Alignment-Based Network Coding for Two-Unicast-Z Networks

Citation	Zeng, Weifei et al. "Alignment-Based Network Coding for Two-Unicast-Z Networks." IEEE Transactions on Information Theory 62, 6 (June 2016): 3183–3211 © 2016 Institute of Electrical and Electronics Engineers (IEEE)
As Published	http://dx.doi.org/10.1109/TIT.2016.2553676
Publisher	Institute of Electrical and Electronics Engineers (IEEE)
Version	Original manuscript
Citable link	http://hdl.handle.net/1721.1/111003
Terms of Use	Creative Commons Attribution-Noncommercial-Share Alike
Detailed Terms	http://creativecommons.org/licenses/by-nc-sa/4.0/

Alignment based Network Coding for Two-Unicast-Z Networks

Weifei Zeng
RLE, MIT
Cambridge, MA, USA
weifei@mit.edu

Viveck R. Cadambe
Department of Electrical Engineering
The Pennsylvania State University
viveck@engr.psu.edu

Muriel Médard
RLE, MIT
Cambridge, MA, USA
medard@mit.edu [*][†]

Abstract

In this paper, we study the wireline *two-unicast-Z* communication network over directed acyclic graphs. The two-unicast-Z network is a two-unicast network where the destination intending to decode the second message has apriori side information of the first message. We make three contributions in this paper:

1. We describe a new linear network coding algorithm for two-unicast-Z networks over directed acyclic graphs. Our approach includes the idea of interference alignment as one of its key ingredients. For graphs of a bounded degree, our algorithm has linear complexity in terms of the number of vertices, and polynomial complexity in terms of the number of edges.
2. We prove that our algorithm achieves the rate-pair $(1, 1)$ whenever it is feasible in the network. Our proof serves as an alternative, albeit restricted to two-unicast-Z networks over directed acyclic graphs, to an earlier result of Wang et. al. which studied necessary and sufficient conditions for feasibility of the rate pair $(1, 1)$ in two-unicast networks.
3. We provide a new proof of the classical max-flow min-cut theorem for directed acyclic graphs.

[*] This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under award No. FA9550-13-1-0023.

[†] This work was also supported by Viveck R. Cadambe's startup grant provided by the Department of Electrical Engineering at the Pennsylvania State University.

1 Introduction

Since the advent of network coding [1], characterizing the capacity region of networks of orthogonal noiseless capacitated links, often termed the network coding capacity, has been an active area of research. Inspired by the success of *linear* network coding for *multicast* networks [2,3], a significant body of work has been devoted to understanding the design and performance limits of linear network codes even for non-multicast communication scenarios. Previous approaches to linear network code design for non-multicast settings have at least one of two drawbacks: the network code design is restricted to a limited set of network topologies, or the approach has a prohibitive computational complexity. The goal of our paper is develop ideas and algorithms that fill this gap in literature. The main contribution of our paper is the development of a low-complexity linear network coding algorithm for *two-unicast-Z* networks - a network communication setting with two independent message sources and two corresponding destination nodes, where one destination has a priori knowledge of the undesired message (See Fig. 1). We begin with a brief survey of previous, related literature.

1.1 Related Work

The simplest linear network code is in fact the technique of routing, which is used to show the max-flow min-cut theorem. In addition to the single-source single-destination setting of the max-flow min-cut theorem, routing has been shown to be optimal for several classes of networks with multiple source messages in [4–6][1]. The technique of random linear network coding, which is optimal for multicast networks [2, 8], is also shown to achieve capacity for certain non-multicast networks [2, 9, 10][2]. The ideas of random linear network coding and routing have been combined to develop network coding algorithms for arbitrary networks in [11]. Nonetheless, it is well known that the techniques of random linear network coding and routing are, in general, sub-optimal linear network codes for networks with multiple sources.

To contrast the optimistic results of [2–4, 6, 7, 9], pessimistic results related to the performance limits of linear network codes have been shown in [12–14]. Through deep connections between linear network coding and matroidal theory, it has been shown that linear network codes are sub-optimal in general [12, 15]. In fact, most recently [14], it has been shown that even the best linear code cannot, in general, achieve the capacity of the *two-unicast* network.

Literature has also studied the computational limits of construction and performance evaluation of linear codes. It is shown in [16] that characterizing the set of rates achievable by *scalar* linear codes[3] is NP-complete. Furthermore, for a given field size, the computational complexity of determining the best (scalar or vector) linear code is associated with challenging open problems related to polynomial solvability [18] and graph coloring [19]. Nonetheless, characterization of the computational complexity of determining the set of all rates achievable by linear network coding remains an open problem. One main challenge in resolving this question is to develop ideas and algorithms for constructing linear network codes for non-multicast settings. The study and devel-

[1] In fact, in [7], routing has been conjectured to achieve network capacity for multiple unicast networks over undirected directed acyclic graphs.

[2] Linear network coding techniques which do not necessarily involve choosing co-efficients randomly have also been studied for multicast and certain non-multicast settings in [3, 10]

[3] We use the term *linear coding* for network codes where all the codewords come from a vector space, and encoding functions are linear operators over this vector space. The term *scalar* linear network coding is used when the dimensionality of this vector space is equal to 1. It is known that vector linear network coding strictly outperforms scalar linear network coding in general [17].

opment of linear network codes for non-multicast settings is also important from an engineering standpoint, especially in current times, because its utility has been recently demonstrated in settings that arise from modeling distributed storage systems [20], wireless networks [21], and content disribution systems [22]. We review the principal approaches to developing linear network codes outside the realm of routing and random linear network coding.

