in \mathbf{d}_2$, $\mathcal{Q}_I$, and $\mathcal{P}_J$ such that

$$\forall d_{1,i} \in \mathbf{d}_1,\ \mathcal{P}_i = \begin{cases} \{P_{s_1,d_{1,i}}, P_{s_2,d_{1,i}}\} & \text{if } \exists \text{ a } (s_2, d_{1,i}) \text{ path} \\ \{P_{s_1,d_{1,i}}\} & \text{otherwise} \end{cases}$$
$$\forall d_{2,j} \in \mathbf{d}_2,\ \mathcal{Q}_j = \begin{cases} \{Q_{s_2,d_{2,j}}, Q_{s_1,d_{2,j}}\} & \text{if } \exists \text{ a } (s_1, d_{2,j}) \text{ path} \\ \{Q_{s_2,d_{2,j}}\} & \text{otherwise} \end{cases}$$
$$\mathcal{Q}_I = \{Q_{s_1,d_{1,i}} : \forall d_{1,i} \in \mathbf{d}_1\}$$
$$\mathcal{P}_J = \{P_{s_2,d_{2,j}} : \forall d_{2,j} \in \mathbf{d}_2\}$$

satisfy the following *controlled edge-overlap conditions*.

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

Continue our example in Fig. 3(a). The following selections of $\mathcal{P}_1 = \{P_{s_1,d_{1,1}}, P_{s_2,d_{1,1}}\}$, $\mathcal{P}_2 = \{P_{s_1,d_{1,2}}, P_{s_2,d_{1,2}}\}$, $\mathcal{Q}_1 = \{Q_{s_2,d_{2,1}}, Q_{s_1,d_{2,1}}\}$, $\mathcal{Q}_J = \{Q_{s_1,d_{1,1}}, Q_{s_1,d_{1,2}}\}$, and $\mathcal{P}_I = \{P_{s_2,d_{2,1}}\}$ with

[2]By our assumption in Section II, any destination d has only one incoming edge. Therefore, the only incoming edge entering d itself is a 1-edge cut separating $\{s_1, s_2\}$ and d. By Lemma 1, the critical 1-edge cut of destination d always exists.

$$
\begin{aligned}
P_{s_1,d_{1,1}} &= s_1 v_2 v_3 v_6 v_8 d_{1,1} \\
P_{s_2,d_{1,1}} &= s_2 v_1 v_8 d_{1,1} \\
P_{s_1,d_{1,2}} &= s_1 v_2 v_3 v_4 v_5 d_{1,2} \\
P_{s_2,d_{1,2}} &= s_2 v_1 v_4 v_5 d_{1,2} \\
P_{s_2,d_{2,1}} &= s_2 v_1 v_2 v_3 v_6 v_7 d_{2,1}
\end{aligned}
\qquad
\begin{aligned}
Q_{s_2,d_{2,1}} &= s_2 v_1 v_4 v_5 v_7 d_{2,1} \\
Q_{s_1,d_{2,1}} &= s_1 v_2 v_3 v_6 v_7 d_{2,1} \\
Q_{s_1,d_{1,1}} &= s_1 v_2 v_3 v_6 v_8 d_{1,1} \\
Q_{s_1,d_{1,2}} &= s_1 v_2 v_3 v_4 v_5 d_{1,2}
\end{aligned}
$$

satisfy Theorem 2.

When only two single destinations are considered $\mathbf{d}_1 = \{d_1\}$ and $\mathbf{d}_2 = \{d_2\}$, Theorem 2 collapses to that for 2-unicast acyclic networks:

Corollary 1: For 2-unicast acyclic networks, a network coding solution exists if and only if one of the following two conditions holds.

- There exist two edge-disjoint paths P_{s_1,d_1} and P_{s_2,d_2}.
- There exist a collection $\mathcal{P}$ of three paths P_{s_1,d_1}, P_{s_2,d_2}, and P_{s_2,d_1}, and a collection $\mathcal{Q}$ of three paths Q_{s_1,d_1}, Q_{s_2,d_2}, and Q_{s_1,d_2}, such that $\max_{e\in E} \mathsf{ncp}_{\mathcal{P}}(e) \le 2$ and $\max_{e\in E} \mathsf{ncp}_{\mathcal{Q}}(e) \le 2$.

Proof: This corollary is a straightforward restatement of Theorem 2 for 2-unicast acyclic networks. ■

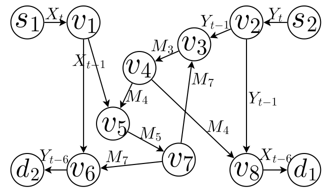
Another interesting case is when $\mathbf{d}_1 = \mathbf{d}_2$,[3] i.e. all destinations are interested in both symbols. Condition 1 of Theorem 2 guarantees the min-cut between $d_{1,i}$ and $\{s_1, s_2\}$ is two. Therefore, Theorem 2 collapses to the classic min-cut/max-flow characterization for the single-session problem.

The intuition behind Theorem 1 is that the critical 1-edge cuts must carry pure, non-corrupted symbols for their corresponding destinations. The feasibility of a network coding solution thus requires the existence of a path connecting s_i and d_i but not using any critical 1-edge cuts of the destinations in the other session $d_j \in \mathbf{d}_2$. This corresponds to the necessary part of the cut-based characterization of PINC. On the other hand, Corollary 1 and Theorem 2 show that as long as we can identify paths with *controlled edge overlaps* (as described in Conditions 1 and 2 of Theorem 2), then we can carry the desired information along those paths by network coding. Since these paths have controlled edge overlap, network coding ensures that the packets can be mixed with other sessions in a way that the interference can be removed in a later stage. The proofs of these theorems in Section IV follow this intuition.

B. Characterization for Cyclic Networks

Most existing theoretic studies of network coding focus on DAGs due to its simpler structure and due to the fact that one can always convert a cyclic network to its acyclic counterpart by taking into account the time index and the causality of information transmission [19].

Nonetheless, when studying PINC on a cyclic network, *one cannot rely on this cyclic-to-acyclic conversion* since after conversion, there are T symbols in the $(s_1, \mathbf{d}_1)$ (resp. $(s_2, \mathbf{d}_2)$) multicast session and network coding allows complete freedom of mixing the $2T$ symbols $[X]_1^T$ and $[Y]_1^T$. Coding can be performed either within its own session (mixing X_t and X_τ for some $t \ne \tau$) or across different sessions (mixing X_t and Y_τ). More than two symbols will be mixed with each other, which is beyond the scope of Theorem 1. Therefore, simply combining Theorem 1 (for acyclic networks) and the cyclic-to-acyclic conversion does not characterize PINC on a cyclic network. To properly address the behavior of a 2-multicast cyclic network, one has to consider both intrasession (mixing within its own string of symbols) and intersession network coding (mixing across different strings of symbols). For the following, we provide a characterization theorem for 2-unicast cyclic networks, which takes into account the interplay between intra- and intersession network coding.



$$
\begin{aligned}
M_3 &= X_{t-4} + Y_{t-2} \\
M_4 &= X_{t-5} + Y_{t-3} \\
M_5 &= X_{t-2} + Y_{t-4} \\
M_7 &= X_{t-3} + Y_{t-5}
\end{aligned}
$$

Fig. 4. Pairwise intersession network coding on cyclic networks.

Theorem 3 (Characterization): Theorem 1 and the equivalent Theorem 2, first stated for 2-multicast acyclic networks, hold for 2-unicast cyclic networks as well.

Fig. 4 illustrates a cyclic network that admits a rate-1 network coding solution. The critical 1-edge cuts are $e_1 = v_8 d_1$ and $e_2 = v_6 d_2$, which satisfy Theorem 1. If we choose the $\mathcal{P} = \{P_{s_1,d_1}, P_{s_2,d_1}, P_{s_2,d_2}\}$ paths and $\mathcal{Q} = \{Q_{s_2,d_2}, Q_{s_1,d_2}, Q_{s_2,d_2}\}$ paths as follows:

$$
\begin{aligned}
P_{s_1,d_1} &= s_1 v_1 v_5 v_7 v_3 v_4 v_8 d_1 & Q_{s_2,d_2} &= s_2 v_2 v_3 v_4 v_5 v_7 v_6 d_2 \\
P_{s_2,d_1} &= s_2 v_2 v_8 d_1 & Q_{s_1,d_2} &= s_1 v_1 v_6 d_2 \\
P_{s_2,d_2} &= s_2 v_2 v_3 v_4 v_5 v_7 v_6 d_2 & Q_{s_1,d_1} &= s_1 v_1 v_5 v_7 v_3 v_4 v_8 d_1.
\end{aligned}
$$

then Theorem 2 is satisfied. Fig. 4 describes the corresponding network-coding solution. The expression along each edge corresponds to the coded symbol along that edge in the t-th second. For example, X_t and Y_t are sent along $s_1 v_1$ and $s_2 v_2$, respectively, in the t-th second. With propagation delay, X_{t-1} and Y_{t-1} are sent along $v_1 v_5$, $v_1 v_6$, $v_2 v_3$, and $v_2 v_8$ respectively in the t-th second. In the beginning of the t-th second, node v_5 has received $(t-1)$ packets $\{X_{\tau-1} : \tau < t\}$ from v_1 and $(t-1)$ packets $\{(X_{\tau-5} + Y_{\tau-3}) : \tau < t\}$ from v_4. By its knowledge about X_{t-6}, X_{t-2}, and $(X_{t-6} + Y_{t-4})$, v_5 generates coded packet $M_5 = X_{t-2} + Y_{t-4}$ and sends it along edge $v_5 v_7$ in the t-th second. Following the coding operations depicted in Fig. 4, an asymptotic rate of 1 packet per second can be achieved. Note that at node v_5, both packets X_{t-6} and X_{t-2} from the same session are used to encode the outgoing symbol M_5. That is, in Fig. 4, both intra- and intersession network coding are performed in order to achieve rate-1 transmission.

The proof of the necessary condition follows from the intuition that the critical 1-edge cuts must carry non-corrupted symbols, similar to that of the acyclic networks. On the other

[3] Since we assume previously that all destination nodes are distinct and with only one incoming edge, we obviously will not have $\mathbf{d}_1 = \mathbf{d}_2$. To be consistent, a more precise statement is that $\{\mathsf{tail}(d_{1,i}) : \forall d_{1,i} \in \mathbf{d}_1\} = \{\mathsf{tail}(d_{2,j}) : \forall d_{2,j} \in \mathbf{d}_2\}$. Namely, *at each physical location* $\mathsf{tail}(d_{1,i}) = \mathsf{tail}(d_{2,j})$*, both symbols X and Y have to be decoded successfully.*

hand, the looping nature of cyclic networks poses additional challenges for the achievability analysis. In particular, a special topology has been identified, which is unique for cyclic networks and requires careful coordination of both intra- and intersession network coding. We conclude this subsection by stating a necessary condition for the 2-multicast cyclic networks and leave the achievability direction as a conjecture.

Theorem 4 (Necessary Condition): For 2-multicast cyclic networks, the necessary part of Theorem 1 (or equivalently the necessary part of Theorem 2) still holds. Namely, if there exists a rate-1 network coding solution, then there exist $|\mathbf{d}_1|+|\mathbf{d}_2|+2$ collections of paths satisfying Theorem 2.

Conjecture 1 (Sufficient Condition): For any 2-multicast cyclic network, if there exist $|\mathbf{d}_1| + |\mathbf{d}_2| + 2$ collections of paths satisfying Theorem 2, then there exists a rate-1 network coding solution.

C. Implications

Several implications of Theorems 1 to 3 are discussed as follows.

C.1 — Sufficiency of Linear Codes

Corollary 2: Linear network codes are sufficient both for 2-multicast acyclic networks and for 2-unicast cyclic networks.

Proof: This corollary is a byproduct of the proof of the achievability of Theorems 1 to 3. ■

C.2 — Complexity of Deciding the Feasibility of PINC

Since finding the cuts is a polynomial time task, the cut-based characterization in Theorem 1 enables efficient algorithms that decide the feasibility of a rate-1 PINC solution:

Proposition 1: Deciding the existence of a network coding solution for 2-multicast acyclic networks is a polynomial-time problem with respect to $(|G| + |\mathbf{d}_1| + |\mathbf{d}_2|)$.

In [18], it is shown that when the number of coexisting sessions N is unbounded, deciding the existence of a network coding solution for N-unicast acyclic networks is an NP-complete problem. Proposition 1 shows that when the number of coexisting sessions is bounded by two, such a problem has polynomial-time complexity with respect to the size of G and the total number of destinations.

A more intriguing complexity result is for 2-unicast cyclic networks.

Proposition 2: Deciding the existence of a network coding solution for the 2-unicast cyclic networks is a polynomial-time problem with respect to $|G|$.

The proofs of Propositions 1 and 2 are relegated to Section V.

Cycles in a network have long provided unique challenges for graph-theoretic studies. For example, when a non-coded solution is used, the existence of a rate-1 solution is equivalent to the existence of two edge-disjoint paths connecting (s_1, d_1) and (s_2, d_2) respectively. Therefore, deciding the existence of a non-coded solution is equivalent to deciding the existence of two edge-disjoint paths. In [6], it is shown that for 2-unicast acyclic networks, such a task is polynomial-time with respect to $|G|$ but for 2-unicast cyclic networks, such a task is NP-complete when $s_1 \neq s_2$ and $d_1 \neq d_2$. There is a significant complexity gap between acyclic and cyclic networks. *In contrast, when network coding is allowed, one can decide the existence of a rate-1 network coding solution in polynomial time even for cyclic networks.* A similar complexity reduction of network coding was first observed in [20].

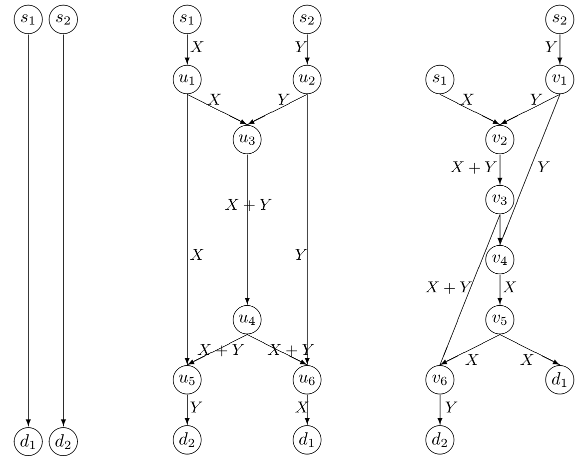


Fig. 5. Three classes of graphs for which simultaneous transmission of X and Y along (s_1, d_1) and (s_2, d_2) is feasible via network coding or non-coded solutions.

We believe this complexity reduction is a universal advantage of both inter- and intrasession network coding over non-coded solutions. In [3], it is shown that for a single multicast session with one source s and two distinct destinations $\mathbf{d} = \{d_1, d_2\}$ in a directed cyclic network, deciding whether there exists a non-coded scheme that sends two symbols simultaneously from s to both destinations d_1 and d_2 is an NP-complete problem. However, when coding is allowed, one only needs to test whether both the max-flow values for (s, d_1) and for (s, d_2) are no less than two, which is a polynomial time task. Proposition 2 and this example suggests that both intra- and intersession network coding enable easier determination of the corresponding feasibility.

C.3 — Minimal Bandwidth Requirements

Corollary 3: For 2-multicast acyclic networks and for 2-unicast cyclic networks, if there exists a PINC solution, then there exists a network coding solution using at most $3|\mathbf{d}_1| + 3|\mathbf{d}_2|$ paths during transmission.

Proof: Corollary 3 is a straightforward result from Theorems 2 and 3, since one can remove any edges not used by the $3|\mathbf{d}_1|+3|\mathbf{d}_2|$ paths in $\mathcal{P}_i, \forall d_{1,i} \in \mathbf{d}_1$, $\mathcal{Q}_j, \forall d_{2,j} \in \mathbf{d}_2$, $\mathcal{Q}_I$ and $\mathcal{P}_J$. By Theorems 2 and 3, a network coding solution exists for the trimmed graph as well. ■

For the special case of 2-unicast acyclic networks, the above bandwidth statement can be further strengthened by the topological analysis presented in the following subsection.