Network Codes over the binary field: Initial approaches to developing codes for non-multicast settings restricted their attention to codes over the binary field. Since the binary field provides a small set of choices in terms of the linear combinations that can be obtained by an encoding node, it is possible to search over the set of all coding solutions relatively efficiently. This idea was exploited to develop a linear programming based network coding algorithms in [23, 24]. Reference [25] presents a noteworthy result that demonstrates the power of carefully designed network codes over the binary field. Through a careful understanding of the network communication graph, [25] characterized the feasibility of rate tuple $(1, 1)$ in two unicast networks. The result of [25] can be interpreted as follows: the rate tuple $(1, 1)$ is achievable if and only if (i) the min-cut between each source and its respective destination is at least equal to 1, and (ii) the *generalized network sharing bound* - a network capacity outer bound formulated in [26] - is at least 2. Furthermore, the rate $(1, 1)$ is feasible if and only if it is feasible through linear network coding over the binary field. In general, however, coding over the binary field does not suffice even for the two unicast network [26].

Interference Alignment: *Interference alignment,* which is a technique discovered in the context of interference management for wireless networks, has recently emerged as a promising tool for linear network code design even for wireline networks. Specifically, an important goal in the design of coding co-efficients for non-multicast networks is to ensure that an unwanted message does not corrupt a desired message at a destination. The unwanted message can be interpreted as an interferer at the destination, and thus, tools from interference management in wireless networks can be inherited into the network coding setting.

Interference alignment was first explicitly identified as a tool for multiple unicast network coding in [27], which describes a class of networks where the asymptotic interference alignment scheme of [28] is applicable. This class of networks has been further studied and generalized in [29,30]. The power of interference alignment for network coding was demonstrated in references [20,31–33]; these references developed alignment-based erasure codes to solve open problems related to minimizing repair bandwidth in distributed data storage systems. Reference [34] used interference alignment to characterize the capacity of classes of the *index coding* problem [35], a sub-class of the class of general network coding capacity problems. The index coding problem is especially important because references [36, 37] have shown an equivalence between the general network coding capacity problem and the index coding problem.

Index Coding Based Approaches: The equivalence of the general network coding problem and the index coding problem, which is established in [36, 37], opens another door to the development of linear network codes for general networks. Specifically, for a given network coding setting, the approach of [37] can be used first to obtain an equivalent index coding setting; then the approaches of [38–41] can be used to develop linear *index* codes. However, these index code design approaches have high computational complexity, since they require solving challenging graph coloring related problems or linear programs whose number of constraints is exponential in terms of the number of users. Because of the nature of the mapping of [37], this means that these approaches require solving linear programs where the number of constraints is exponential number in terms of the number of edges of the network in consideration. Another common approach to obtaining

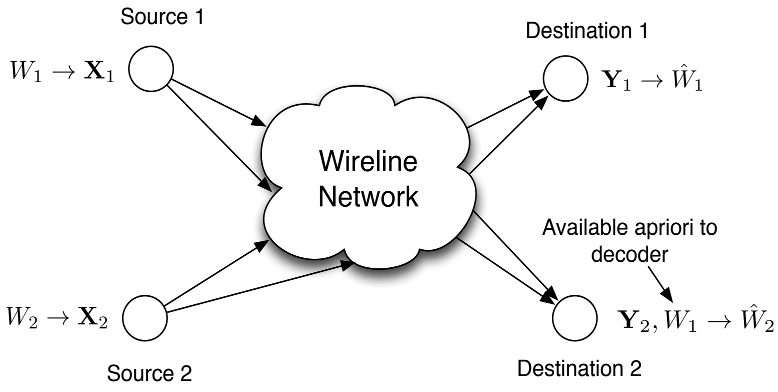


Figure 1: The Two-unicast-Z Network

index coding solutions is given in [42], which connects the rate achievable via linear index coding to a graph functional known as *minrank*. While in principle, the minrank characterizes the rate achievable by the best possible linear index code, the min-rank of a matrix over a given field size is difficult to evaluate, and NP-Hard in general [43]; furthermore, there is no systematic approach to characterizing the field size.

In summary, while recent ideas of interference alignment and connections to index coding broaden the scope of linear network coding, these approaches inherit the main drawbacks of linear network coding. Specifically, network coding approaches outside the realm of routing or random linear network coding are either carefully hand-crafted for a restricted set of network topologies, or their enormous computational complexity inhibits their utility. The motivation of our paper is to partially fill this gap in literature by devising algorithms for linear network coding. We review our contributions next.

1.2 Contributions

The goal of this paper is to devise systematic algorithms for linear network coding that incorporate ideas from interference alignment. In this paper, we focus on *two-unicast-Z* networks over directed acyclic graphs. The two-unicast-Z network communication problem consists of two sources s_1, s_2, two destinations t_1, t_2 and two independent messages W_1, W_2. Message W_i is generated by source s_i and is intended to be decoded by destination t_i. In the two-unicast-Z setting, destination t_2 has apriori side information of the message W_1. Our nomenclature is inspired from the Z-interference channel [44] in wireless communications, where, like our network, only one destination faces interference[4]. In fact, with linear coding, relation between the sources and destinations in the two-unicast-Z setting is the same as a Z-interference channel. We note that the two-unicast-Z network can be interpreted as a two unicast network with an infinitely capacitated link between source 1 and destination 2. Therefore two-unicast-Z networks form a subclass of two-unicast networks.

[4] It is perhaps tempting to consider a network where s_1 is not connected to t_2 and use the nomenclature of the two-unicast-Z for such a network. However, it is worth noting that routing is optimal for such a network and each message achieves a rate equal to the min-cut between the respective source and destination. Therefore, the study of such a network is not particularly interesting from a network capacity viewpoint.

Despite the simplicity of its formulation, little is known about the two-unicast-Z network. For instance, while the generalized network sharing (GNS) bound - a network capacity outerbound formulated in [26] - is loose in general for two-unicast networks, we are not aware of any two-unicast-Z network where the GNS bound is loose. Similarly, while linear network coding is insufficient for two-unicast, it is not known whether linear network coding suffices for two-unicast-Z networks. In this paper, we use two-unicast-Z networks over directed acyclic graphs as a framework to explore our ideas of linear network coding. We make three main contributions in this paper:

(1) In Section 4, we describe an algorithm that obtains linear network codes for two-unicast-Z networks over directed acyclic graphs. Our approach is based on designing coding co-efficients to maximize the capacity of the implied end-to-end Z-interference channels in the network [45]. For graphs whose degree is bounded by some parameter, the complexity of our algorithm is linear in the number of vertices and polynomial in the number of edges. We provide a high level intuitive description of our algorithm in Section 2.

(2) In Section 5, we provide an alternate proof of the classical max-flow min-cut theorem for directed acyclic graphs. Our proof therefore adds to the previous literature that has uncovered several proofs of the theorem [1,46,47]. Our proof is a direct, linear coding based proof, and relies on tools from elementary linear algebra. This is in contrast to previous proofs which rely on graph theoretic results (Menger's theorem) or linear programming. This proof, in fact, inspires our algorithm for two-unicast-Z networks. We provide an intuitive description of the proof in Section 2.

(3) In Section 6, we prove that our algorithm achieves a rate of $(1,1)$ whenever it is feasible. Note that the necessary and sufficient conditions for the feasibility of the rate pair $(1,1)$ has been characterized in [25]. Our proof in Section 6 provides an alternate proof of the feasibility of the rate pair $(1,1)$, when restricted to two-unicast-Z networks over directed acyclic graphs. Like our alternate proof to the max-flow min-cut theorem, our proof of Section 6 also relies significantly on elementary linear algebra.

2 Intuition Behind the algorithm

Consider the two-unicast-Z network described in Fig. 1. In this network, with linear coding at all the nodes in the network, the input output relationships can be represented as

$$\mathbf{Y}_1 = \mathbf{X}_1\mathbf{H}_1 + \mathbf{X}_2\mathbf{H}_2, \quad \mathbf{Y}_2 = \mathbf{X}_2\mathbf{G}_2 \ ,$$

where for $i \in \{1,2\}$, $\mathbf{X}_i$ is a row vector representing the input symbols on the outgoing edges of the ith source, $\mathbf{Y}_i$ is row vector representing the symbols on the incoming edges of the ith destination node followed by interference cancellation with the side information if $i=2$. Note that if $\mathbf{H}_1, \mathbf{H}_2$ are fixed and known, the input-output relations are essentially akin to the Z-interference channel. Using the ideas of El Gamal and Costa [48] for the interference channel, the set of achievable rate pairs (R_1, R_2) can be described (see [45,49,50]) as the rate tuples (R_1, R_2) satisfying

$$R_1 \leq \text{rank}\left(\mathbf{H}_1\right), \quad R_2 \leq \text{rank}\left(\mathbf{G}_2\right) \tag{1}$$

$$R_1 + R_2 \leq \text{rank}\left(\begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\end{bmatrix}\right) + \mathsf{rank}\left([\mathbf{H}_2 \ \ \mathbf{G}_2]\right) - \mathbf{rank}(\mathbf{H}_2) \tag{2}$$

The goal of our algorithm is to specify the linear coding co-efficients at all the nodes in the network, which in turn specifies the matrices $\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2$. Once these matrices are specified, the rate region achieved in the network is specified by (1), (2). Here, we describe our approach to designing the linear coding co-efficients in the two-unicast-Z network.

To describe our intuition, we begin with the familiar single source setting, and describe the ideas of our algorithm restricted to this setting. Note that with a single source and single destination, with linear coding in the network, the end-to-end relationship can be represented as $\mathbf{Y} = \mathbf{XH}$, where $\mathbf{X}$ and $\mathbf{Y}$ respectively represent the symbols carried by the source and destination edges, and $\mathbf{H}$ represents the transfer matrix from the source to the destination edges. We know from classical results that the linear coding co-efficients can be chosen such that the rank of $\mathbf{H}$ is equal to the min-cut of the network. Here, we provide an alternate perspective of this classical result. Our examination of the single-source setting provides a template for our algorithm for the two-unicast-Z network which is formally described in Section 4. Our approach also yields an alternate proof for the max-flow min-cut theorem which is provided in Section 5.

2.1 Algorithm for Single-Unicast Network

We focus on a scenario shown in Fig. 2. Denote the network communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ denotes the set of edges. Now, suppose that, as shown in Figure 2, a linear coding solution has been formulated for $\tilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E} - \{e\})$, where e denotes an edge coming into the destination node. The question of interest here is the following: How do we encode the edge e so that the end-to-end rate is maximized? We assume that our coding strategy is restricted to linear schemes.

Let $\mathbf{X}$ be a $1 \times S$ vector denoting the source symbols input on the S edges emanating from the source node. Let $\mathbf{H}$ denote a $S \times (D-1)$ linear transform between the input and $D-1$ destination edges - all the destination edges excluding edge e. Let $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k$ be $1 \times S$ vectors respectively denoting the linear transform between the source and the k edges coming into edge e, that is, the ith incoming edge. Now, our goal is to design the coding strategy for edge e, that is, to choose scalars $\alpha_1, \alpha_2, \ldots, \alpha_k$ such that the rate of the system

$$\mathbf{Y} = \mathbf{X} \left[\mathbf{H} \quad \sum_{i=1}^{k} \alpha_j \mathbf{p}_j \right]$$

is maximized, given matrix $\mathbf{H}$ and vectors $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k$. Equivalently, the goal is to choose scalars $\alpha_1, \alpha_2, \ldots, \alpha_k$ such that the rank of $\left[\mathbf{H} \quad \sum_{i=1}^{k} \alpha_j \mathbf{p}_j \right]$ is maximized. The solution to this problem is quite straightforward - one can notice that if

$$\text{rank}\left([\mathbf{H} \quad \mathbf{p}_1 \quad \mathbf{p}_2 \quad \ldots \quad \mathbf{p}_k] \right) > \text{rank}\left(\mathbf{H} \right) \tag{3}$$

then the scalars $\alpha_1, \ldots, \alpha_k$ can be chosen such that the rank of $\left[\mathbf{H} \quad \sum_{i=1}^{k} \alpha_j \mathbf{p}_j \right]$ is equal to the rank$\left(\mathbf{H} \right) + 1$. Since rank$\left(\mathbf{H} \right)$ is the rate obtained by the destination if edge e is ignored, the implication is that if (3) is satisfied, then, we can design a linear coding strategy such that edge e provides one additional dimension to the destination. In fact, if (3) is satisfied and the field of operation is sufficiently large, then choosing the scalars $\alpha_1, \ldots, \alpha_k$ randomly, uniformly over the field of operation and independent of each other increases the rank of $\left[\mathbf{H} \quad \sum_{i=1}^{k} \alpha_j \mathbf{p}_j \right]$ by 1, implying the existence a linear coding solution.