C.4 — Topological Analyses for 2-Unicast Acyclic Networks

Consider three simple graphs: the 2 edge-disjoint paths (2-EDPs) (see Fig. 5(a)), the butterfly (Fig. 5(b)), and a special graph termed *the grail* (Fig. 5(c)). As described in the corresonding figures, for all these three graphs, it is feasible to simultaneously send two symbols X and Y from s_1 and s_2 to d_1 and d_2 respectively. In parallel, Corollary 1 proves the equivalence between the feasibility of network coding and

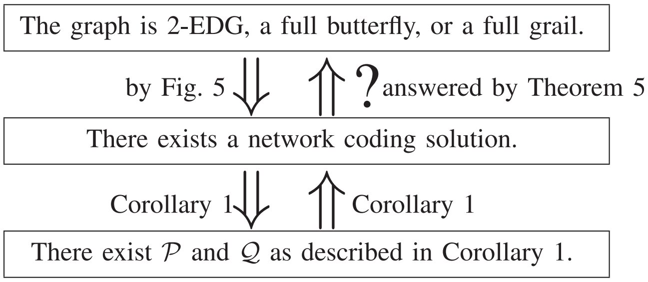
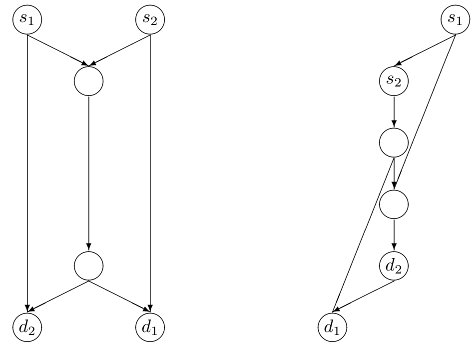


Fig. 6. Comparison of Corollary 1 and Theorem 5.



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

Fig. 7. Two examples of graphs that can be obtained from the full butterfly and the full grail by edge contraction.

the existence of two special path collections $\mathcal{P}$ and $\mathcal{Q}$. Fig. 6 summarizes our knowledge about 2-unicast acyclic networks, in which three arrows have been established, except the upper right one. We will complete the picture by showing that *for a 2-unicast acyclic network, if a network coding solution exists, then the graph must "contain" one of the three simple graphs: 2-EDPs, the butterfly, and the grail.* This statement is made rigorous in the following theorem.

Theorem 5: Consider any 2-unicast acyclic network and assume that all sources and destinations have degree one. If a network coding solution exists, then one of the following two conditions must hold.

1) There exist two EDPs connecting (s_1, d_1) and (s_2, d_2).
2) Graph G contains a subgraph G' that can be obtained from a full butterfly (Fig. 5(b)) or from a full grail (Fig. 5(c)) by replacing the edges with *independent paths* connecting the corresponding ends.[4] A more formal statement can be made by the language of subgraph homeomorphism [6]. More explicitly, the sources and destinations of the butterfly (resp. the grail) are mapped to the sources and destinations of G and other intermediate nodes of the butterfly (resp. the grail) can be mapped to any node in G. The subgraph G' is homeomorphic to the butterfly (resp. the grail).

The proof of Theorem 5 is based on the topological analysis of the path-/cut-based characterization theorem. More explicitly, it relies on the relative positions of the critical 1-edge cuts. When the two critical 1-edge cuts e_1 and e_2 are *parallel* to each other, then the network corresponds to the butterfly structure. (In Fig. 5(b), the critical 1-edge cuts $e_1 = u_6 d_1$ and $e_2 = u_5 d_2$ are parallel to each other.) When one critical 1-edge cut (e_1) is *upstream* of the other (e_2), then the network corresponds to the grail structure. (In Fig. 5(c), the critical 1-edge cut $e_1 = v_4 v_5$ is upstream of $e_2 = v_6 d_2$.) The detailed proof is relegated to Section VI.

Fig. 7 contains two other examples that admit network coding solutions and we note that the sources and destinations in Fig. 7 have degrees more than one. After converting Fig. 7 so that all sources and destinations have degree one (as discussed in Section II-B), the two examples in Fig. 7 are indeed the full butterfly and the full grail as predicted in Theorem 5.

Theorem 5 says that if a network coding solution exists, then it is either the case that a non-coded solution also exists (with 2-EDPs) or the case in which the graph contains a subgraph that is topologically identical to a butterfly or to a grail. The closest counterpart of Theorem 5 for intrasession network coding is the information-decomposition arguments in [7], which specify the minimal coding-based topology. There are at least three fundamental differences between Theorem 5 and the results in [7]. First, the subject of the former is the 2-unicast acyclic network while the latter focuses on the 1-multicast acyclic network. Second, Theorem 5 is obtained from the pure graph-theoretic argument rather than coding-based simplification. Third, Theorem 5 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 [7], the question of whether two paths share a vertex is irrelevant as long as these two paths are edge disjoint. For comparison, Theorem 5 makes a stronger statement on the vertex-disjointness via the notion of independent paths.

Theorem 5 can be translated to the corresponding network coding statement. Namely, the network coding solutions for Figs. 5(b) and 5(c) can be applied in a straightforward manner to the graph G by letting the independent paths in $G' \subseteq G$ carry the same messages as the corresponding edges in Figs. 5(b) and 5(c). From the above observation, we have the following corollaries.

Corollary 4: If there exists any network coding solution for the 2-unicast acyclic network, there exists a *binary* linear network coding solution as well. Furthermore, either a non-coded solution exists or one needs exactly three binary exclusive OR (XOR) operations for the entire network encoding/decoding solution.

Proof: This corollary is proven in a straightforward manner by Theorem 5 and by observing that there are exactly three binary XOR operations in the cases of a butterfly and a grail. The discussions on the number of coding nodes for intrasession network coding can be found in [2]. ■

Theorem 5 also implies the following results on the bandwidth and coding efficiency of network coding solutions.

Definition 2: Consider two network coding solutions NC_1

[4]Unlike the butterfly, the grail graph is asymmetric for source-destination pairs (s_1, d_1) and (s_2, d_2). A mirrored image of the grail can be obtained by swapping (s_1, d_1) and (s_2, d_2) of Fig. 5(c). For simplicity, Theorem 5 does not explicitly discuss the mirrored image of the grail but the inclusion of the mirrored image of the grail is implicitly implied.

and NC_2 for the same 2-unicast acyclic network 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 use $G_2 = (V_2, E_2) \subseteq G$ 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 $q_1 \leq q_2$.

A network coding solution is bandwidth-optimal (resp. coding-optimal) if it is more bandwidth-efficient (resp. coding-efficient) than any other network coding solution with the same support rates.

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

We close this section by stating the *source-sink reciprocity* for 2-unicast acyclic and cyclic networks.

Corollary 6 (Source-sink Reciprocity): Consider the communication problem of a 2-unicast acyclic/cyclic network in the reverse direction. Namely, construct G_R by reversing the orientation of all edges in G and swap the roles of the source s_i and destination d_i for $i = 1, 2$. A network coding solution for G exists if and only if a network coding solution for G_R exists. In other words, the feasibility in one orientation implies the feasibility in the reverse orientation.

Proof: Suppose there exists a network coding solution for the forward direction. Then there exist path collections $\mathcal{P}$ and $\mathcal{Q}$ satisfying Corollary 1. Once the orientation is reversed, the path collections become $\mathcal{P}' = \{P'_{d_1,s_1}, P'_{d_1,s_2}, P'_{d_2,s_2}\}$ and $\mathcal{Q}' = \{Q'_{d_2,s_1}, Q'_{d_2,s_2}, Q'_{d_1,s_1}\}$. Since both $\mathcal{P}'$ and $\mathcal{Q}'$ satisfy the controlled edge overlap condition, there exists a network coding solution for the reverse direction. The proof is thus complete. ■

IV. The Proofs of The Characterization Theorems

In this section, we present the proofs of the characterization theorems in Section III.

A. The Equivalence Between Theorems 1 and 2

We first prove the equivalence between the cut-based conditions in Theorem 1 and the path-based conditions in Theorem 2.

Proof of the equivalence between Theorems 1 and 2: Suppose that the cut-based conditions in Theorem 1 are not satisfied. Namely, there exists a d_{1,i_0} such that d_{1,i_0} is not reachable from s_1 using edges in $G\backslash\{e_j : d_{2,j} \in \mathbf{d}_2\}$, or equivalently, any $Q_{s_1,d_{1,i_0}}$ path has to use at least one e_{j_0} for some $d_{2,j_0} \in \mathbf{d}_2$. Since this d_{2,j_0} is reachable from s_1 (using part of the $Q_{s_1,d_{1,i_0}}$ path), the corresponding $\mathcal{Q}_{j_0}$ contains two paths $\{Q_{s_1,d_{2,j_0}}, Q_{s_2,d_{2,j_0}}\}$. Since such e_{j_0} is a critical 1-edge cut, $\mathcal{Q}_{j_0} = \{Q_{s_1,d_{2,j_0}}, Q_{s_2,d_{2,j_0}}\}$ must have $\mathsf{ncp}_{\mathcal{Q}_{j_0}}(e_{j_0}) = 2$, which implies $\mathsf{ncp}_{\mathcal{Q}_{j_0}\cup\{Q_{s_1,d_{1,i_0}}\}}(e_{j_0}) = 3$ for any possible choices of $\mathcal{Q}_{j_0}$. Since for any $Q_{s_1,d_{1,i_0}}$ path there is at least one e_{j_0} such that $\mathsf{ncp}_{\mathcal{Q}_{j_0}\cup\{Q_{s_1,d_{1,i_0}}\}}(e_{j_0}) = 3$, one cannot find $\mathcal{Q}_I$ and $\mathcal{Q}_j$ for all $d_{2,j} \in \mathbf{d}_2$ that satisfy jointly Condition 2 of Theorem 2.

Suppose the cut-based conditions are satisfied. We can construct the path collections $\mathcal{P}_i$, $\mathcal{Q}_j$, $\mathcal{P}_J$, and $\mathcal{Q}_I$ satisfying Theorem 2 in the following way.

For each d_{1,i_0} that is not reachable from s_2, we can arbitrarily choose any $P_{s_1,d_{1,i_0}}$ path and use it as $\mathcal{P}_{i_0}$. Since there are totally only two paths in $\mathcal{P}_{i_0} \cup \{P_{s_2,d_{2,j}}\}$ for any j, Condition 1 of Theorem 2 is satisfied for any choice of $\mathcal{P}_J$. Hence, we only need to focus on the construction for those $d_{1,i}$ that are reachable from both s_1 and s_2.

For each critical 1-edge cut $e_i = u_i v_i$ for $d_{1,i}$, construct two edge-disjoint paths P_{s_1,u_i} and P_{s_2,u_i}, which is always possible since e_i is a critical 1-edge cut and $d_{1,i}$ is reachable from both s_1 and s_2. Let $P_{v_i,d_{1,i}}$ denote *any* path connecting v_i and $d_{1,i}$. Let $P^{G\backslash\{\forall e_i\}}_{s_2,d_{2,j}}$ denote *any* path connecting s_2 and $d_{2,j}$ using only edges in $G\backslash\{e_i : \forall d_{1,i} \in \mathbf{d}_1\}$. Consider the following construction:

$$\begin{aligned}\forall d_{1,i} \in \mathbf{d}_1, \mathcal{P}_i &\triangleq \{P_{s_1,u_i} u_i v_i P_{v_i,d_{1,i}}, P_{s_2,u_i} u_i v_i P_{v_i,d_{1,i}}\}\\ \mathcal{P}_J &\triangleq \left\{P^{G\backslash\{\forall e_i\}}_{s_2,d_{2,j}} : \forall d_{2,j} \in \mathbf{d}_2\right\}. \end{aligned} \tag{1}$$

We will show that the above collections contain only valid paths (no cycles) and jointly they satisfy Condition 1 of Theorem 2.

For acyclic networks, the above construction contains no cycle and they are indeed valid paths. For cyclic networks, since $e_i = u_i v_i$ is a 1-edge cut, P_{s_1,u_i} and $P_{v_i,d_{1,i}}$ must not share any vertex. Therefore $P_{s_1,u_i} u_i v_i P_{v_i,d_{1,i}}$ is a valid path containing no cycle. Similarly, $P_{s_2,u_i} u_i v_i P_{v_i,d_{1,i}}$ is a valid path.

Suppose there is an edge e that violates Condition 1 of Theorem 2. Since by construction, P_{s_1,u_i} and P_{s_2,u_i} are edge-disjoint and share no common edge, such e must satisfy $e \in u_i v_i P_{v_i,d_{1,i}}$. By our construction of $\mathcal{P}_J$ in (1), any $P_{s_2,d_{2,j}} \in \mathcal{P}_J$ must not use $u_i v_i$. Therefore, the violating edge e must satisfy

$$e \in P_{v_i,d_{1,i}} \text{ and } e \in P_{s_2,d_{2,j}} \tag{2}$$

for some $d_{2,j}$. Nonetheless, since $e_i = u_i v_i$ is a 1-edge cut separating $\{s_1, s_2\}$ and $d_{1,i}$ and by (2) the walk $P_{s_2,d_2} e P_{v_i,d_{1,i}}$ is from s_2 to $d_{1,i}$, we must have $u_i v_i \in P_{s_2,d_{2,j}}$. This contradicts the construction of $\mathcal{P}_J$ in (1). As a result, no such violating edge e exists and the proof is complete. ■

B. The Necessary Condition for 2-Multicast Acyclic/Cyclic Networks

We first prove Theorem 4, the necessary condition for 2-multicast cyclic networks. Since acyclic networks are a special class of cyclic networks, the proof of the necessary condition of Theorems 1 and 2 for the 2-multicast acyclic networks follows similarly and its detailed derivation is omitted.

The equivalence of the conditions in Theorems 1 and 2 allows us to prove the necessary condition based on the cut-based characterization in Theorem 1. Assuming there exists an asymptotic rate-1 PINC solution, we show that for any $d_{2,j}$ there must exist a $P_{s_2,d_{2,j}}$ path that does not use any critical 1-edge cuts of any $d_{1,i}$. The symmetric case for the existence of $P_{s_1,d_{2,j}}$ in $G\backslash\{e_i : d_{1,i} \in \mathbf{d}_1\}$ can be proven by symmetry.

We need the following lemma before the proof.

Lemma 2: Suppose a rate-$(1-\epsilon)$ PINC solution exists in a 2-multicast cyclic network with duration T for some $\epsilon > 0$. For each edge e, define the edge weight $w_e \in \mathbb{R}$ by the conditional mutual information:

$$w_e \triangleq \frac{1}{T} I([M_e]_1^T; [Y]_1^T | [X]_1^T).$$

For any $d_{2,j} \in \mathbf{d}_2$, then there must exist a path $P_{s_2,d_{2,j}}$ such that

$$\forall e \in P_{s_2,d_{2,j}}, \quad w_e \geq \frac{1}{|E|}(1-\epsilon)\log(q). \tag{3}$$

Proof: If there exists no path $P_{s_2,d_{2,j}}$ with $\min_{e \in P_{s_2,d_{2,j}}} w_e \geq \frac{1}{|E|}(1-\epsilon)\log(q)$, then s_2 and $d_{2,j}$ must be disconnected after removing all edges with $w_e < \frac{1}{|E|}(1-\epsilon)\log(q)$. Let C denote the collection of those removed edges. C must be an edge-cut separating s_2 and $d_{2,j}$. Since the network coding solution is feasible, we have the following contradiction:

$$(1-\epsilon)\log(q) \leq \frac{1}{T} I([M_{d_{2,j}}]_1^T; [Y]_1^T) \tag{4}$$

$$\leq \frac{1}{T} I(\{[M_e]_1^T, e \in C\}; [Y]_1^T) \tag{5}$$

$$\leq \frac{1}{T} I(\{[M_e]_1^T, e \in C\}; [Y]_1^T | [X]_1^T) \tag{6}$$

$$\leq \sum_{e \in C} \frac{1}{T} I([M_e]_1^T; [Y]_1^T | [X]_1^T) \tag{7}$$

$$= \sum_{e \in C} w_e \leq |E| \max_{e \in C} w_e < (1-\epsilon)\log(q), \tag{8}$$

where (4) is by the definition of being a feasible network coding solution, (5) follows from the information conservation law that the mutual information between the cut messages $\{[M_e]_1^T, e \in C\}$ and $[Y]_1^T$ must be no less than that between $[M_{d_{2,j}}]_1^T$ and $[Y]_1^T$. (6) follows from that conditioning on the "independent interference" $[X]_1^T$ will only increase the mutual information. (7) follows from the basic entropy equalities/inequalities explained below:

$$\begin{aligned} T \cdot (6) &= H(\{[M_e]_1^T, e \in C\} | [X]_1^T) \\ &\qquad - H(\{[M_e]_1^T, e \in C\} | [Y]_1^T, [X]_1^T) \\ &= H(\{[M_e]_1^T, e \in C\} | [X]_1^T) - 0 \\ &= \sum_{i=1}^{|C|} H([M_{e_i}]_1^T | [X]_1^T, \{[M_{e_j}]_1^T, j < i\}) \\ &\leq \sum_{i=1}^{|C|} H([M_{e_i}]_1^T | [X]_1^T) \\ &= \sum_{e \in C} I([M_e]_1^T; [Y]_1^T | [X]_1^T). \end{aligned}$$

The contradiction in (8) completes the proof. ∎

To prove the necessary condition, we assume that one can achieve asymptotically rate-1 transmission by network coding. We then choose an ϵ such that $(1-\epsilon)(1+\frac{1}{|E|}) > 1$. By the definition of achievability, there exists a rate-$(1-\epsilon)$ PINC solution for the given ϵ with a sufficiently large duration T. By Lemma 2, there exists a $P_{s_2,d_{2,j}}$ satisfying (3) for any $d_{2,j}$. In the following, we show that any such $P_{s_2,d_{2,j}}$ does not use any critical 1-edge cuts of $d_{1,i}$.