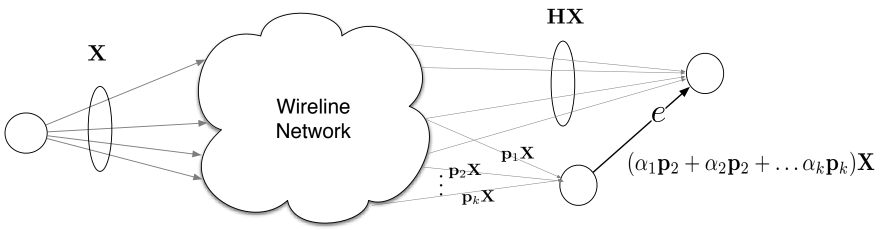


Figure 2: A single-unicast scenario depicted pictorially. The goal is to find scalars $\alpha_1, \ldots, \alpha_k$.

A solution to the scenario of Fig. 2 naturally suggests a linear coding algorithm for the single unicast problem. Suppose we are given a *directed* acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of source edges $S \subset \mathcal{E}$, a set of destination edges $D \subset \mathcal{E}$. Our strategy removes the last topologically ordered edge $e \in D$ and finds a linear coding solution for the remaining graph. That is, specifically, we develop a linear coding solution for $\mathcal{G} = (\mathcal{V}, \mathcal{E} - \{e\})$, with source edges S and destination edges $D - \{e\} \cup \text{In}(v)$, where v represents the tail node of edge e, and $\text{In}(v)$ represents the set of edges incoming on to edge v. Therefore, we have reduced our original problem, which intended to design coding co-efficients for $|\mathcal{E}|$ edges, to one which needs to design coding co-efficients for $|\mathcal{E}| - 1$ edges, albeit with a slightly different set of destination edges in mind. We can now recursively iterate the same procedure to this smaller problem, removing the last edge as per topological ordering at each iteration and modifying the destination edge set accordingly until all the edges are removed except the source edges. A trivial coding solution applies to this graph, which forms a starting point for the recursive algorithm we have described.

While our insight might appear superfluous in the context of the single unicast setting, it does lead to an alternate proof for the max-flow min-cut theorem for directed acyclic graphs. To conclude our discussion, we provide an intuitive description of the proof; the proof is formally provided in Section 5. In our proof, we make the inductive assumption that the max-flow min-cut theorem is valid for the source S and for any destination set which is a subset of $\mathcal{E} - \{e\}$. Under this assumption, we show using ideas from classical multicast network coding literature that the optimal linear coding solutions for the two possible destination sets $D - \{e\}$ and $D - \{e\} \cup \text{In}(v)$ can be combined into a single linear coding solution that simultaneously obtains the min-cut for both destination sets. Then, we use this combined solution along with the solution to Fig. 2 and show that this linear coding solution achieves a rate equal to the min-cut for destination set D. More specifically, we show that if the edge e belongs to a min-cut for destination D, then the inductive assumption implies that, for this combined solution, (3) holds; our strategy of choosing coding co-efficients randomly over the field ensures that a rank that is equal to the min-cut is achieved for graph $\mathcal{G}$ with destination D as well.

2.2 Algorithm for Two-Unicast-Z Networks

Consider a two-unicast-Z network of the form shown in Fig. 3. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of two sets of source edges S_1, S_2, two sets of destination edges T_1, T_2, with the destination 2 being aware of the message of the first source apriori. Now, consider a situation where a coding solution has been formulated for all the edges of the graph, with the exception of edge $e \in \mathcal{D}_1$. We are interested in understanding how to encode the edge e so that the end-to-end rate is maximized.

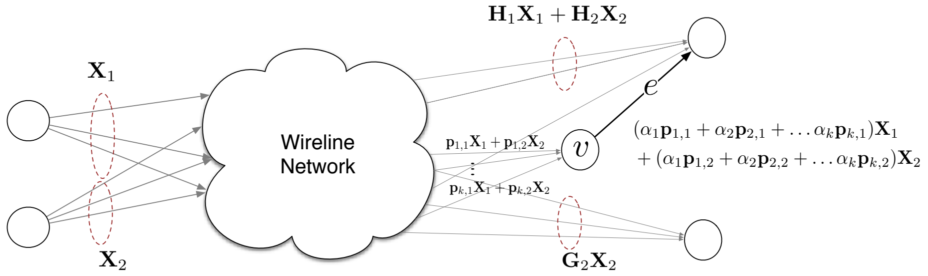

$\mathbf{H}_1\mathbf{X}_1 + \mathbf{H}_2\mathbf{X}_2$
$\mathbf{X}_1$
e
Wireline
Network
$\mathbf{p}_{1,1}\mathbf{X}_1 + \mathbf{p}_{1,2}\mathbf{X}_2$
v
$(\alpha_1\mathbf{p}_{1,1} + \alpha_2\mathbf{p}_{2,1} + \dots \alpha_k\mathbf{p}_{k,1})\mathbf{X}_1$
$+ (\alpha_1\mathbf{p}_{1,2} + \alpha_2\mathbf{p}_{2,2} + \dots \alpha_k\mathbf{p}_{k,2})\mathbf{X}_2$
$\vdots$
$\mathbf{p}_{k,1}\mathbf{X}_1 + \mathbf{p}_{k,2}\mathbf{X}_2$
$\mathbf{X}_2$
$\mathbf{G}_2\mathbf{X}_2$

Figure 3: A two-unicast-Z scenario depicted pictorially. The goal is to find scalars $\alpha_1, \dots, \alpha_k$ to maximize (4).

Our heuristic is based on maximizing the *sum-rate* that is, the right hand side of equation (2). Based on Fig. 3, our goal is to find $\alpha_1, \alpha_2, \dots, \alpha_k$ such that

$$\text{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \sum_{i=1}^{k}\alpha_i\mathbf{p}_{i,1}\\ \mathbf{H}_2 & \sum_{i=1}^{k}\alpha_i\mathbf{p}_{i,2}\end{bmatrix}\right) + \text{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \sum_{i=1}^{k}\alpha_i\mathbf{p}_{i,2} & \mathbf{G}_2\end{bmatrix}\right) - \text{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \sum_{i=1}^{k}\alpha_i\mathbf{p}_{i,2}\end{bmatrix}\right) \quad (4)$$

is maximized. To do so, we examine two cases:

Case 1 If $\mathbf{p}_{1,2}, \dots, \mathbf{p}_{k,2}$ lie in the span of $\mathbf{H}_2$, then, clearly, choosing α_is randomly and uniformly over the field is the best strategy with a probability that tends to 1 as the field size increases. This is because in the third and negative term in (4), the column corresponding to the random linear combination of $\mathbf{p}_{i,2}$'s do not contribute to the rank of the matrix, while random linear coding maximizes the first two terms with high probability.

Case 2 If $\mathbf{p}_{1,2}, \dots, \mathbf{p}_{k,2}$ does not lie in the span of $\mathbf{H}_2$, then the solution is a bit more involved. We divide this case into two sub-cases

Case 2a Suppose that $\mathbf{p}_{1,2}, \dots, \mathbf{p}_{k,2}$ do not lie in the span of $[\mathbf{H}_2 \ \ \mathbf{G}_2]$, then chosing α_is randomly maximizes the expression of (4). In particular, we note that choosing α_is randomly increases the two positive terms and the negative term of (4) by 1, effectively increasing the the expression of (4) by 1, as compared with $\text{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right) + \text{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right) - \text{rank}\left(\mathbf{H}_2\right)$. We later show in Lemma 1 that the expression of (4) cannot be increased by more than 1; this implies the optimality of the random coding approach for the case in consideration here.

Case 2b Suppose that $\mathbf{p}_{1,2}, \dots, \mathbf{p}_{k,2}$ lies in the span of $\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}$, but does not lie in the span of $\mathbf{H}_2$. In this case, the optimal strategy is to choose co-efficients $\alpha_1, \alpha_2, \dots, \alpha_k$ so that $\sum_{i=1}^{k}\alpha_i\mathbf{p}_{i,2}$ is *a random vector in the intersection of the column spaces of* $\mathbf{H}_2$ and $\begin{bmatrix}\mathbf{p}_{1,2} & \mathbf{p}_{2,2} & \dots & \mathbf{p}_{k,2}\end{bmatrix}$. In other words, we intend to *align the local coding vector on edge* e *in the space of* $\mathbf{H}_2$.

From the above discussion, it is interesting to note that we naturally uncover interference alignment in Case 2b as a technique that maximizes the expression of (4). In fact, we show later in Lemma

2 that, if

$$\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \mathbf{p}_{1,1} & \mathbf{p}_{2,1} & \cdots & \mathbf{p}_{k,1}\\ \mathbf{H}_2 & \mathbf{p}_{1,2} & \mathbf{p}_{2,2} & \cdots & \mathbf{p}_{k,2}\end{bmatrix}\right)+\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{p}_{1,2} & \mathbf{p}_{2,2} & \ldots & \mathbf{p}_{k,2} & \mathbf{G}_2\end{bmatrix}\right)$$
$$-\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{p}_{1,2} & \mathbf{p}_{1,2} & \ldots & \mathbf{p}_{k,2}\end{bmatrix}\right)$$
$$> \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)+\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right)-\mathrm{rank}\left(\mathbf{H}_2\right) \qquad (5)$$

then, our choice of $\alpha_1, \alpha_2, \ldots, \alpha_k$ increases (4) by 1 as compared with $\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)+\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right)-\mathrm{rank}\left(\mathbf{H}_2\right)$.

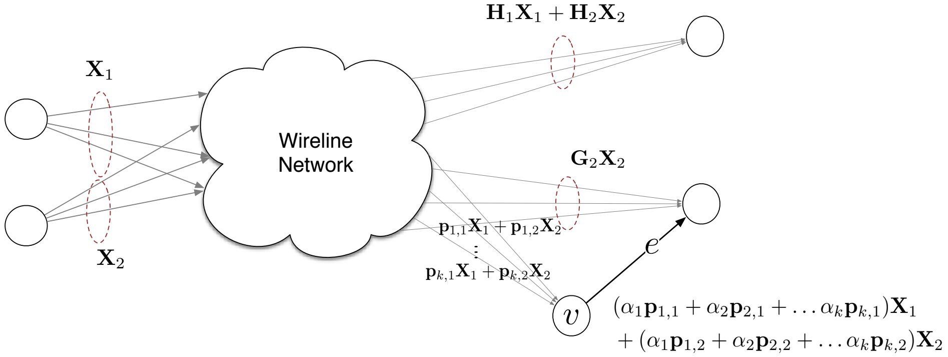

$\mathbf{H}_1\mathbf{X}_1+\mathbf{H}_2\mathbf{X}_2$
$\mathbf{X}_1$
Wireline Network
$\mathbf{G}_2\mathbf{X}_2$
$\mathbf{p}_{1,1}\mathbf{X}_1+\mathbf{p}_{1,2}\mathbf{X}_2$
$\vdots$
$\mathbf{p}_{k,1}\mathbf{X}_1+\mathbf{p}_{k,2}\mathbf{X}_2$
$\mathbf{X}_2$
e
v
$(\alpha_1\mathbf{p}_{1,1}+\alpha_2\mathbf{p}_{2,1}+\ldots\alpha_k\mathbf{p}_{k,1})\mathbf{X}_1$
$+(\alpha_1\mathbf{p}_{1,2}+\alpha_2\mathbf{p}_{2,2}+\ldots\alpha_k\mathbf{p}_{k,2})\mathbf{X}_2$

Figure 4: A two-unicast-Z scenario depicted pictorially. The goal is to find scalars $\alpha_1, \ldots, \alpha_k$ to maximize (4).