We prove this by contradiction. Suppose a $P_{s_2,d_{2,j}}$ satisfying (3) uses a critical 1-edge cut e_i of $d_{1,i}$ for some i. Then

$$\begin{aligned} &I([M_{e_i}]_1^T; [X]_1^T, [Y]_1^T) \\ &= I([M_{e_i}]_1^T; [X]_1^T) + I([M_{e_i}]_1^T; [Y]_1^T | [X]_1^T). \end{aligned} \tag{9}$$

The left-hand side of (9) is no larger than $H([M_{e_i}]_1^T) \leq T\log(q)$. The first term of the right-hand side of (9) satisfies

$$I([M_{e_i}]_1^T; [X]_1^T) \geq I([M_{d_{1,i}}]_1^T; [X]_1^T) \geq T(1-\epsilon)\log(q),$$

since e_i is a 1-edge cut separating $\{s_1, s_2\}$ from $d_{1,i}$ and since the code has rate $(1-\epsilon)$. The second term of the right-hand side of (9) is $Tw_{e_i} \geq \frac{T}{|E|}(1-\epsilon)\log(q)$. By the above reasoning, we have the following contradiction

$$\begin{aligned} T\log(q) &\geq T(1-\epsilon)\log(q) + \frac{T}{|E|}(1-\epsilon)\log(q) \\ &= T(1-\epsilon)(1+\frac{1}{|E|})\log(q) > T\log(q). \end{aligned} \tag{10}$$

Therefore, $P_{s_2,d_{2,j}}$ does not use the critical 1-edge cut e_i for any $d_{1,i}$. The proof is complete.

C. The Sufficient Condition for 2-Multicast Acyclic Networks

In this subsection, we will show that if the conditions of Theorem 1 are satisfied, then there exists a linear intersession network coding solution for the 2-multicast acyclic network based on a new component of *sequential reset operations*.

Our network code construction consists of two stages. The first stage we construct a "strengthened *generic linear code multicast* (LCM)" based on the explicit construction of LCMs in [12], [19]. Then we perform "reset-to-X" (resp. "reset-to-Y") operations sequentially on the critical 1-edge cuts according to their topological order.

Stage 1: Strengthened Generic Linear Code Multicast (LCM)

For any linear network coding scheme, the message M along any edge is a linear combination $M = c_1 X + c_2 Y$ of the symbols X and Y, 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 (vector) along a specific edge e. We first give the intuition of our construction and a detailed step-by-step proof will be provided shortly after. Our construction is based on the *generic LCM* [19] that maintains the independence among all messages in the network to the largest possible degree. More rigorously, for a generic LCM on the 2-multicast acyclic network, by Theorem 3.3 in [19], we must have

1) If two messages on edges e_1 and e_2 are linearly dependent, then the min-cut/max-flow value between $\{s_1, s_2\}$ and $\{\mathsf{tail}(e_1), \mathsf{tail}(e_2)\}$ must be one.

By Theorem 5.1 in [19], every acyclic network admits a generic LCM, which can be explicitly constructed in polynomial time [12] for sufficiently large $\mathsf{GF}(q)$.

For a sufficiently large field space $\mathsf{GF}(q)$, we can further strengthen a generic LCM such that the following properties are also satisfied.

2) If two messages M_1 and M_2 are not identical, they are linearly independent.
3) Along any edge that is reachable from s_i, the i-th component of the corresponding message must be non-zero.

The second property is to remove the ambiguity of scaling by a constant. The third property is to ensure that the influence of s_1 and s_2 on the intermediate edges are preserved during the construction of a generic LCM.

For acyclic networks, a simple exponential-time algorithm for constructing a strengthened generic LCM is to assign coding vectors in an edge-by-edge fashion, starting from the most upstream to the most downstream edges while maintaining the largest degree of independence (ensuring Property 1) and preserving the influence of s_1 and s_2 on the intermediate edges (ensuring Property 3). This sequential construction was first used in the proof of Theorem 5.1 in [19] and will be the basis of our explicit construction.

The strengthened generic LCM will then be complemented with the "reset-to-X" and "reset-to-Y" operations described as follows. Take the reset-to-X operation, for example, and consider the sequential construction according to the topological order of the edges. For any edge $e = uv$ that has two independent incoming messages, the designer of the network code can choose whether to apply the reset-to-X operation or to use the regular construction of a strengthened generic LCM. If the reset-to-X operation is chosen, then instead of ensuring the satisfaction of Properties 1 to 3, one simply set $M_e = (1, 0)$. The message M_e is indeed "reset" to X. The coding vector of the remaining edges can be assigned sequentially with the objective functions of maximizing the independence between different edges and preserving the effects of s_1 and s_2 to the maximum extent, unless another reset operation is invoked.

The inclusion of the reset operation will affect the properties of the strengthened generic LCM. New properties after applying reset operations are listed as follows.

- Property 1: If two messages M_1 and M_2 on edges e_1 and e_2 are identical, it is either that the min-cut/max-flow value between $\{s_1, s_2\}$ and $\{\mathsf{tail}(e_1), \mathsf{tail}(e_2)\}$ is one, or $M_1 = M_2$ is of value either $(1, 0)$ or $(0, 1)$.
- Property 2: If two messages M_1 and M_2 are not identical, they are linearly independent.
- Property 3: Along any edge that is not reset to Y and is reachable from s_1 without using edges that are "reset to Y", the X-component of the corresponding message must be non-zero. Symmetrically, along any edge that is not reset to X and is reachable from s_2 without using any reset-to-X edges, the Y-component of the corresponding message must be non-zero.

Based on the above discussion, we describe, in the following, a sequential method of constructing a strengthened generic LCM with reset operations such that the resulting network code satisfies the above three properties. To that end, we first index the edges from the most upstream ($e = 1$) to the most downstream edges ($e = |E|$). The coding vectors of the edges will be assigned from $e = 1$ to $e = |E|$ and we will prove that the above three properties hold during construction by induction on the edge e being processed.

When $e = 1$, the edge e must be a source edge, say s_1. There is no incoming edge to the source edge. The only possible coding vector assignment is $(1, 0)$ (ignoring the scaling coefficient). The three properties hold for $e = 1$. For an intermediate edge $e = k$, we assume that the three properties hold for edges 1 to $k - 1$. Consider two cases depending on whether a reset operation is performed on $e = k$ or not.

Case 1: the designer is allowed to perform the reset-to-X operation on e and indeed chooses to reset. For Property 1, if there exists an $e' < e = k$ such that $M_e = M_{e'}$, then $M_e = M_{e'} = (1, 0)$. Property 1 thus holds for the first k edges. To prove Property 2, suppose M_e is linearly dependent to another $M_{e'}$ with $e' < k$ and $M_e \neq M_{e'}$. Then $M_{e'}$ must be of the form $(x, 0)$ for some $x \neq 1$. Nonetheless, since $M_{s_1} = (1, 0) \neq M_{e'}$ and Property 2 holds for $e = 1, \cdots, k - 1$, we must have $M_{s_1} = (1, 0)$ being independent of $M_{e'} = (x, 0)$ for some $x \neq 1$. This contradiction proves that Property 2 holds for the first k edges. For Property 3, since $M_e = (1, 0)$ contains non-zero X component and e is reset-to-X, Property 3 holds for $e = k$ by definition. Since all three properties hold after processing $e = k$, by induction, the proof is complete. The proof for the symmetric scenario in which the designer chooses to perform the reset-to-Y operation follows by symmetry.

Case 2: No reset operation is performed on e. We have two subcases. Case 2.1: The designer is not allowed to perform a reset operation, i.e., all incoming edges entering e carry identical coding vectors. In this sub-case, we simply set the coding vector of e by $M_e = M_{e_{\text{in},l}}$ for any incoming edge $e_{\text{in},l}$. For Property 1, if there exists another e' such that $M_{e'} = M_e = M_{e_{\text{in},l}}$, by induction, one possibility is that $M_{e'} = M_{e_{\text{in},l}} = M_e$ are $(1, 0)$ or $(0, 1)$. Property 1 is thus satisfied for $e = k$. The other possibility is that the min-cut/max-flow value from $\{s_1, s_2\}$ to $\{\mathsf{tail}(e'), \mathsf{tail}(e_{\text{in},l})\}$ is one for all incoming edges $e_{\text{in},l}$ entering e and the min-cut/max-flow value from $\{s_1, s_2\}$ to $\{\mathsf{tail}(e_{\text{in},l_1}), \mathsf{tail}(e_{\text{in},l_2})\}$ for all e_{in,l_1} and e_{in,l_2} entering e. It can be shown that in this case the min-cut/max-flow value from $\{s_1, s_2\}$ to $\{\mathsf{tail}(e')\} \cup \{\mathsf{tail}(e_{\text{in},l}) : \forall l\}$ must also be one since they share the same unique critical 1-edge cut as predicted in Lemma 1. Therefore, the min-cut/max-flow value from $\{s_1, s_2\}$ to $\{\mathsf{tail}(e'), \mathsf{tail}(e)\}$ is also 1 as $e_{\text{in},l}$ are the edges entering $e = k$. Property 1 thus holds for Case 2.1. For Property 2, for any $M_{e'} \neq M_e = M_{e_{\text{in},l}}$, by induction, we must have $M_{e'}$ being independent to $M_{e_{\text{in},l}} = M_e$. Property 2 thus holds for Case 2.1. For Property 3, if e is reachable from s_1 without using any reset-to-Y edges, then one of the $e_{\text{in},l}$ is not reset-to-Y and is reachable from s_1 without using any reset-to-Y edges. Therefore, the X component of $M_{e_{\text{in},l}}$ is non-zero, so is the X component of M_e. All three properties hold for Case 2.1.

Case 2.2: At least two incoming edges carry distinct coding vectors. By induction, e must receive two independent coding vectors. In this case, the designer is allowed to perform a reset operation but chooses not to. Since $\mathsf{tail}(e)$ can decode both X and Y, the designer has the freedom to send any (a, b) vector along e. To satisfy the three properties, one can simply choose $a \neq 0$ and $b \neq 0$ such that the coding vectors are independent to any upstream edges of $e = k$. With a sufficiently large $\mathsf{GF}(q)$, such a choice always exists. All three properties thus hold for Case 2.2.

By induction, a strengthened generic LCM can thus be constructed with arbitrary reset operations. For the following, we will refer to these three properties as Properties 1 to 3 of a generic LCM with *reset operations*.

Stage 2: Sequential Reset Operations

In the previous stage, we have discussed the properties of a generic LCM with reset operations. The remaining task is to decide which edge to follow normal coding operation and which edge to "reset". It turns out that we simply need to reset-to-X all critical 1-edge cuts $e_i = u_i v_i$ for destinations $d_{1,i} \in \mathbf{d}_1$ and reset-to-Y all critical 1-edge cuts $e_j = u_j v_j$ for destination $d_{2,j} \in \mathbf{d}_2$.

We first observe that the above reset operations are unambiguous since the 1-edge cuts e_i for $d_{1,i}$ are disjoint from the critical 1-edge cuts e_j for $d_{2,j}$. Nonetheless, it is possible that we cannot perform the reset operations on some of the target edges since resetting an edge e requires that the incoming edges of e carry independent vectors. Our goal is to prove the following two statements: (i) it is possible to perform reset-to-X (resp. reset-to-Y) operations on all $e_i = u_i v_i$ (resp. $e_j = u_j v_j$), and (ii) after performing the reset operations, all $d_{1,i}$ and $d_{2,j}$ are able to receive their desired symbols successfully.

We prove statement (ii) first. Take a $u_{i_0} v_{i_0}$ edge for example. Suppose the reset-to-X operation can be performed, namely, suppose the incoming messages of $u_{i_0} v_{i_0}$ are either independent or are of the form $(1, 0)$. (If the incoming messages are dependent and of the form (a, b), $b \neq 0$, then a reset-to-X operation is not possible.) Once the reset-to-X operation is performed, d_{1,i_0} is able to receive X successfully since as $u_{i_0} v_{i_0}$ being a critical 1-edge cut, there is no other "interfering path" that can corrupt the transmission from $u_{i_0} v_{i_0}$ to d_{1,i_0}. The newly reset symbol X can thus arrive d_{1,i_0} without contamination. Similarly, for any d_{2,j_0}, if the reset-to-Y operation is performed on $u_{j_0} v_{j_0}$, then d_{2,j_0} will receive Y successfully. Statement (ii) is thus proven.

It remains to show that on each $u_i v_i$ or $u_j v_j$, performing the reset operation is indeed always feasible. Take an $e_{i_0} = u_{i_0} v_{i_0}$ edge for example. If there are two incoming edges entering e_{i_0} that carry distinct messages, then u_0 can decode both X and Y and reset-to-X is always possible. Suppose that all incoming edges entering e_{i_0} carry identical coding vectors. Consider the following cases. The conditions in Theorem 1 guarantee that there is a $P_{s_1, d_{1,i_0}}$ path that does not use any critical 1-edge cut $u_j v_j$ of destination $d_{2,j}$. Since $u_{i_0} v_{i_0}$ is a 1-edge cut separating $\{s_1, s_2\}$ and d_{1,i_0}, such $P_{s_1, d_{1,i_0}}$ path must use $u_{i_0} v_{i_0}$. Since we only perform reset-to-Y operations on the critical 1-edge cut $u_j v_j$ of destination $d_{2,j}$, the $e_{i_0} = u_{i_0} v_{i_0}$ edge is reachable from s_1 (through the $P_{s_1, d_{1,i_0}}$ path) without using any reset-to-Y edges. By Property 3 of the strengthened generic LCM with reset operations, at least one incoming edges of $e_{i_0} = u_{i_0} v_{i_0}$ must have a non-zero X component. So the incoming edges must carry identical coding vectors of the form (a, b) for some $a \neq 0$. Since e_{i_0} is a critical 1-edge cut, there are two edge-disjoint paths $P_{s_1, u_{i_0}}$ and $P_{s_2, u_{i_0}}$. Therefore, there are at least two incoming edges e_{in, l_1} and e_{in, l_2} entering u_{i_0} such that the min-cut/max-flow value between $\{s_1, s_2\}$ and $\{\mathsf{tail}(e_{\text{in}, l_1}), \mathsf{tail}(e_{\text{in}, l_2})\}$ is two. Since $M_{e_{\text{in}, l_1}} = M_{e_{\text{in}, l_2}} = (a, b)$, by Property 1 of the strengthened generic LCM with reset operations, we must have $(a, b) = (1, 0)$. Therefore the reset-to-X is always possible on $e_{i_0} = u_{i_0} v_{i_0}$. The proof is complete.

D. The Sufficient Condition for 2-Unicast Cyclic Networks

The looping behavior of cyclic networks and the associated delay constraint increase the complexity of analysis. For example, in the 2-multicast acyclic network, one needs only to apply reset operations to the critical 1-edge cut for each destination. Messages on other edges are based on the generic LCM and can be approximated by random linear network coding. For cyclic networks, more edges have to cooperate jointly with each other in a non-random fashion and one generally needs both intra- and intersession network coding when two strings of symbols $X_1, \cdots, X_t$ and $Y_1, \cdots, Y_t$ are considered.

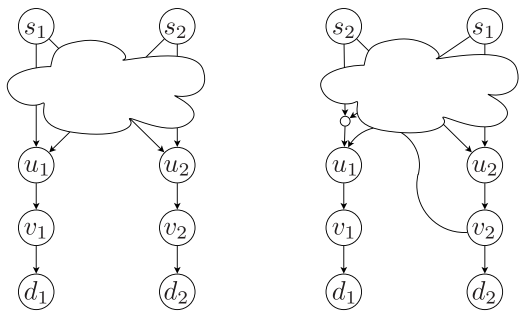
For the following, we assume that the path-based conditions in Corollary 1 are satisfied for a 2-unicast cyclic network, for which a linear PINC solution will be constructed. If there exists a pair of edge-disjoint paths P_{s_1, d_1} and P_{s_2, d_2}, one can achieve rate-1 communication by a non-coded solution. We focus on the more interesting scenario and assume that there exists no such edge-disjoint path pair. For a node $v \in V$, we use $[v]_1^t$ to denote the collection $\{[M_e]_1^t : \mathsf{head}(e) = v\}$ that is the cumulative information available to v after the first t seconds.