Now consider the scenario of Fig. 4. In this scenario, the goal of maximizing the right hand side of (2) is tantamount to choosing scalars $\alpha_1, \alpha_2, \ldots, \alpha_k$ to maximize $\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2 & \sum_{i=1}^{k}\alpha_i\mathbf{p}_{i,2}\end{bmatrix}\right)$. The scenario is similar to the single-unicast problem discussed earlier, and choosing the scalars α_i randomly, uniformly over the field of operation and independent of each other maximizes the sum-rate. In fact, it is easy to see that this strategy improves the sum-rate by 1, if

$$\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)+\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2 & \mathbf{p}_{1,2} & \mathbf{p}_{2,2} & \ldots & \mathbf{p}_{k,2}\end{bmatrix}\right)-\mathrm{rank}\left(\mathbf{H}_2\right) \qquad (6)$$
$$> \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)+\mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right)-\mathrm{rank}\left(\mathbf{H}_2\right). \qquad (7)$$

Remark 1. *Strictly speaking, the sum-rate of the two-unicast-Z network is equal to the minimum of*

$$\mathit{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \sum_{\ell=1}^{k}\alpha_\ell\mathbf{p}_{\ell,1}\end{bmatrix}\right)+\mathit{rank}(\mathbf{G}_2)$$

and the expression of (4). In this paper, for the sake of simplicity, we restrict our attention to maximizing the expression of (4); our ideas and algorithms can be easily modified to maximize the smaller of

$$\mathit{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \sum_{\ell=1}^{k}\alpha_\ell\mathbf{p}_{\ell,1}\end{bmatrix}\right)+\mathit{rank}(\mathbf{G}_2)$$

and (4).

To summarize, we observe from (5) and (7) that an edge e can increase the right hand side of (2) by 1 if, the parent edges of the edge e in combination to the other already existing edges in the destination can increase the right hand side of (2) by at least 1. Furthermore, our strategy to maximize the right hand side of (4) automatically uncovers the idea of alignment. Before proceeding, we briefly explain how the problems of Fig. 3 and 4 can be composed naturally into a recursive algorithm to design linear coding co-efficients for the entire network.

We represent the network as a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices, and $\mathcal{E}$ denotes the set of edges. We begin with the last edge of the graph, that is the edge e with the highest topological order in the graph. This edge e is incident on either the first destination or the second destination. Given a coding solution for the graph $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E} - \{e\})$, we can design a linear coding solution based on the above approach. Therefore, we aim to design a coding solution for the smaller graph $\mathcal{G}_1$. To do this, we add all the parent edges of e to the corresponding destination. That is, if edge e is in destination 1, modify destination 1 in $\mathcal{G}_1$ to include all the edges incoming on to the vertex v, where v is the vertex from which edge e emanates (See Figs. 3). Similarly, if edge e is incident onto destination 2, we remove edge e to reduce the problem to a smaller graph $\mathcal{G}_1$ and all the parent edges of e to destination 2 (See Fig. 4). Now, our goal is to find a linear coding solution to the smaller problem $\mathcal{G}_1$. We proceed similarly by identifying the last topologically ordered edge in $\mathcal{G}_1$ and removing it to obtain $\mathcal{G}_2$, and further modifying the destinations. If we proceed similarly, removing one edge at a time from graph $\mathcal{G}$, we obtain a sequence of graphs $\mathcal{G}_1, \mathcal{G}_2, \ldots,$ to eventually obtain a graph $\mathcal{G}_N$ where the destination edges coincide with the source edges. Starting with a trivial coding solution for $\mathcal{G}_N$, we build a coding solution for the sequence of graphs $\mathcal{G}_N, \mathcal{G}_{N-1}, \ldots, \mathcal{G}_1$ and eventually obtain a coding solution for graph $\mathcal{G}$. Our approach is formally outlined in Section 4. Before we proceed, we note that in our sequence of graphs obtained above, it can transpire that the last topologically ordered edge in one of the graphs belongs to both destinations. We omit an explanation of this scenario here, since our approach in handling this scenario is similar to the one depicted in Fig. 3.

3 System Model

Consider a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ denotes the set of edges. We allow multiple edges between vertices, hence, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V} \times \mathbb{Z}_+$, where $\mathbb{Z}_+$ denotes the set of positive integers. For an edge $e = (u, v, i) \in \mathcal{E}$, we denote $\text{Head}(e) = v$ and $\text{Tail}(e) = u$; in other words, when the direction of the edge is denoted by an arrow, the vertex at the arrow head is the head vertex, and the vertex at the tail of the arrow is the tail vertex of the edge. When there is only one edge between node u and node v, we simply denote the edge as (u, v). For a given vertex $v \in \mathcal{V}$, we denote $\text{In}(v) = \{e \in \mathcal{E} : \text{Head}(e) = v\}$ and $\text{Out}(v) = \{e \in \mathcal{E} : \text{Tail}(e) = v\}$.

In this paper, we focus on the networks with one or more unicast sessions. We define each source as a node in $\mathcal{V}$, while each destination as *a subset of edges* in $\mathcal{E}$. Subsequently, a single unicast network problem Ω can be specified by a 3-tuple $(\mathcal{G}, s, T)$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the underlying graph, $s \in \mathcal{V}$ is the source node and $T \subset \mathcal{E}$ is set of destination edges. Every node in the graph represents an encoding node, and every edge in the graph represents an orthogonal, delay-free, link of unit capacity.

In two-unicast-Z networks, we use the set $\mathcal{S} = \{s_1, s_2\}$, where $s_1, s_2 \in \mathcal{V}$, to denote set of two sources. We use $\mathcal{T} = \{T_1, T_2\}$ to denote the set of two destinations, where T_1 and T_2 each is a set of edges, i.e., $T_i \subset \mathcal{E}, i = 1, 2$. To keep the scenario general, we allow an edge to belong both destinations, i.e., $T_1 \cap T_2$ need not be the null set. Furthermore, the head vertices for edges in the same destination may not have to be the same, i.e., for $e_1, e_2 \in T_i, i = 1, 2$ it is possible that $\text{Head}(e_1) \neq \text{Head}(e_2)$. Without loss of generality, we consider graphs where the edges with the highest topological order belongs to $T_1 \cup T_2$.

In the two-unicast-Z network, the sources s_1 and s_2 generate independent messages W_1 and W_2 respectively. The message W_1 is available a priori to destination T_2. The goal of the two-unicast-Z network is to design encoding functions at every node in the network and decoding functions such that W_1 is recoverable from the symbols carried by the edges in T_1, and W_2 is recoverable from the symbols carried by the edges in T_2 and the side information W_1. Without loss of generality, assume that s_i communicates with at least one edge in T_i for $i \in \{1, 2\}$. Similar to the case of single unicast, we can denote a two-unicast-Z network coding problem Ω using a 3-tuple, i.e. $\Omega = (\mathcal{G}, \mathcal{S}, \mathcal{T})$, where $\mathcal{S} = \{s_1, s_2\}, \mathcal{T} = \{T_1, T_2\}$.

A rate pair (R_1, R_2) is *achievable* if for every $\epsilon > 0, \delta > 0$, there exists a coding scheme which encodes message W_i at a rate $R_i - \delta_i$, for some $0 \leq \delta_i \leq \delta$, such that the average decoding error probability is smaller than ϵ. The capacity region is the closure of the set of all achievable rate pairs.

Topological Order

Since the graphs considered in this paper are directed acyclic graphs, there exists a standard topological order $\text{Ord}_{\mathcal{V}}$ on the set of vertices $\mathcal{V}$ of the graph. The order $\text{Ord}_{\mathcal{V}}$ satisfies the following property: if there is an edge $(u, v, i) \in \mathcal{E}$, then, $\text{Ord}_{\mathcal{V}}(u) < \text{Ord}_{\mathcal{V}}(v)$. We define a partial order $\text{Ord}_{\mathcal{E}}$ on the set of edge $\mathcal{E}$ such that the order of an edge is equal to the order of the tail node of the edge, i.e., $\text{Ord}_{\mathcal{E}}(e) = \text{Ord}_{\mathcal{E}}(\text{Tail}(e)), e \in \mathcal{E}$. Note that all edges sharing the same tail node have the same order. When there is no ambiguity, we omit the subscript in the ordering and simply denote the ordering on the edges (or vertices) as Ord.

Linear Network Coding

In this paper, we consider scalar linear coding, where the encoded symbol along each edge is an element of a finite field $\mathbb{F}$. We use the algebraic framework of linear network coding of [2] to relate the linear coding co-efficients at the vertices of the graph to the encoded symbols carried by the edges. Specifically, we describe a linear coding solution for a network using a local coding matrix $\mathbf{F}$, whose i-th column corresponds to the edge with topological order i and stores the local coding vector on the edge from the symbols carried by its parent edges. The linear transfer matrix of the entire network therefore given by $\mathbf{M} = (\mathbf{I} - \mathbf{F})^{-1}$ and the transfer matrices between sources and destinations can be obtained as submatrices of the network transfer matrix $\mathbf{M}$.

For $i \in \{1, 2\}$, we denote the symbols carried by the edges emanating from source i by the $1 \times |\text{Out}(s_i)|$ vector $\mathbf{X}_i$. Similarly, we denote the symbols carried by the edges in T_i to be the $1 \times |T_i|$ vector $\mathbf{Y}_i$. For a linear coding scheme, we write

$$\mathbf{Y}_1 = \mathbf{X}_1\mathbf{H}_1 + \mathbf{X}_2\mathbf{H}_2 \tag{8}$$

$$\mathbf{Y}_2 = \mathbf{X}_1\mathbf{G}_1 + \mathbf{X}_2\mathbf{G}_2 \ , \tag{9}$$

where $\mathbf{H}_i$ is the $|\text{Out}(s_i)| \times |T_1|$ transfer matrix from $\mathbf{X}_i$ to $\mathbf{Y}_1$ and $\mathbf{G}_i$ is the $|\text{Out}(s_i)| \times |T_2|$ transfer matrix from $\mathbf{X}_i$ to $\mathbf{Y}_2$. Note that the matrices $\mathbf{H}_i$ and $\mathbf{G}_i$ are sub-matrices of the network transfer matrix $\mathbf{M}$ by selecting the rows and columns corresponding to the specified source and destination edges respectively.

For the two-unicast-Z network, the rate pair (R_1, R_2) achieved by a linear coding scheme is characterized by

$$R_1 \leq \text{rank}\left(\mathbf{H}_1\right), \quad R_2 \leq \text{rank}\left(\mathbf{G}_2\right) \ , \tag{10}$$

$$R_1 + R_2 \leq \text{rank}\left(\begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\end{bmatrix}\right) + \text{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right) - \text{rank}\left(\mathbf{H}_2\right) \ . \tag{11}$$

Note that the rate region does not depend on the matrix $\mathbf{G}_1$ since destination 2 cancels the effect of $\mathbf{X}_1\mathbf{G}_1$ using its side information. The rate region can be derived as a simple corollary of the result of [48], which obtains the capacity of a class of deterministic 2-user interference channels. We refer the reader to [45] for a proof.