Let $e_1 = u_1 v_1$ denote the critical 1-edge cut separating $\{s_1, s_2\}$ and d_1. Similarly, $e_2 = u_2 v_2$ is the critical 1-edge cut separating $\{s_1, s_2\}$ and d_2. We have $e_1 \neq e_2$ otherwise $e_1 = e_2$ is a 1-edge cut separating $\{s_1, s_2\}$ and $\{d_1, d_2\}$, and Corollary 1, as a necessary condition, cannot be satisfied. By the construction of e_i as critical 1-edge cuts, there must exist two vertex-disjoint paths $P_{v_1, d_1}^{G \backslash \{e_1, e_2\}}$ and $P_{v_2, d_2}^{G \backslash \{e_1, e_2\}}$ in the subgraph $G \backslash \{e_1, e_2\}$. Otherwise, either (i) one of $\{e_1, e_2\}$ is not a critical 1-edge cut, or (ii) e_1 and e_2 coincides, which contradicts our construction.

Consider the following three major cases, and for each case we describe within the parentheses the main intuition behind the network code construction.

Case 1: (Direct construction based on the concept of generations.) There exist two edge-disjoint paths $(P_{s_1, u_1}^{G \backslash \{e_1, e_2\}}, P_{s_2, u_1}^{G \backslash \{e_1, e_2\}})$ using only edges in $G \backslash \{e_1, e_2\}$, and there exist two edge-disjoint paths $(P_{s_1, u_2}^{G \backslash \{e_1, e_2\}}, P_{s_2, u_2}^{G \backslash \{e_1, e_2\}})$ using only edges in $G \backslash \{e_1, e_2\}$. We first remove any edges that are not one of the critical 1-edge cuts $e_1 = u_1 v_1$ and $e_2 = u_2 v_2$ and are not in the following six paths:

$$P_{v_1, d_1}^{G \backslash \{e_1, e_2\}}, P_{v_2, d_2}^{G \backslash \{e_1, e_2\}}, P_{s_1, u_1}^{G \backslash \{e_1, e_2\}}, P_{s_2, u_1}^{G \backslash \{e_1, e_2\}},$$
$$P_{s_1, u_2}^{G \backslash \{e_1, e_2\}}, \text{ and } P_{s_2, u_2}^{G \backslash \{e_1, e_2\}}. \tag{11}$$



(a) Case 1 (b) Case 2

Fig. 8. Cases 1 and 2 of Section IV-D.

We will show that one can construct a rate-1 network coding solution on the remaining graph (see Fig. 8(a)).

Since both the max-flow values from $\{s_1, s_2\}$ to u_1 and from $\{s_1, s_2\}$ to u_2 are two, respectively, [19] shows that for any $\epsilon > 0$, there exists a linear network code (the generic LCM [19]) with sufficiently large duration T such that after T second,

$$\frac{1}{T} I([u_1]_1^T; [X]_1^T, [Y]_1^T) > 2(1-\epsilon)\log(q)$$
$$\frac{1}{T} I([u_2]_1^T; [X]_1^T, [Y]_1^T) > 2(1-\epsilon)\log(q),$$

i.e. u_1 and u_2 can recover both $[X]_1^T$ and $[Y]_1^T$.[5] We will use this particular generic LCM without modification for $t = 1, \cdots T$. It is crucial to notice that the existence of a generic LCM does not guarantee that node u_1 can decode $[X]_1^t$ based on $[u_1]_1^t$ *on the fly* for $t < T$, but only guarantees that after T seconds, u_1 can decode all T symbols $[X]_1^T$ at once. Therefore, in the first T seconds, destination d_1 (resp. d_2) cannot receive uncorrupted messages as u_1 (resp. u_2) does not decode $[X]_1^t$ on the fly. We define the first T seconds as the first period and denote the corresponding symbols as $[X^{(1)}]_1^T$ and $[Y^{(1)}]_1^T$.

For the second period, the sources send $[X^{(2)}]_1^T$ and $[Y^{(2)}]_1^T$ for a different "generation" of packets instead of the old generation symbols $[X^{(1)}]_1^T$ and $[Y^{(1)}]_1^T$. Again, we use a generic LCM to send it to u_1 and u_2 with the following modifications. In the t-th second of the second duration, u_1 sends along e_1 the first generation symbol $X_t^{(1)}$ instead of a linear combination of the cumulative information of the second generation, denoted by $[u_1^{(2)}]_1^{t-1}$. Similarly, u_2 sends the first generation symbol $Y_t^{(1)}$ along e_2. As a result, in the end of the second period, d_1 and d_2 are able to recover, respectively, $[X^{(1)}]_1^T$ and $[Y^{(1)}]_1^T$ sent in the first period. In the third period, the third-generation symbols $[X^{(3)}]_1^T$ and $[Y^{(3)}]_1^T$ are sent by s_1 and s_2 using a generic LCM and the second-generation symbols $X_t^{(2)}$ (resp. $Y_t^{(2)}$) are sent along e_1 (resp. e_2). By repeating the above scheme for a sufficiently large number of periods K, we can send $(K-1)T(1-\epsilon)$ symbols in KT seconds. The data rate can thus be made arbitrarily close to $(1-\epsilon)$ for sufficiently large K. The construction is complete.

Case 2: (Direct construction based on the concepts of generations and super-generations.) There exist two edge-disjoint paths $(P_{s_1,u_2}^{G\backslash\{e_1,e_2\}}, P_{s_2,u_2}^{G\backslash\{e_1,e_2\}})$ using only edges in $G\backslash\{e_1, e_2\}$. But there exist no edge-disjoint paths $(P_{s_1,u_1}^{G\backslash\{e_1,e_2\}}, P_{s_2,u_1}^{G\backslash\{e_1,e_2\}})$ using only edges in $G\backslash\{e_1, e_2\}$. Since e_1 is a critical 1-edge cut, there still exist two edge-disjoint paths $P_{s_1,u_1}^{G\backslash e_1}$ and $P_{s_2,u_1}^{G\backslash e_1}$ but one of them must use e_2. Moreover, it must be $P_{s_1,u_1}^{G\backslash e_1}$ that uses e_2. Otherwise, there exists a pair of EDPs $P_{s_1,d_1} = P_{s_1,u_1}^{G\backslash e_1} e_1 P_{v_1,d_1}^{G\backslash\{e_1,e_2\}}$ and $P_{s_2,d_2} = P_{s_2,u_1}^{G\backslash e_1} e_2 P_{v_2,d_2}^{G\backslash\{e_1,e_2\}}$, which contradicts the assumption. Remove any edges that are not one of the critical 1-edge cuts $e_1 = u_1v_1$ and $e_2 = u_2v_2$ and are not in the following six paths:

$$P_{v_1,d_1}^{G\backslash\{e_1,e_2\}}, P_{v_2,d_2}^{G\backslash\{e_1,e_2\}}, P_{s_1,u_1}^{G\backslash e_1}, P_{s_2,u_1}^{G\backslash e_1},$$
$$P_{s_1,u_2}^{G\backslash\{e_1,e_2\}}, \text{ and } P_{s_2,u_2}^{G\backslash\{e_1,e_2\}}.$$

We will show that one can construct a rate-1 network coding solution on the remaining graph (see Fig. 8(b) for illustration).

For this case, we need the notion of "super-period" or correspondingly "super-generation." More explicitly, we have totally K_1 super-periods and each super-period contains K_2 periods. So the total duration of the scheme is K_1K_2T seconds. The "(k_1, k_2) period" refers to the k_2-th period of the k_1-th super-period. Let $[X^{(k_1,k_2)}]_1^T$ and $[Y^{(k_1,k_2)}]_1^T$ denote the packets sent by s_1 and s_2 in the (k_1, k_2) period.

In the $(1, 1)$ period, a generic LCM is used and after the $(1, 1)$ period, u_2 is able to recover both $[X^{(1,1)}]_1^T$ and $[Y^{(1,1)}]_1^T$. For the $(1, k_2)$ period, $k_2 = 2, \cdots, K_2 - 1$, u_2 sends $[Y^{(1,k_2-1)}]_1^T$ along edge e_2 while the rest of the network uses the original generic LCM. This is possible since there are two edge-disjoint paths connecting $\{s_1, s_2\}$ and u_2 in $G\backslash e_1$ and all the contamination of the cyclic traffic (along the v_2 to u_1 path in Fig. 8(b)) can be cancelled by u_2's knowledge about the X and Y in the previous periods. So u_2 is able to recover $[X^{(1,k_2)}]_1^T$ and $[Y^{(1,k_2)}]_1^T$ in the end of the $(1, k_2)$ period. u_2 then sends out the $[Y^{(1,k_2)}]_1^T$ in the $(1, k_2 + 1)$ period. After the $(1, K_2 - 1)$ period, d_2 is able to recover $[Y^{(1,k_2-1)}]_1^T$ for $2 \leq k_2 \leq K_2 - 1$. On the other hand, u_1 is facing a bigger challenge. Take the $(1, 2)$ period as an example. The messages received by u_1 are linear combinations of the current generation $[X^{(1,2)}]_1^T$, $[Y^{(1,2)}]_1^T$ and of the previous generation $[Y^{(1,1)}]_1^T$, where the latter is due to the modified messages sent along e_2. Even when u_1 cancels out the impact of $[Y^{(1,1)}]_1^T$ by its previous knowledge obtained in the $(1, 1)$ period, u_1 *still cannot recover both* $[X^{(1,2)}]_1^T$ *and* $[Y^{(1,2)}]_1^T$ *since the min-cut/max-flow value between* $\{s_1, s_2\}$ *and* u_1 *is one in* $G\backslash\{e_1, e_2\}$. u_1 thus cannot send any uncorrupted X packets to d_1.[6]

To solve this problem, we perform the "unwrap operation" in the $(1, K_2)$ period. In the $(1, K_2)$ period, the sources resend $[X^{(1,K_2-1)}]_1^T$ and $[Y^{(1,K_2-1)}]_1^T$ of the previous $(1, K_2 - 1)$ generation using a generic LCM *without any modification.* As

[5]More rigorously, to fully recover $[X]_1^T$ and $[Y]_1^T$, we have to send (linearly) coded packets instead of i.i.d. packets since we cannot have the mutual information $I([u_1]_1^T; [X]_1^T, [Y]_1^T) = 2T\log(q)$ with equality due to the delay of the network.

[6]In the $(1, 2)$ period, u_1 can still send $[X^{(1,1)}]_1^T$ along e_1 as u_1 has obtained $[X^{(1,1)}]_1^T$ in the $(1, 1)$ period. However, u_1 has no uncorrupted X symbols to send along e_1 during the $(1, 3)$ period as all symbols received by u_1 in the $(1, 2)$ period are corrupted and cannot be recovered.

a result, d_2 is not receiving any new uncorrupted Y. The total transmission rate for the (s_2, d_2) is then

$$\frac{(K_2-2)(1-\epsilon)T\log(q)}{K_2T},$$

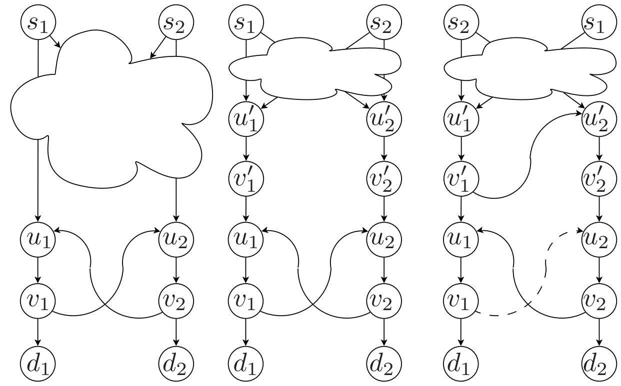
since only $\{[Y^{(1,k_2)}]_1^T : k_2 = 1, \cdots K_2-2\}$ are received by d_2.

On the other hand, since a generic LCM is used, u_1 receives $[X^{(1,K_2-1)}]_1^T$ and $[Y^{(1,K_2-1)}]_1^T$ successfully in the $(1, K_2)$ period. Since u_1 also has the $2T$ linear combinations of $[X^{(1,K_2-1)}]_1^T$, $[Y^{(1,K_2-1)}]_1^T$, and $[Y^{(1,K_2-2)}]_1^T$ received in the $(1, K_2-1)$ period, u_1 can use the new information $[X^{(1,K_2-1)}]_1^T$ and $[Y^{(1,K_2-1)}]_1^T$ to cancel out the effect of $[X^{(1,K_2-1)}]_1^T$ and $[Y^{(1,K_2-1)}]_1^T$ in the $2T$ linear equations and thus decode the remaining T unknown variables $[Y^{(1,K_2-2)}]_1^T$. This new piece of information can subsequently help decode another $2T$ symbols $[X^{(1,K_2-2)}]_1^T$ and $[Y^{(1,K_2-3)}]_1^T$, by cancelling the effect of $[Y^{(1,K_2-2)}]_1^T$ in the $2T$ linear combinations of $[X^{(1,K_2-2)}]_1^T$, $[Y^{(1,K_2-2)}]_1^T$, and $[Y^{(1,K_2-3)}]_1^T$ received in the $(1, K_2-2)$ period. Since we have $2T$ remaining unknown variables, $2T$ equations, and since the min-cut/max-flow value between $\{s_1, v_2\}$ and u_1 is two in $G\backslash\{e_1, e_2\}$, decoding is feasible when a sufficiently large finite field is used. By iteratively decoding information symbols based on the linear combinations received in the $(1, K_2-2)$, $(1, K_2-3)$, to $(1, 2)$ periods, u_1 can recover the information of $\{[X^{(1,k_2)}]_1^T, [Y^{(1,k_2)}]_1^T : \forall k_2 = 1, \cdots, K_2-1\}$ after the $(1, K_2)$ period. The convoluted linear combinations are thus *unwrapped* in the end of the first super-period. Note that d_1 has not received any uncorrupted X symbols during the first super-period.

For the $(2, 1)$ period, sources send $[X^{(2,1)}]_1^T$ and $[Y^{(2,1)}]_1^T$. A generic LCM is used and after the $(2, 1)$ period, u_2 is able to recover both $[X^{(2,1)}]_1^T$ and $[X^{(2,1)}]_1^T$. For the $(2, k_2)$ period, $k_2 = 2, \cdots, K_2-1$, u_2 sends $[Y^{(2,k_2-1)}]_1^T$ along edge e_2, while the u_1 sends the previous super-generation $[X^{(1,k_2-1)}]_1^T$ along edge e_1. As a result, after the $(2, k_2)$ period, d_1 and d_2 recover $[X^{(1,k_2-1)}]_1^T$ and $[Y^{(2,k_2-1)}]_1^T$, respectively. We note that it can be shown that in the construction of the subgraph (11), u_2 is not reachable from v_1. Therefore, from u_2's perspective there is no need to worry about any interference caused by u_1 sending $[X^{(1,k_2-1)}]_1^T$ along edge e_1. Similar to that in the first super-period, u_1 on the other hand cannot recover any symbols in the current super-period when $k_2 < K_2$. In the $(2, K_2)$ period, we again perform the "unwrap operation." The sources resend $[X^{(2,K_2-1)}]_1^T$ and $[Y^{(2,K_2-1)}]_1^T$ of the previous generation using a generic LCM without any modification. All information in this super-generation can be unwrapped and recovered at u_1. Both u_1 and u_2 are now ready for the next super-period.

The average transmission rates r_1 and r_2 for (s_1, d_1) and (s_2, d_2) respectively then become

$$r_1 = \frac{(K_1-1)(K_2-2)(1-\epsilon)T\log(q)}{K_1K_2T}$$
$$r_2 = \frac{K_1(K_2-2)(1-\epsilon)T\log(q)}{K_1K_2T},$$



(a) Case 3 (b) Case 3.1 (c) Case 3.2

Fig. 9. Cases 3 to 3.2 of Section IV-D. Only the main structure of the network is drawn. Our proof still considers the possible cyclic structure, which on the other hand is not included in these three figures for readability.

where d_1 receives only K_1-1 super generations while d_2 receives full K_1 super generations. By choosing a small $\epsilon > 0$ and by choosing sufficiently large K_1 and K_2, we have a rate-1 PINC solution.

Case 3: There exist no edge-disjoint paths $(P_{s_1,u_2}^{G\backslash\{e_1,e_2\}}, P_{s_2,u_2}^{G\backslash\{e_1,e_2\}})$ using only edges in $G\backslash\{e_1, e_2\}$ and there exist no edge-disjoint paths $(P_{s_1,u_1}^{G\backslash\{e_1,e_2\}}, P_{s_2,u_1}^{G\backslash\{e_1,e_2\}})$ using only edges in $G\backslash\{e_1, e_2\}$. See Fig. 9(a) for illustration. Depending on the network topology between $\{s_1, s_2\}$ and $\{u_1, u_2\}$, we have three sub-cases.

We first note that there must exist a pair of edge-disjoint path connecting $\{s_1, s_2\}$ and $\{d_1, d_2\}$, otherwise simultaneous communication is simply impossible. In our assumption, we have ruled out the case that there exists a pair of edge-disjoint paths P_{s_1,d_1} and P_{s_2,d_2}. Therefore, there must exist a pair of edge-disjoint paths P_{s_1,d_2} and P_{s_2,d_1}, which in turn implies that u_1 is reachable from s_2 in $G\backslash\{e_1, e_2\}$ and u_2 is reachable from s_1 in $G\backslash\{e_1, e_2\}$. Also note that u_1 must be reachable from s_1 otherwise the cut-based condition in Theorem 1 cannot be satisfied. From the above reasoning, u_1 is reachable from both s_1 and s_2 in $G\backslash\{e_1, e_2\}$ and so is u_2. Let $e_1' = u_1'v_1'$ denote the critical 1-edge cut in $G\backslash\{e_1, e_2\}$ that separates $\{s_1, s_2\}$ and u_1. Note that by the condition of Case 3, the min-cut/max-flow value between $\{s_1, s_2\}$ and u_1 is one for graph $G\backslash\{e_1, e_2\}$. Therefore, the new critical 1-edge cut e_1' always exists by Lemma 1. Symmetrically, we define the critical 1-edge cut $e_2' = u_2'v_2'$ separating $\{s_1, s_2\}$ and u_2 in $G\backslash\{e_1, e_2\}$. One can show that e_1' and e_2' must be distinct. Otherwise, e_1' and e_2' will be the unique critical 1-edge cut separating $\{s_1, s_2\}$ and $\{d_1, d_2\}$, which violates the assumption that the min-cut/max-flow value between $\{s_1, s_2\}$ and $\{d_1, d_2\}$ is two. By the construction of e_1' and e_2' as the critical 1-edge cuts, one can also prove that there exist two edge-disjoint paths $P_{v_1',d_1}^{G\backslash\{e_1',e_2'\}}$ and $P_{v_2',d_2}^{G\backslash\{e_1',e_2'\}}$ in $G\backslash\{e_1', e_2'\}$. The reason is that any EDP pair P_{s_1,d_2} and P_{s_2,d_1} must satisfy $e_2 \in P_{s_1,d_2}$ and $e_1 \in P_{s_2,d_1}$, which in turn implies $e_2' \in P_{s_1,d_2}$ and $e_1' \in P_{s_2,d_1}$. We can then construct $P_{v_1',d_1}^{G\backslash\{e_1',e_2'\}} = e_1'P_{s_2,d_1}$ and $P_{v_2',d_2}^{G\backslash\{e_1',e_2'\}} = e_2'P_{s_1,d_2}$. The three sub-cases can now be described based on the relationship between the new edges e_1'

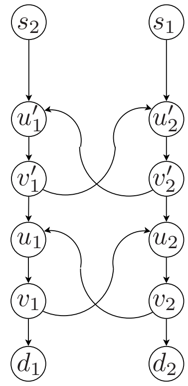


Fig. 10. Case 3.3 of Section IV-D.

and e_2'.

Case 3.1: (The same construction as in Case 1.) There exist two edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_1'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_1'}$ in $G\backslash\{e_1',e_2'\}$, and there exist two edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_2'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_2'}$ in $G\backslash\{e_1',e_2'\}$ (see Fig. 9(b)). By noticing that $e_1' = u_1'v_1'$ and $e_2' = u_2'v_2'$ in Case 3.1 (Fig. 9(b)) have the same role as $e_1 = u_1v_1$ and $e_2 = u_2v_2$ in Case 1 (Fig. 8(a)), we can follow the same period-based construction as in Case 1. A rate-1 PINC solution is feasible.

Case 3.2: (The same construction as in Case 2.) There exist two edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_1'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_1'}$ in $G\backslash\{e_1',e_2'\}$, and there exist no edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_2'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_2'}$ in $G\backslash\{e_1',e_2'\}$ (see Fig. 9(c)). Recall that by the construction of e_1 there exists a pair of edge-disjoint paths $P^{G\backslash e_1}_{s_1,u_1}$ and $P^{G\backslash e_1}_{s_2,u_1}$. One of them must use e_2 since we are in Case 3. One can show that it must be $P^{G\backslash e_1}_{s_1,u_1}$ that uses e_2 otherwise there will exist an EDP pair P_{s_1,d_1} and P_{s_2,d_2}, which contradicts the assumption. By the construction of e_1' and e_2' as critical 1-edge cuts, this implies that $P^{G\backslash e_1}_{s_1,u_1}$ must use e_2' and $P^{G\backslash e_1}_{s_2,u_1}$ must use e_1' (see Fig. 9(c) for illustration). Remove any edges that are not one of the critical 1-edge-cuts e_1, e_2, e_1', and e_2', and are not in the following six path segments:

$$P^{G\backslash\{e_1,e_2\}}_{v_1,d_1}, P^{G\backslash\{e_1,e_2\}}_{v_2,d_2}, P^{G\backslash e_1}_{s_1,u_1}, P^{G\backslash e_1}_{s_2,u_1}, P^{G\backslash\{e_1,e_2'\}}_{s_1,u_2'}, \text{ and } P^{G\backslash\{e_1,e_2'\}}_{s_2,u_2'}. \quad (12)$$

Intuitively, the above subgraph construction is equivalent to ignoring the path segment from v_1 to u_2 (the dashed arrow in Fig. 9(c)). We then notice that $e_1 = u_1v_1$ and $e_2' = u_2'v_2'$ in Case 3.2 (Fig. 9(c)) now have the same role as $e_1 = u_1v_1$ and $e_2 = u_2v_2$ in Case 2 (Fig. 8(b)), i.e., e_1 (resp. e_2') is now the critical 1-edge cuts separating $\{s_1,s_2\}$ and d_1 (resp. d_2) and e_1 and e_2' satisfy the conditions in Case 2. We can thus follow the same super-period-based construction as in Case 2. A rate-1 PINC solution is feasible. The case that there exist two edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_2'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_2'}$ in $G\backslash\{e_1',e_2'\}$, and there exist no edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_1'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_1'}$ in $G\backslash\{e_1',e_2'\}$ can be obtained by symmetry.

Case 3.3: (A unique case for cyclic networks that is not present in any acyclic network. Direct construction is provided for this case.) There exist no edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_1'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_1'}$ in $G\backslash\{e_1',e_2'\}$. And there exist no edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_2'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_2'}$ in $G\backslash\{e_1',e_2'\}$ (see Fig. 10). In this case, there must exist two edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{v_1',d_1}$ and $P^{G\backslash\{e_1',e_2'\}}_{v_2',d_2}$ in $G\backslash\{e_1',e_2'\}$. By the assumption that there are no edge-disjoint paths P_{s_1,d_1} and P_{s_2,d_2}, there must exist two edge-disjoint paths $P^{G\backslash\{e_1',e_2'\}}_{s_2,u_1'}$ and $P^{G\backslash\{e_1',e_2'\}}_{s_1,u_2'}$ as illustrated in Fig. 10. We now use P_{s_1,d_2} and P_{s_2,d_1} to denote the concatenation of the two EDP pairs: $(P^{G\backslash\{e_1',e_2'\}}_{s_2,u_1'}, P^{G\backslash\{e_1',e_2'\}}_{s_1,u_2'})$ and $(P^{G\backslash\{e_1',e_2'\}}_{v_1',d_1}, P^{G\backslash\{e_1',e_2'\}}_{v_2',d_2})$. Note that with this construction P_{s_1,d_2} must use e_2' and e_2 and P_{s_2,d_1} must use e_1' and e_1. Moreover, P_{s_1,d_2} and P_{s_2,d_1} must not share any vertex otherwise there will exist a pair of EDPs connecting (s_1,d_1) and (s_2,d_2). By the same reasons as in Case 3.2, we can define $P_{v_1,u_2} \triangleq v_1P^{G\backslash e_2}_{s_2,u_2}$, $P_{v_2,u_1} \triangleq v_2P^{G\backslash e_1}_{s_1,u_1}$, $P_{v_1',u_2'} \triangleq v_1'P^{G\backslash\{e_1,e_2'\}}_{s_2,u_2'}$, and $P_{v_2',u_1'} \triangleq v_2'P^{G\backslash\{e_1',e_2\}}_{s_1,u_1'}$.

For the following, we consider only the subgraph induced by the following six path segments

$$P_{s_1,d_2}, P_{s_2,d_1}, P_{v_1,u_2}, P_{v_2,u_1}, P_{v_1',u_2'}, \text{ and } P_{v_2',u_1'}, \quad (13)$$

and we will establish that from a coding perspective, the six path segments are equivalent to the vertical concatenation of "two figure-eight knots" as illustrated in Fig. 10.

By the construction of P_{v_1,u_2} as part of $P^{G\backslash e_2}_{s_2,u_2}$, we have $e_2' \notin P_{v_1,u_2}$ since $e_2' \notin P^{G\backslash e_2}_{s_2,u_2}$ as discussed in Case 3.2. Since by construction $e_1, e_2 \notin P_{v_1,u_2}$, we have that P_{v_1,u_2} does not touch any vertex in $P_{s_1,d_2}u_2'$ otherwise e_2' is not a critical 1-edge cut in $G\backslash\{e_1,e_2\}$ separating $\{s_1,s_2\}$ and u_2. Since e_2 is a critical 1-edge cut in G separating $\{s_1,s_2\}$ and d_2, we also have that P_{v_1,u_2} does not touch any vertex in $v_2P_{s_1,d_2}$. Therefore, P_{v_1,u_2} must only touch the path segment $v_2'P_{s_1,d_2}u_2$. Without loss of generality, we can assume that along P_{v_1,u_2}, the first vertex that also belongs to $v_2'P_{s_1,d_2}u_2$ is u_2 otherwise we can relabel the first such vertex as u_2. Similarly, assume that along P_{v_1,u_2}, the last vertex that also belongs to $v_1P_{s_2,d_1}$ is v_1 otherwise we can relabel the last such vertex as v_1. We then argue that P_{v_1,u_2} also does not touch $P_{s_2,d_1}u_1$. Otherwise e_2' cannot be a critical 1-edge cut in $G\backslash\{e_1,e_2\}$ separating $\{s_1,s_2\}$ and u_2. With the above construction, P_{v_1,u_2} touches P_{s_2,d_1} and P_{s_1,d_2} only at v_1 and u_2, respectively. Symmetrically, P_{v_2,u_1} touches P_{s_1,d_2} and P_{s_2,d_1} only at at v_2 and u_1, respectively. We have thus established the lower figure-eight knot.

By a similar argument, we can also establish the upper figure-eight knot. That is $P_{v_2',u_1'}$ touches P_{s_1,d_2} and P_{s_2,d_1} only at v_2' and u_1', respectively. And $P_{v_1',u_2'}$ touches P_{s_2,d_1} and P_{s_1,d_2} only at v_1' and u_2', respectively. For the following, we first assume that the upper and the lower figure-eight knots do not share any common edge and construct a coding scheme for this scenario (as illustrated in Fig. 10). After the introduction of the coding scheme, we then discuss the cases in which the upper and the lower figure-eight knots do share some common edges.

To this end, we will convert Fig. 10 to a special network in Fig. 11 and construct a rate-1 PINC scheme for the particular

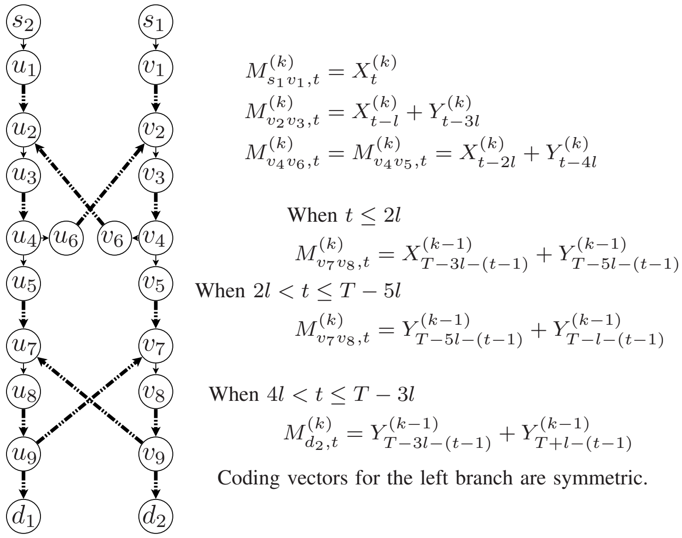


Fig. 11. Detailed construction for a vertical concatenation of two figure-eight knots $\binom{\infty}{\infty}$.

graph in Fig. 11. Fig. 11 has the same two figure-eight knots as in Fig. 10. We use thick, dashed-dotted arrows in Fig. 11 to represent independent paths while the regular thin arrows represent single edges. We assume that each independent path in Fig. 11 is of the same length $(l-1)$. To convert Fig. 10 to a form in Fig. 11, we simply replace each "arrow" in Fig. 10 by an independent path of equal finite length $l < \infty$. This is always possible as the effect of elongating the paths is equivalent to introducing more delay and each edge $e = uv$ can introduce an arbitrary amount of delay by postponing its transmission. Therefore by noting that we only need to introduce a finite amount of delay to make each route be of the same delay, we can assume each independent path in Fig. 11 is of the same finite length $(l-1)$ without loss of generality. For easy reference, we relabel the nodes as in Fig. 11. For example, the total delay from v_2 to v_4 is $1 + (l-1) = l$.

We again divide the total duration into K periods of size T, and the total duration is thus KT. Packet $X^{(k)}_t$ is sent along s_1v_1 in the t-th second of the k-th period. Similarly $Y^{(k)}_t$ is sent along s_2u_1 in the t-th second of the k-th period. Due to the l-second delay of each independent path, we let v_2v_3 carries coded packet $X^{(k)}_{t-l}+Y^{(k)}_{t-3l}$ in the t-th second of the k-th period. The X component of the v_2v_3 packet follows from the P_{s_1,v_2} path, which has an l-second delay. The Y component follows from the $P_{s_2,u_2}P_{u_2,u_4}P_{u_4,v_2}$, which has a $3l$-second delay. If the subscript $(t-l)$ (resp. $(t-3l)$) is less than one, the corresponding component $X^{(k)}_{t-l}$ (resp. $Y^{(k)}_{t-3l}$) is simply zero. Any interference caused by the cyclic feedback loop $P_{v_4,u_2}P_{u_2,u_4}P_{u_4,v_2}$ can be cancelled since all packets received by v_4 come from v_2, and v_2 thus has complete knowledge about the interference emitted from v_4. Symmetrically, u_2u_3 carries coded packet $X^{(k)}_{t-3l} + Y^{(k)}_{t-l}$ in the t-th second of the k-th period. For the v_4v_5 and the v_4v_6 edges, we let the message be

$$M^{(k)}_{v_4v_5,t} = M^{(k)}_{v_4v_6,t} = X^{(k)}_{t-2l} + Y^{(k)}_{t-4l},$$

where the $X^{(k)}_{t-2l}$ message follows from the s_1-to-v_4 path of length $2l$ and the $Y^{(k)}_{t-4l}$ message follows from the s_2-to-v_4 path of length $4l$. Symmetrically,

$$M^{(k)}_{u_4u_5,t} = M^{(k)}_{u_4u_6,t} = X^{(k)}_{t-4l} + Y^{(k)}_{t-2l}.$$

The above is the full description of the packets carried in the upper figure-eight knot.

For the first period,[7] the lower figure-eight knot $(\{v_7, v_8, v_9, d_2, u_7, u_8, u_9, d_1\})$ remains idle. For the t-th second in the k-th period, $1 \leq t \leq 2l, k > 1$, v_7v_8 carry

$$M^{(k)}_{v_7v_8,t} = X^{(k-1)}_{T-3l-(t-1)} + Y^{(k-1)}_{T-5l-(t-1)}. \quad (14)$$

Namely, v_7 buffers the packets it has received in the previous $(k-1)$-th period, and sends them along v_7v_8 in the current k-th period but in a reverse order. Symmetrically, u_7u_8 carries

$$M^{(k)}_{u_7u_8,t} = X^{(k-1)}_{T-5l-(t-1)} + Y^{(k-1)}_{T-3l-(t-1)} \quad (15)$$

in the t-th second in the k-th period.

When $2l < t \leq T-5l$, v_7 is still able to transmit packet $X^{(k-1)}_{T-3l-(t-1)} + Y^{(k-1)}_{T-5l-(t-1)}$ but at the same time also starts to receive packet $X^{(k-1)}_{T-3l-(t-1)} + Y^{(k-1)}_{T-l-(t-1)}$, a $2l$-second-delayed version of $M^{(k)}_{u_7u_8,t}$ in (15). By sending the difference of the buffered packet and the recently received packet, v_7v_8 sends

$$M^{(k)}_{v_7v_8,t} = Y^{(k-1)}_{T-5l-(t-1)} - Y^{(k-1)}_{T-l-(t-1)}$$

when $2l < t \leq T-5l$. Symmetrically,

$$M^{(k)}_{u_7u_8,t} = \begin{cases} X^{(k-1)}_{T-5l-(t-1)} + Y^{(k-1)}_{T-3l-(t-1)} & \text{if } 1 \leq t \leq 2l \\ X^{(k-1)}_{T-5l-(t-1)} - X^{(k-1)}_{T-l-(t-1)} & \text{if } 2l < t \leq T-5l \end{cases}.$$

Again, we cancel the effects of the cyclic feedback loops (ex: $P_{v_9,u_7}P_{u_7,u_9}P_{u_9,v_7}$) by the existing knowledge at the coding nodes u_7 and v_7. The additional $2l$-second-delays of the paths $P_{v_7,v_9}P_{v_9,d_2}$ and $P_{u_7,u_9}P_{u_9,d_1}$ imply that d_1 and d_2 receive uncorrupted packets

$$M^{(k)}_{d_1,t} = X^{(k-1)}_{T-3l-(t-1)} + X^{(k-1)}_{T+l-(t-1)}$$
$$M^{(k)}_{d_2,t} = Y^{(k-1)}_{T-3l-(t-1)} + Y^{(k-1)}_{T+l-(t-1)}$$

for $4l < t \leq T-3l$, and it can be easily shown that the received packets are linearly independent. With the k-th period, $k > 1$, being capable of transmitting $T-7l$ independent packets, the overall rate of the system is thus

$$\frac{(K-1)(T-7l)\log(q)}{KT}.$$

When both K and T are sufficiently large, we have a rate-1 PINC solution.

An observation is that Cases 1 and 2 (and Cases 3.1 and 3.2) are very similar to its acyclic network counterpart, although we have to use a slightly more complicated approach for cyclic network such as the *unwrap* operations and the super-period structure. Nonetheless, *all coding coefficients in Cases 1 and 2 (and Cases 3.1 and 3.2) can be chosen randomly*

[7] We assume T is much larger than l, and discuss the packets carried by the lower figure-eight knot based on different values of t.

except for the two decoding 1-edge cuts, which is similar to that for the acyclic network. Case 3.3 (Figs. 10 and 11) is a unique scenario for cyclic networks. The network now has to cooperate as a whole. For example, our construction carefully gauges the delay between each path, introduces artificial delays, buffers the packets, and reverses the sending order of each packet. Case 3.3 shows that in order to achieve the capacity of a cyclic network, it is critical to design jointly inter- and intrasession network coding even for the simple 2-unicast scenario.

In the last part of this proof, we consider the cases in which the upper and the lower figure-eight knots share some edges. We will reuse the construction of the six path segments in (13) (see Fig. 10 for illustration). The potential interfering effects are discussed as follows.

- The case in which P_{v_1,u_2} and P_{v_2,u_1} share some edge: When the two paths share an edge, let M_{v_1,u_2} and M_{v_2,u_1} denote the messages that should be carried originally if the two paths do not share that common edge. Then we let the shared edge carry $c_1M_{v_1,u_2} + c_2M_{v_2,u_1}$ for some randomly chosen coefficients c_1 and c_2, which multiplexes two messages on a single edge. For all the following cases in which a pair of paths share an edge, we use this random addition to multiplex the symbols together.
  The interference caused by P_{v_1,u_2} and P_{v_2,u_1} sharing the same edge will have zero effect from the coding's perspective. The reason is that the message along the P_{v_2,u_1} path will now induce interference along the P_{v_1,u_2} paths. However, u_2 is able to completely remove the induced interference as all the information available at v_2 was sent by u_2. Symmetrically, u_1 is able to completely remove the interference sent by v_1. After the removal of the interference, the proposed coding scheme works in this scenario as well.
- Similarly, in the case that $P_{v'_1,u'_2}$ and $P_{v'_2,u'_1}$ share some edge, u'_2 and u'_1 can completely remove the interference from a coding's perspective.
- The case in which P_{v_2,u_1} and $P_{v'_2,u'_1}$ share an edge. This case is impossible. Otherwise, there will exist a $\{s_1, s_2\}$-to-u_1 path without using e'_1, e_1, and e_2, which contradicts that e'_1 is a critical 1-edge cut separating $\{s_1, s_2\}$ and u_1 in $G\backslash\{e_1, e_2\}$.
- The last case is when P_{v_2,u_1} and $P_{v'_1,u'_2}$ share some common edge. In this case, the information of the lower figure-eight knot will be an interference to u'_2 (and potentially also an interference to u'_1 if $P_{v'_1,u'_2}$ and $P_{v'_2,u'_1}$ share an edge). Recall that in the k-th period, the upper figure-eight knot performs coding over the k-th generation while the lower figure-eight knot performs coding over the $(k-1)$-th generation. Since in the beginning of the k-th period u'_2 has obtained the complete knowledge of the $(k-1)$-th generation in the previous $(k-1)$-th period, in the k-th period u'_2 can completely remove the interference caused by the symbols of the $(k-1)$-th generation. Similarly, u'_1 can also remove any interference caused by the lower figure-eight knot. The coding scheme thus works verbatim for the upper figure-eight knot.
  On the other hand, the information of the upper figure-eight knot will now be the interference of the lower figure-eight knot. We first note that by our construction, u_1 is not reachable from v'_2 in $G\backslash\{e'_1, e_2\}$. Therefore, if there is any additional interference along the P_{v_2,u_1} caused by the upper and lower figure-eight knots sharing edges, the interference must come only from v'_1 instead of v'_2. Otherwise, if there is any direct interference from v'_2 along P_{v_2,u_1}, then there exists a path entering u_1 without using e'_1 and e_2. Therefore, we only need to cancel the interference emitted from v'_1 on the P_{v_2,u_1} path. For this scenario, since u_1 is also receiving information directly from v'_1 along the P_{s_2,d_1} path, u_1 can completely remove the interference from v'_1. It may be possible that the interference caused by P_{v_2,u_1} and $P_{v'_1,u'_2}$ sharing an edge also enters u_2 if P_{v_2,u_1} and P_{v_1,u_2} share some edge. We rule out this case by the following arguments. Since u_2 must not be reachable from v'_1 in $G\backslash\{e_1, e'_2\}$, the interference caused by P_{v_2,u_1} and $P_{v'_1,u'_2}$ share some common edge must not reach u_2 along P_{v_1,u_2}. Otherwise there exists a path entering u_2 without using e_1 and e'_2, which contradicts that e'_2 is a critical 1-edge cut separating $\{s_1, s_2\}$ and u_2 in $G\backslash\{e_1, e_2\}$. By the above reasoning, our coding scheme works verbatim for the lower figure-eight knot.
- All other cases can be obtained by symmetry.

In summary, the original coding scheme in Fig. 11 takes care of the primary interferences along the path segments of the figure-eight knots. Any secondary interference caused by P_{v_2,u_1}, P_{v_1,u_2}, $P_{v'_2,u'_1}$, and $P_{v'_1,u'_2}$ cross-talking with each other can be canceled out locally at nodes u'_1, u'_2, u_1, and u_2. ■

Remark: In [16], [19], *symbol-based convolutional network codes* are used to achieve the multicast capacity of a cyclic network by incorporating the delay element D. More explicitly, the source messages become $X(D) = \sum_{t=1}^{\infty} X_t D^{t-1}$ and $Y(D) = \sum_{t=1}^{\infty} Y_t D^{t-1}$. Each coded symbol is $M_e(D) = \sum_{t=1}^{\infty} M_{e,t} D^{t-1} = A(D)X(D) + B(D)Y(D)$ for some polynomials $A(D)$ and $B(D)$. Although the symbol-based convolutional network codes can achieve the multicast capacity for cyclic networks, one can show that the capacity of Case 3.3 is not achievable by any symbol-based convolutional network codes. This is due to the fact that the delay-element D cannot capture the necessary buffering and packet-order-reversing performed at nodes u_7 and v_7. Case 3.3 thus serves as an example that the intersession-network-coding capacity of symbol-based convolutional network codes is strictly smaller than that of a block network code.

V. The Complexity Analysis

This section provides the proofs of Propositions 1 and 2 that the complexity of deciding whether a PINC solution exists is polynomial time with respect to $|G| + |\mathbf{d}_1| + |\mathbf{d}_2|$ for both 2-multicast acyclic networks and 2-unicast cyclic networks.

Proof of Propositions 1 and 2: The cut-based condition in Theorem 1 enables the following scheme to determine the existence a PINC solution for 2-multicast acyclic networks and for 2-unicast cyclic network. We need some prerequisite complexity results, which apply to both cyclic and acyclic networks.

- For any two nodes u and v, whether v is reachable from u can be decided in polynomial time of $|G|$.
- For any three nodes u, v, and w, identifying the critical 1-edge cut separating $\{u, v\}$ from w can be achieved in polynomial time of $|G|$. (One simply needs to construct any arbitrary path P from one of $\{u, v\}$ to w and then identify the edges in P that are also a 1-edge cut separating $\{u, v\}$ and w. Among those 1-edge cuts, choose the one that is the farthest away from w.)

The following 5-step algorithm decides whether there exists a PINC solution for a given 2-multicast acyclic network or a given 2-unicast cyclic network.

Step 1: Check all $d_{1,i} \in \mathbf{d}_1$ whether they are reachable from s_1. If any one of them is not reachable from s_1, **return NEGATIVE.** Check all $d_{2,j} \in \mathbf{d}_1$ whether they are reachable from s_2. If any one of them is not reachable from s_2, **return NEGATIVE.**

Step 2: For all $d_{1,i} \in \mathbf{d}_1$, find its critical 1-edge cut e_i. Similarly, find the critical 1-edge cut e_j for all $d_{2,j} \in \mathbf{d}_2$.

Step 3: For all $d_{1,i} \in \mathbf{d}_1$, test whether it is reachable from s_1 in the subgraph $G \backslash \{e_j : d_{2,j} \in \mathbf{d}_2\}$. If the answer to any one of them is negative, **return NEGATIVE.**

Step 4: For all $d_{2,j} \in \mathbf{d}_2$, test whether it is reachable from s_2 in the subgraph $G \backslash \{e_i : d_{1,i} \in \mathbf{d}_1\}$. If the answer to any one of them is negative, **return NEGATIVE.**

Step 5: If the answers in Steps 1, 3, and 4 are all positive, **return POSITIVE.**

The correctness of the above algorithm for 2-multicast acyclic networks and for 2-unicast cyclic networks follows directly by Theorem 1. The polynomial running time with respect to $|G| + |\mathbf{d}_1| + |\mathbf{d}_2|$ is straightforward from analyzing the complexity of each step. ■

VI. Topological Analysis for 2-Unicast Acyclic Networks

In the following, we provide detailed sketches of the proofs. Specifically, we describe how to establish the topological relationship in Theorem 5 from the perspective of edge-disjointness. The proofs of path independence (interior vertex-disjointness) are straightforward and some of them are omitted for streamlining the discussion.

Corollary 1 states that the existence of a network coding solution is equivalent to either the existence of two edge-disjoint paths (2-EDPs) or the existence of path collections $\mathcal{P}$ and $\mathcal{Q}$ with controlled edge overlap. For the following, we assume that each source (resp. destination) has exactly one outgoing (resp. incoming) edge, all sources and destinations are distinct, and there exist no 2-EDPs connecting (s_1, d_1) and (s_2, d_2), which, together with the feasibility assumption, implies that each destination d_i is reachable from both s_1 and s_2. We also assume that the graph G of interest is *minimal,* namely, any proper subgraph $G' \subsetneq G$ will not admit any $\mathcal{P}$ and $\mathcal{Q}$ paths satisfying Corollary 1.

Proof of Theorem 5: Let $e_1 = u_1 v_1$ denote the critical 1-edge cut separating $\{s_1, s_2\}$ from d_1. Similarly, $e_2 = u_2 v_2$ denotes the critical 1-edge cut separating $\{s_1, s_2\}$ and d_2. There are two major cases depending on whether there exist 2-EDPs P_{s_1,u_2} and P_{s_2,u_2} in $G \backslash e_1$.

Case 1: There exist no 2-EDPs P_{s_1,u_2} and P_{s_2,u_2} in $G \backslash e_1$. Since e_1 is a critical 1-edge cut, there exist 2-EDPs P_{s_1,u_1} and P_{s_2,u_1}. Let P_{v_1,d_1} denote an arbitrarily chosen path connecting v_1 and d_1. The relationship among P_{s_1,u_1}, P_{s_2,u_1}, and P_{v_1,d_1} are illustrated in Fig. 12(a). Since e_2 is a critical 1-edge cut, there must exist a P_{v_1,d_2} path otherwise there will be 2-EDPs P_{s_1,u_2} and P_{s_2,u_2} in $G \backslash e_1$. Among vertices shared by P_{v_1,d_2} and P_{v_1,d_1}, let w_1 denote the vertex that is the closest to d_2. Since G satisfies Corollary 1, there exists a P_{s_2,d_2} path not using e_1. Among all vertices shared by P_{s_2,d_2} and $V(P_{s_1,u_1}) \cup V(P_{s_2,u_1})$, let w_2 denote the vertex that is the closest to d_2. Among all vertices shared by $w_2 P_{s_2,d_2}$ and $w_1 P_{v_1 d_2}$, let w_3 denote the vertex that is the farthest away from d_2. Note that $w_2 \neq w_3$, otherwise there is a cycle from w_2 to e_1 to w_1 and back to $w_3 = w_2$ (see Fig. 12(a) for illustration). Depending on whether $w_2 \in P_{s_1,u_1}$ or $w_2 \in P_{s_2,u_1}$, we have the following two subcases.[8]

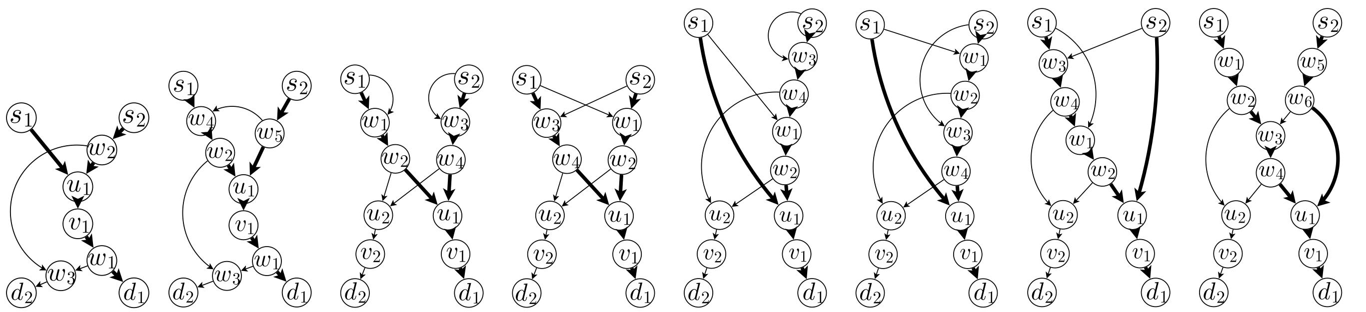
Case 1.1: $w_2 \in P_{s_2,u_1}$ (see Fig. 12(a) for illustration of the topological relationship among w_1, w_2, and w_3). In this case, consider the following two paths

$$P_{s_1,u_1} u_1 v_1 P_{v_1,d_1} \text{ and } P_{s_2,u_1} w_2 P_{s_2,d_2} w_3 P_{v_1,d_2}. \quad (16)$$

$P_{s_2,u_1} w_2 P_{s_2,d_2} w_3 P_{v_1,d_2}$ is edge-disjoint from P_{v_1,d_1} since $P_{s_2,u_1} w_2 P_{s_2,d_2} w_3 P_{v_1,d_2}$ does not use the 1-edge cut $e_1 = u_1 v_1$. $P_{s_2,u_1} w_2 P_{s_2,d_2} w_3 P_{v_1,d_2}$ is also edge-disjoint from P_{s_1,u_1} due to the construction of w_2 as the shared vertex closest to d_2. Therefore the above two paths in (16) are edge-disjoint, which contradicts the initial assumption that G contains no 2-EDPs connecting (s_1, d_1) and (s_2, d_2).

Case 1.2: $w_2 \in P_{s_1,u_1}$. In this subcase, we define two additional nodes w_4 and w_5 as follows. Among all vertices shared by $P_{s_2,d_2} w_2$ and $P_{s_1,u_1} w_2$, let w_4 denote the vertex that is the closest to s_1. Among the vertices shared by $P_{s_2,d_2} w_4$ and P_{s_2,u_1}, let w_5 denote the one that is the farthest away from s_2. See Fig. 12(b) for illustration of the topological relationship among w_1 to w_5. From Fig. 12(b), it is easy to see that *Case 1.2 is topologically identical to a grail in Fig. 5(c)* from the edge-disjointness perspective. To establish the interior-vertex-disjointness among all path segments, we either use the construction that w_1 to w_5 are the farthest or the closest node along a given path, or we use contradiction based on the non-existence of two edge-disjoint paths. For example, w_1 must be distinct from w_3, otherwise there will exist 2-EDPs. Path segment connecting (w_1, w_3) must be independent from the path segment connecting (w_2, w_3) because the construction of w_3 as the farthest node away from d_2. By exhaustively enumerating all case combinations and devising proofs similar to the given two examples, the interior-vertex disjointness for Case 1.2 can be established. Detailed case discussion can be found in Appendix A

[8]Two subcases may be active simultaneously. The argument of either one of them will be sufficient for this proof.



(a) Case 1.1 (b) Case 1.2 (c) Case 2.1 (d) Case 2.2 (e) Case 2.3 (f) Case 2.4 (g) Case 2.5 (h) Case 2.6

Fig. 12. Cases 1.1 to 2.6 for the proof of Theorem 5 in Section VI. The thick arrows represent the paths P_{s_1,u_1}, P_{s_2,u_1} and P_{v_1,d_1} from the perspective of d_1. The thin arrows represent the paths P_{s_1,u_2}, P_{s_2,u_2} and P_{v_2,d_2} from the perspective of d_2.

Case 2: There exist 2-EDPs P_{s_1,u_2} and P_{s_2,u_2} in $G\backslash e_1$ and there exist 2-EDPs P_{s_1,u_1} and P_{s_2,u_1} in $G\backslash e_2$. Let P_{v_1,d_1} denote an arbitrarily chosen path connecting v_1 and d_1 and similarly does P_{v_2,d_2}. Among all edges shared by P_{s_1,u_2} and $E(P_{s_1,u_1}) \cup E(P_{s_2,u_1})$, let w_1w_2 denote the edge that is the closest to d_2. This choice of w_1w_2 is always possible since P_{s_1,u_2} and P_{s_1,u_1} share at least one edge, the unique outgoing edge of s_1. Among all edges shared by P_{s_2,u_2} and $E(P_{s_1,u_1}) \cup E(P_{s_2,u_1})$, let w_3w_4 denote the edge that is the closest to d_2. We then have six subcases.

Case 2.1: $w_1w_2 \in P_{s_1,u_1}$ and $w_3w_4 \in P_{s_2,u_1}$ (see Fig. 12(c) for illustration). In this case, consider the following two paths

$$P_{s_1,u_1}u_1v_1P_{v_1,d_1} \text{ and } P_{s_2,u_1}w_3w_4P_{s_2,u_2}u_2v_2P_{v_2,d_2}. \quad (17)$$

By the construction of w_3w_4 being the shared edge closest to d_2 and by the construction of u_1v_1 being a critical 1-edge cut, $P_{s_1,u_1}u_1v_1P_{v_1,d_1}$ is edge-disjoint from $w_4P_{s_2,u_2}u_2$. More explicitly, $w_4P_{s_2,u_2}u_2$ is edge-disjoint from P_{s_1,u_1} since w_3w_4 is constructed as the shared edge that is closest to d_2. $w_4P_{s_2,u_2}u_2$ is edge-disjoint from u_1v_1 since we are in Case 2. $w_4P_{s_2,u_2}u_2$ is edge-disjoint from $v_1P_{v_1,d_1}$ since $u_1v_1 \notin w_4P_{s_2,u_2}u_2$ is a critical 1-edge cut. As a result, the two paths in (17) are edge-disjoint, which contradicts the initial assumption that there are no such 2-EDPs.

Case 2.2: $w_1w_2 \in P_{s_2,u_1}$ and $w_3w_4 \in P_{s_1,u_1}$ (see Fig. 12(d) for illustration). In this case, the following two paths

$$P_{s_1,u_1}u_1v_1P_{v_1,d_1} \text{ and } P_{s_2,u_1}w_1w_2P_{s_1,u_2}u_2v_2P_{v_2,d_2}$$

are edge-disjoint by similar reasons as those discussed in Case 2.1, which contradicts again the assumption that there are no such 2-EDPs.

Case 2.3: $w_1w_2 \in P_{s_2,u_1}$, $w_3w_4 \in P_{s_2,u_1}$, and w_1w_2 is a downstream edge of w_3w_4 (see Fig. 12(e) for illustration). In this case, the following two paths

$$P_{s_1,u_1}u_1v_1P_{v_1,d_1} \text{ and } P_{s_2,u_1}w_1w_2P_{s_2,u_2}u_2v_2P_{v_2,d_2}$$

are edge-disjoint, which contradicts the assumption that there are no such 2-EDPs.

Case 2.4: $w_1w_2 \in P_{s_2,u_1}$, $w_3w_4 \in P_{s_2,u_1}$, and w_1w_2 is an upstream edge of w_3w_4 (see Fig. 12(f) for illustration). In this case, the following two paths

$$P_{s_1,u_1}u_1v_1P_{v_1,d_1} \text{ and } P_{s_2,u_1}w_3w_4P_{s_2,u_2}u_2v_2P_{v_2,d_2}$$

are edge-disjoint, which contradicts the assumption that there are no such 2-EDPs.

Case 2.5: $w_1w_2 \in P_{s_1,u_1}$, $w_3w_4 \in P_{s_1,u_1}$, and w_1w_2 is a downstream edge of w_3w_4 (see Fig. 12(g) for illustration). In this case, the following two paths

$$P_{s_1,u_2}w_1w_2P_{s_1,u_1}u_1v_1P_{v_1,d_1} \text{ and } P_{s_2,u_2}u_2v_2P_{v_2,d_2}$$

are edge-disjoint, which contradicts the assumption that there are no such 2-EDPs.

Case 2.6: $w_1w_2 \in P_{s_1,u_1}$, $w_3w_4 \in P_{s_1,u_1}$, and w_1w_2 is an upstream edge of w_3w_4. Note that both w_1w_2 and w_3w_4 are constructed from d_2's perspective. In this final case, we have to consider the d_1's perspective as well. Among all edges shared by P_{s_2,u_1} and $E(P_{s_1,u_2}) \cup E(P_{s_2,u_2})$, let w_5w_6 denote the edge that is the closest to d_1. Among all edges shared by P_{s_1,u_1} and $E(P_{s_1,u_2}) \cup E(P_{s_2,u_2})$, let w_7w_8 denote the edge that is the closest to d_1. If the relationship between w_5w_6 and w_7w_8 is one of the symmetric versions of Cases 2.1–2.5, then the proof is complete. So the remaining case is to consider that $w_5w_6 \in P_{s_2,u_2}$, $w_7w_8 \in P_{s_2,u_2}$, and w_5w_6 is an upstream edge of w_7w_8. In this case, one can easily show that $w_3w_4 = w_7w_8$ as this edge is the shared edge that is the closest to $\{d_1, d_2\}$. See Fig. 12(h) for illustration. From the perspective of edge-disjointness, Fig. 12(h) is clearly a butterfly. Again the interior-vertex-disjointness can be established by exhaustively considering all pairs of path segments. Any shared vertex will result in either 2-EDPs or violating the minimality of G. For example, if path segments P_{w_4,u_2} and P_{w_2,u_2} share at least one common interior vertex w, among the shared vertices we can choose the w^* that is the farthest away from u_2 and construct new $P'_{w_4,u_2} = P_{w_4,u_2}$ and $P'_{w_2,u_2} = P_{w_2,u_2}w^*P_{w_4,u_2}$, which contradicts the minimality of G. Detailed case discussion can be found in Appendix B

The discussions of Cases 1.1 to 2.6 have considered all possible cases, and the proof is thus complete.

■

VII. Conclusion

In this work, we have provided a graph-theoretic, flow-based, characterization theorem for pairwise intersession network coding on directed acyclic/cyclic networks with two simple multicast sessions. More explicitly, network coding is allowed only between two symbols (for acyclic networks) or between two strings of symbols (for cyclic networks) and the new characterization theorem determines whether any given network admits such a pairwise intersession network coding solution. Practically, restricting the number of to-coded sessions is an appealing solution for delay-sensitive data, such as multi-resolution video multicast, since generally the larger the number of the to-be-coded sessions, the longer the delay in an asynchronous network.

Based on a new *controlled edge-overlap condition*, the proposed characterization theorems have generalized the *edge-disjoint path* characterization for non-coded solutions and included the well-studied butterfly graph as a special case. Various aspects of pairwise intersession network coding have been discussed, including the sufficiency of linear codes and the complexity advantages of identifying coding opportunities versus identifying non-coded transmission opportunities. For the simplest scenario of two unicast sessions in an acyclic network, the corresponding topological analysis has been conducted. An exhaustive list of three representative structures, the two edge-disjoint paths, the butterfly, and the grail, has been identified, and the associated bandwidth-optimal and coding-optimal conditions have been discussed.

We conclude this paper by discussing new implications of pairwise intersession network coding on practical systems and some future directions.

- *New capacity inner bounds for multiple multicast sessions.* Existing works search for the butterfly structure and use them to construct capacity inner bounds when N unicast sessions are present in an acyclic network [26]. The topological analysis Theorem 5 shows that if a pattern-search approach is used, one needs only to search for two different structures: the butterflies and the grails in Fig. 5, and the consideration of any other structures is redundant. It is an open problem how the pattern-search approach can be generalized to multicast sessions and even to cyclic networks. Instead of searching for ad-hoc patterns, the flow-based characterization focuses on *finding good paths* and provides a more general method of constructing new capacity inner bounds for multiple multicast sessions and their corresponding linear network coding schemes. Some preliminary results along this direction can be found in [15], [27].
- *Information decomposition analysis for pairwise intersession network coding.* For a single multicast session with source s and destinations $\{d_i\}$, a bandwidth-optimal network coding solution uses only the edges participating in the max flows between s and d_i for some i. When the minimal bandwidth usage is known to the network designer, the information decomposition analysis in [7] focuses on finding the simplest coding scheme on the bandwidth-optimal network and its results lead to solutions that are both coding- and bandwidth-optimal. The results in this work describe what is the "minimal bandwidth usage" when network coding is performed between two multicast sessions. Based upon this new knowledge, we are interested in generalizing the information-decomposition analysis and deciding the coding- and bandwidth-optimal solutions for pairwise intersession network coding with two multicast sessions. This direction will generalize our discussion in Section III-C about the coding and bandwidth optimal solutions for two unicast sessions.
- *Graph-theoretic characterization for multiple multicast sessions.* From the information-theoretic perspective, it is a notoriously challenging problem to find a complete characterization of the network coding capacity with multiple multicast sessions [16], [24], [25], [30]. This work shows that finding the network coding capacity region (at least for some restricted classes of linear intersession network coding schemes) is a more tractable problem from a graph-theoretic perspective. The decoupled choices of the $\mathcal{P}$ and $\mathcal{Q}$ path selections mirror the min-cut/max-flow theorem for the single multicast problem as in the latter case, the max-flow between (s, d_i) is also identified independently from other (s, d_j), $j \neq i$. The controlled edge-overlap condition also generalizes the edge-disjointness characterizations of non-coding solutions. The proposed concept of combining decoupled path selections with controlled edge-overlap thus serves as a precursor to the full understanding of general multiple-multicast-session problems.

APPENDIX A
DETAILED CASE DISCUSSION OF CASE 1.2 IN THE TOPOLOGICAL ANALYSIS

In this appendix, we follow directly from the discussion of Case 1.2 in the proof of Theorem 5 in Section VI. We reuse the notations defined in the proof of Theorem 5. We use Fig. 12(b) for illustration purposes only.

The interior-vertex-disjointness (independence) of the path segments in Fig. 12(b) can be derived in the following way.

- By our construction P_{s_1,u_1} and P_{s_2,u_1} do not use u_1v_1. Also by construction P_{w_5,w_4} and P_{w_2,w_3} are parts of P_{s_2,d_2} that does not use u_1v_1. P_{w_3,d_2} is part of P_{v_1,d_2} that does not use u_1v_1. Therefore P_{u_1,d_1} is independent from the following five path segments: P_{s_1,u_1}, P_{s_2,u_1}, P_{w_4,w_5}, P_{w_2,w_3}, and P_{w_3,d_2}. Otherwise, there will be a walk from $\{s_1, s_2\}$ to d_1 without using u_1v_1, which contradicts that u_1v_1 is a critical 1-edge cut. By the construction of w_1 as the closest shared vertex between P_{v_1,d_2} and P_{v_1,d_1}, we also have P_{w_1,w_3} is vertex-disjoint from P_{u_1,d_1} except for the end node w_1. The interior-vertex-disjointness is thus established for P_{u_1,d_1}.
- P_{w_1,d_2} is vertex-disjoint from P_{s_1,u_1}, P_{s_2,u_1}, and P_{w_5,w_4}, otherwise there exists a cycle in the acyclic network. By the construction of w_2 and w_3 as the closest and the furthest such nodes with respect to d_2, respectively, we also have P_{w_1,d_2} is vertex-disjoint from P_{w_2,w_3} except for the end point w_3. The interior-vertex-disjointness is thus established for P_{w_1,d_2}.
- By the construction of w_2 being the closest such node with respect to d_2, we also have P_{w_2,w_3} is edge-disjoint from $w_5P_{s_2,u_1}$. Therefore, the following two paths:

$$P_{s_1,u_1}w_2P_{w_2,w_3}w_3P_{w_1,d_2} \text{ and } P_{s_2,u_1}u_1v_1P_{v_1,d_1} \quad (18)$$

must also be edge-disjoint. As a result, these two edge-disjoint paths must not share any vertex, otherwise there will be two EDPs directly connecting (s_1, d_1) and (s_2, d_2). Hence, P_{w_2,w_3} is vertex-disjoint from P_{s_2,u_1}, which also implies that $w_2 \neq w_5$. By the construction of w_2 being the closest such node with respect to d_2, P_{w_2,w_3} is vertex-disjoint from P_{s_1,u_1} except for the end node w_2. The interior-vertex-disjointness is thus established for P_{w_2,w_3}.

- By the same argument as in the previous statement, we have P_{s_2,u_1} is vertex-disjoint from $P_{s_1,u_1}w_2$, which implies that $w_4 \neq w_5$. By the construction of w_4 and w_5 as the closest and the furthest such nodes with respect to s_1 and s_2, respectively, we have P_{s_2,u_1} is vertex-disjoint from P_{w_5,w_4} except for the end node w_5. For the following, we will establish that P_{s_2,u_1} and $w_2P_{s_1,u_1}$ are vertex-disjoint except for the end point u_1. Since G is acyclic, we must have $P_{s_2,u_1}w_5$ and $w_2P_{s_1,u_1}$ are vertex-disjoint. We thus only need to show that $w_5P_{s_2,u_1}$ and $w_2P_{s_1,u_1}$ are vertex-disjoint except for the end point u_1. Suppose not, let w_6 denote any vertex that is shared by $w_5P_{s_2,u_1}$ and $w_2P_{s_1,u_1}$ but not equal to u_1. Then remove all edges in $w_6P_{s_1,u_1}$. In this case, one can check that in the remaining proper subgraph, Theorem 1 is still satisfied, which contradicts the minimality of G. Therefore $w_5P_{s_2,u_1}$ and $w_2P_{s_1,u_1}$ are vertex-disjoint except for the end point u_1. The interior-vertex-disjointness is thus established for P_{s_2,u_1}.
- It remains to show that P_{s_1,u_1} is vertex-disjoint from P_{w_5,w_4} except for the end node w_4. This holds naturally due to our construction of w_4. The one final case is to show that $w_4 \neq w_2$. If $w_4 = w_2$, then our previous statements show that the following two paths:

$$\begin{aligned}&P_{s_1,u_1}u_1v_1P_{v_1,d_1}\\ &\text{and } P_{s_2,u_1}w_5P_{w_5,w_4}P_{w_2,w_3}w_3P_{w_1,d_2}\end{aligned} \tag{19}$$

must be edge-disjoint. This contradicts the assumption that there are no two EDPs directly connecting (s_1, d_1) and (s_2, d_2). As a result $w_4 \neq w_2$.

The interior-vertex-disjointness of Case 1.2 is thus established. The graph G indeed contains a subdivision of the grail topology in Fig. 5(c).

APPENDIX B
DETAILED CASE DISCUSSION OF CASE 2.6 IN THE TOPOLOGICAL ANALYSIS

In this appendix, we follow directly from the discussion of Case 2.6 in the proof of Theorem 5 in Section VI. We reuse the notations defined in the proof of Theorem 5. We use Fig. 12(h) for illustration purposes only.

We first focus on the following 10 nodes s_1, s_2, w_2, w_3, w_4, w_6, u_1, u_2, d_1, and d_2, and the corresponding 11 path segments.

$$\begin{aligned}&P_{s_1,w_2} \triangleq P_{s_1,u_1}w_2, \quad P_{w_2,u_2} \triangleq w_2P_{s_1,u_2},\\ &P_{w_2,w_3} \triangleq w_2P_{s_1,u_1}w_3, \quad P_{s_2,w_6} \triangleq P_{s_2,u_1}w_6,\\ &P_{w_6,w_3} \triangleq w_6P_{s_2,u_2}w_3, \quad P_{w_6,u_1} \triangleq w_6P_{s_2,u_1},\\ &P_{w_3,w_4} \triangleq w_3P_{s_1,u_1}w_4, \quad P_{w_4,u_2} \triangleq w_4P_{s_2,u_2},\\ &P_{w_4,u_1} \triangleq w_4P_{s_1,u_1}, \quad P_{u_2,d_2} \triangleq u_2v_2P_{v_2,d_2},\\ &P_{u_1,d_1} \triangleq u_1v_1P_{v_1,d_1}.\end{aligned}$$

We now establish the edge-disjointness among these 11 path segments. For 10 out of the 11 path segments except for P_{w_6,w_3}, the edge-disjointness follows directly from the acyclicness of the network, the two edges u_1v_1 and u_2v_2 being the critical 1-edge cuts, the construction of EDP pairs $(P_{s_1,u_1}, P_{s_2,u_1})$ and $(P_{s_1,u_2}, P_{s_2,u_2})$, and the construction of w_1w_2, w_3w_4, and w_5w_5 edges as the most downstream such edges. The only possible edge-overlaps are between P_{w_6,w_3} and P_{s_1,w_2}, and between P_{w_6,w_3} and P_{w_2,w_3}.

Due to the symmetry of Case 2.6, there is an alternative choice of P_{s_1,w_2} and P_{s_2,w_6}. Namely, we can let

$$P'_{s_1,w_2} \triangleq P_{s_1,u_2}w_2, \quad P'_{s_2,w_6} \triangleq P_{s_2,u_2}w_6.$$

For the following, we show that if P_{w_6,w_3} and P_{s_1,w_2} share an edge, then P_{w_2,w_3} and P'_{s_2,w_6} must be edge-disjoint. Conversely, if P_{w_2,w_3} and P'_{s_2,w_6} share an edge, then P_{w_6,w_3} and P_{s_1,w_2} must be edge-disjoint. Suppose P_{w_6,w_3} and P_{s_1,w_2} share an edge e_1 and simultaneously P_{w_2,w_3} and P'_{s_2,w_6} share an edge e_2. Then there is a cyclic walk $P_{w_2,w_3}e_2P'_{s_2,w_6}P_{w_6,w_3}e_1P_{s_1,w_2}$, which contradicts the acyclicness assumption. As a result, we can assume that P_{w_6,w_3} and P_{s_1,w_2} are edge-disjoint without loss of generality.

We then establish the edge-disjointness between P_{w_6,w_3} and P_{w_2,w_3}. Suppose P_{w_6,w_3} and P_{w_2,w_3} share an edge. Then there exists a node $w \neq w_3$ that is shared by P_{w_6,w_3} and P_{w_2,w_3}. Among all such w, choose the w^* that is the most upstream one. We can then relabel w^* as w_3 and redefine

$$\begin{aligned}&P_{w_6,w_3} \triangleq w_6P_{s_2,u_2}w^*, \quad P_{w_2,w_3} \triangleq w_2P_{s_1,u_1}w^*,\\ &\text{and } P_{w_3,w_4} \triangleq w^*P_{s_1,u_1}w_4.\end{aligned} \tag{20}$$

This new w_3 thus guarantees the edge-disjointness between P_{w_6,w_3} and P_{w_2,w_3}.

Based on the above arguments, we have established the topological mapping between the 10 nodes and 11 path segments to that of the butterfly from an edge-disjoint perspective. To complete the proof of the vertex-based topological analysis, we first establish the distinction among the eight nodes: s_1, s_2, w_2, w_3, w_4, w_6, u_1, u_2, d_1, and d_2, which implies that no path is degenerate. Note that the remaining proof relies only on the edge-disjointness of the 11 path segments and does not rely on how the 11 path segments were constructed.

- We must have $w_4 \neq u_1$. Otherwise, the two edge-disjoint paths $P_{s_1,w_2}P_{w_2,w_3}P_{w_3,w_4}P_{u_1,d_1}$ and $P_{s_2,w_6}P_{w_6,u_1}P_{w_4,u_2}P_{u_2,d_2}$ contradict the assumption that there are no 2 EDPs connecting (s_1, d_1) and

(s_2, d_2). By symmetry, we also have $w_4 \neq u_2$, $w_2 \neq w_3$, $w_6 \neq w_3$.

- We must have $u_2 \neq u_1$. Otherwise, the two edge-disjoint paths $P_{s_1,w_2}P_{w_2,u_2}P_{u_1,d_1}$ and $P_{s_2,w_6}P_{w_6,u_1}P_{u_2,d_2}$ contradict the assumption that there are no 2 EDPs connecting (s_1, d_1) and (s_2, d_2). By symmetry, we also have $w_2 \neq w_6$.
- We must have that $u_2 \neq w_2$, $u_2 \neq w_6$, and $u_2 \neq w_3$ due to the acyclicness of the graph. Similarly, $u_1 \neq w_2$, $u_1 \neq w_6$, $u_1 \neq w_3$, $w_4 \neq w_2$, and $w_4 \neq w_6$.
- By construction, $s_1 \neq s_2$, $d_1 \neq d_2$. Since $u_2 \neq v_2$, we must have $u_2 \neq d_2$. Similarly, we have $u_1 \neq d_1$, $s_1 \neq w_1$, and $s_2 \neq w_5$.

The above statements thus establish the vertex distinction among s_1, s_2, w_2, w_3, w_4, w_6, u_1, u_2, d_1, and d_2. The remaining task is to show that the 11 path segments are interior-vertex-disjoint. The proofs can be divided into three main categories. Type I: The proofs are based on graph minimality. In this type of proofs, we simply rename one of the nodes to ensure interior-vertex-disjointness in a similar way as when we replace w_3 by a w^* in the previous statements. Type II: The proofs are based on the contradiction to the assumption that there exist no EDPs connecting (s_1, d_1) and (s_2, d_2). Type III: The proofs are based on the acyclicness assumption. For the following, we list all case combinations and will explicitly specify which type of the proofs one should use to establish the interior-vertex-disjointness.

- Consider P_{u_1,d_1}.
  - P_{u_1,d_1} is interior-vertex-disjoint (IVD) from P_{s_1,w_2}, P_{s_2,w_6}, P_{w_2,w_3}, P_{w_6,w_3}, P_{w_6,u_1}, P_{w_3,w_4}, and P_{w_4,u_1} due to the acyclicness assumption (Type III).
  - P_{u_1,d_1} is IVD from P_{w_4,u_2}. Otherwise, suppose there exists a shared vertex w, then we have the following two edge-disjoint paths $P_{s_1,w_2}P_{w_2,w_3}P_{w_3,w_4}P_{w_4,u_2}wP_{u_1,d_1}$ and $P_{s_2,w_6}P_{w_6,u_1}P_{u_1,d_1}wP_{w_4,u_2}P_{u_2,d_2}$, which contradicts the assumption that there are no 2 EDPs connecting (s_1, d_1) and (s_2, d_2) (Type II). Similarly, P_{u_1,d_1} is IVD from P_{w_2,u_2} and P_{u_2,d_2} based on a Type II proof.
- Since P_{u_2,d_2} is symmetric to P_{u_1,d_1}, we have also established the interior-vertex-disjointness for P_{u_2,d_2}.
- Consider P_{w_6,u_1}.
  - P_{w_6,u_1} is IVD from P_{s_2,w_6} due to the acyclicness assumption (Type III).
  - P_{w_6,u_1} is IVD from P_{w_6,w_3}. Otherwise, among the interior vertices shared by P_{w_6,u_1} and P_{w_6,w_3}, choose w^* that is the most downstream one. Relabel w^* as w_6, change the path segments accordingly, and remove unused edges. After relabeling, the vertex distinction must still hold for the 10 vertices. Otherwise we can repeat the vertex distinction arguments in our previous discussion and lead to a contradiction. Similarly, after relabeling, the interior-vertex-disjointness for P_{u_1,d_1} and P_{u_2,d_2} must still hold. Otherwise, we can repeat the interior-vertex-disjointness arguments in our previous discussion and lead to a contradiction.
    Using the new w_6, we thus establish that P_{w_6,u_1} is IVD from P_{w_6,w_3} (Type I).
  - P_{w_6,u_1} is IVD from P_{w_4,u_1}. Otherwise, among the interior vertices shared by P_{w_6,u_1} and P_{w_4,u_1}, choose w^* that is the most upstream one. Relabel w^* as u_1, change the path segments accordingly, and remove the unused edges. This Type I proof thus follows similarly to the previous Type I proof.
  - P_{w_6,u_1} is IVD from P_{w_3,w_4}, P_{w_4,u_2}, P_{w_2,u_2}, P_{w_2,w_3}, and P_{s_1,w_2}, which can be established by a Type II proof.
- Consider P_{w_4,u_1}.
  - P_{w_4,u_1} is IVD from P_{s_1,w_2}, P_{s_2,w_6}, P_{w_2,w_3}, P_{w_6,w_3}, and P_{w_3,w_4} due to the acyclicness assumption (Type III).
  - P_{w_4,u_1} is IVD from P_{w_4,u_2}. Otherwise, among the interior vertices shared by P_{w_4,u_1} and P_{w_4,u_2}, choose w^* that is the most downstream one. Relabel w^* as w_4, change the path segments accordingly, and remove the unused edges. This Type I proof thus follows similarly to the previous Type I proofs.
  - P_{w_4,u_1} is IVD from P_{w_2,u_2}, which can be established by a Type II proof.
- Since P_{w_4,u_2} is symmetric to P_{w_4,u_1}, we have also established the interior-vertex-disjointness for P_{w_4,u_2}.
- Since P_{w_2,u_2} is symmetric to P_{w_6,u_1}, we have also established the interior-vertex-disjointness for P_{w_2,u_2}.
- Consider P_{w_3,w_4}.
  - P_{w_3,w_4} is IVD from P_{s_1,w_2}, P_{s_2,w_6}, P_{w_2,w_3}, and P_{w_6,w_3} due to the acyclicness assumption (Type III).
- Consider P_{w_6,w_3}.
  - P_{w_6,w_3} is IVD from P_{s_2,w_6} due to the acyclicness assumption (Type III).
  - P_{w_6,w_3} is IVD from P_{w_2,w_3}. Otherwise, among the interior vertices shared by P_{w_6,w_3} and P_{w_2,w_3}, choose w^* that is the most upstream one. Relabel w^* as w_3, change the path segments accordingly, and remove the unused edges. This Type I proof thus follows similarly to the previous Type I proofs.
  - P_{w_6,w_3} is IVD from P_{s_1,w_2}, which can be established by a Type II proof.
- Since P_{w_2,w_3} is symmetric to P_{w_6,w_3}, we have also established the interior-vertex-disjointness for P_{w_2,w_3}.
- Consider P_{s_2,w_6}.
  - P_{s_2,w_6} is IVD from P_{s_1,w_2}, which can be established by a Type II proof.

The above statements establish the interior-vertex-disjointness for all 11 path segments. The proof is thus complete.

References

[1] 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.

[2] J. Cannons and K. Zeger, "Network coding capacity with a constrained number of coding nodes," *IEEE Trans. Inform. Theory*, vol. 54, no. 3, pp. 1287–1291, March 2008.

[3] J. Cheriyan and M. Salavatipour, "Hardness and approximation results for packing Steiner trees," *Algorithmica*, vol. 45, no. 1, pp. 21–43, 2006, the special issue for selected papers of ESA 2004.

[4] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, October 2003.

[5] 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.

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

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

[8] 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.

[9] N. Harvey, K. Jain, L. Lau, C. Nair, and Y. Wu, "Conservative network coding," in *Proc. 44th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, September 2006.

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

[12] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen., "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inform. Theory*, vol. 51, 2005.

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

[14] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard, "The importance of being opportunistic: Practical network coding for wireless environments," in *Proc. 43rd Annual Allerton Conf. on Comm., Contr., and Computing*, 2005.

[15] A. Khreishah, C.-C. Wang, and N. Shroff, "An optimization-based rate control for communication networks with inter-session network coding," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

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

[17] G. Kramer 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.

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

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

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

[21] ——, "Network coding: the case of multiple unicast sessions," in *Proc. 42nd Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, Illinois, USA, September 2004.

[22] 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.

[23] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding form unicast sessions: the case for coding aware routing," in *Proc. 26th IEEE Conference on Computer Communications (INFOCOM)*. Anchorage, USA, May 2007.

[24] 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, December 2003.

[25] ——, "A separation theorem for single-source network coding," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 1861–1871, May 2006.

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

[27] 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. IEEE Int'l Symp. Inform. Theory*. Nice, France, June 2007.

[28] Y. Wu, "On constructive multi-source network coding," in *Proc. IEEE Int'l Symp. Inform. Theory*. Seattle, USA, July 2006, pp. 1349–1353.

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

[30] X. Yan, R. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," in *Proc. IEEE Int'l Symp. Inform. Theory*. Nice, France, 2007.

PLACE PHOTO HERE

Chih-Chun Wang (M'06) is currently an Assistant Professor of the School of Electrical and Computer Engineering of Purdue University. He received the B.E. degree in E.E. from National Taiwan University, Taipei, Taiwan in 1999, the M.S. degree in E.E., the Ph.D. degree in E.E. from Princeton University in 2002 and 2005, respectively. He worked in Comtrend Corporation, Taipei, Taiwan, as a design engineer in 2000 and spent the summer of 2004 with Flarion Technologies, New Jersey. In 2005, he held a post-doctoral researcher position in the Electrical Engineering Department of Princeton University. He joined Purdue University as an Assistant Professor in 2006. His current research interests are in the graph-theoretic and algorithmic analysis of iterative decoding and of network coding. Other research interests of his fall in the general areas of networking, optimal control, information theory, detection theory, and coding theory.

Dr. Wang received the National Science Foundation Faculty Early Career Development (CAREER) Award in 2009.

PLACE PHOTO HERE

Ness B. Shroff (F'07) received his Ph.D. degree from Columbia University, NY, in 1994 and joined Purdue university as an Assistant Professor. At Purdue, he became Professor of the School of Electrical and Computer Engineering in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined The Ohio State University as the Ohio Eminent Scholar of Networking and Communications, a chaired Professor of ECE and CSE. He is also a guest chaired professor of Wireless Communications and Networking in the department of Electronic Engineering at Tsinghua University. His research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, pricing, and security of these networks.

Dr. Shroff is a past editor for IEEE/ACM Trans. on Networking and the IEEE Communications Letters and current editor of the Computer Networks Journal. He has served as the technical program co-chair and general co-chair of several major conferences and workshops, such as the IEEE INFOCOM 2003, ACM Mobihoc 2008, IEEE CCW 1999, and WICON 2008. He was also a co-organizer of the NSF workshop on Fundamental Research in Networking (2003) and the NSF Workshop on the Future of Wireless Networks (2009).

Dr. Shroff is a fellow of the IEEE. He received the IEEE INFOCOM 2008 best paper award, the IEEE INFOCOM 2006 best paper award, the IEEE IWQoS 2006 best student paper award, the 2005 best paper of the year award for the Journal of Communications and Networking, the 2003 best paper of the year award for Computer Networks, and the NSF CAREER award in 1996 (his INFOCOM 2005 paper was also selected as one of two runner-up papers for the best paper award)