Notations

The cardinality of a set E is denoted by $|E|$. For sets A and B, $A\backslash B$ denotes the set of elements in A but not in B. For a matrix $\mathbf{A}$, $\mathsf{colspan}(\mathbf{A})$ denotes its column span and $\text{Ker}(\mathbf{A})$ denotes the nullspace of $\mathsf{colspan}(\mathbf{A})$. In a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, For $u, v \in \mathcal{V}, u \rightsquigarrow v$ indicates that u communicates with v, i.e. there is a path from u to v on $\mathcal{G}$. $u \not\rightsquigarrow v$ means that u does not communicates with v on $\mathcal{G}$. For $i, j \in \mathcal{E}$, $i \rightsquigarrow j$ is equivalent to $\text{Out}(i) \rightsquigarrow \text{In}(j)$. We denote sub-matrices of the local and global coding matrices by super-scripts. Specifically, for two edge sets $\mathcal{E}_1, \mathcal{E}_2 \subset \mathcal{E}$, the matrices $\mathbf{F}^{\mathcal{E}_1,\mathcal{E}_2}$ and $\mathbf{M}^{\mathcal{E}_1,\mathcal{E}_2}$ respectively represent $|\mathcal{E}_1| \times |\mathcal{E}_2|$ dimensional sub-matrices of $\mathbf{F}$ and $\mathbf{M}$ derived from the rows corresponding to $\mathcal{E}_1$ and columns corresponding to $\mathcal{E}_2$. For example, we can write $\mathbf{H}_1 = \mathbf{M}^{\text{Out}(s_1),T_1}$.

In the context of the two-unicast-Z networks, a *GNS-cut set* as defined in [26] is a set $\mathcal{Q} \in \mathcal{E}$, such that $\mathcal{Q}$ is

1. a $s_1 - T_1$ cut-set, and,
2. a $s_2 - T_2$ cut-set, and,
3. a $s_2 - T_1$ cut-set.

The *GNS-cut set bound* of a two-unicast-Z network is defined to be the cardinality of the smallest GNS set in the network. It is shown in [26] that the GNS-cut set bound is an information theoretic upper bound on the sum-rate achievable in the network.

4 Recursive, alignment-based, linear network coding algorithm

In this section, we present a scalar linear network code construction for the two-unicast-Z network problem. The algorithm consists of two sub-routines, the *destination reduction* algorithm which is described in Section 4.2, and the *recursive code construction* which is described in Section 4.3. When both of them are run, the recursive coding routine returns the coding matrix $\mathbf{F}$. The rate achieved can be obtained via (10),(11). The sub-routines are pictorially depicted in Fig. 6 for the network in Fig. 5. Before we describe these sub-routines, we begin with some preliminary definitions and lemmas that will be useful in the algorithm description.

4.1 Preliminaries

Definition 1 (Grank)**.** *Given matrices $\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2$ of dimensions $P_1 \times Q_1$, $P_2 \times Q_1$ and $P_2 \times Q_2$ respectively, where P_1, P_2, Q_1, Q_2 are positive integers, the* Grank *is defined as*

$$\textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2) = \textit{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right) + \textit{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right) - \textit{rank}\left(\mathbf{H}_2\right).$$

Note that the Grank is related to the sum-rate of the two-unicast-Z network where $\mathbf{H}_1, \mathbf{H}_2$ and $\mathbf{G}_2$ respectively represent the transfer matrices between source 1 and destination 1, source 2 and destination 1, and source 2 and destination 2.

Remark 2. *We can show that*

$$\textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2) = \min_{\mathbf{G}_1 \in \mathbb{F}^{P_1 \times Q_2}} \textit{rank}\begin{bmatrix}\mathbf{H}_1 & \mathbf{G}_1\\ \mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}.$$

Our use of the term "Grank" is inspired by the above observation which indicates the quantity of interest is closely related to the rank of an appropriate matrix.

Remark 3. *We have shown in [45], that, if $\mathbf{H}_1, \mathbf{H}_2$ and $\mathbf{G}_2$ respectively represent the transfer matrices between source 1 and destination 1, source 2 and destination 1, and source 2 and destination 2, then Grank$(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2)$ is upper bounded by minimum generalized network sharing cut value of the network.*

We state some useful properties of the Grank next.

Lemma 1. *Let $\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2, \mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ be matrices with entries from a finite field $\mathbb{F}$, respectively having dimensions $P_1 \times Q_1$, $P_2 \times Q_1$, $P_2 \times Q_2$, $P_1 \times M$, $P_2 \times M$ and $P_2 \times N$, for positive integers P_1, P_2, Q_1, Q_2, M, N. Then the following properties hold.*

(i) Concatenation of columns to matrices does not reduce Grank.

$$\textit{Grank}\left([\,\mathbf{H}_1\ \ \mathbf{A}\,], [\,\mathbf{H}_2\ \ \mathbf{B}\,], [\,\mathbf{G}_2\ \ \mathbf{C}\,]\right) \geq \textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2).$$

(ii) Concatenating M column increases the Grank by at most M.

$$\textit{Grank}([\mathbf{H}_1\ \ \mathbf{A}], [\mathbf{H}_2\ \ \mathbf{B}], \mathbf{G}_2) \leq \textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2) + M$$

$$\textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, [\mathbf{G}_2\ \ \mathbf{C}]) \leq \textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2) + N.$$

(iii) Concatenation of linearly dependent columns does not change the Grank. Suppose that

$$\textit{colspan}\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \subseteq \textit{colspan}\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}.$$

$$\textit{colspan}\begin{bmatrix}\mathbf{C}\end{bmatrix} \subseteq \textit{colspan}\begin{bmatrix}\mathbf{G}\end{bmatrix},$$

then

$$\textit{Grank}\left([\,\mathbf{H}_1\ \ \mathbf{A}\,], [\,\mathbf{H}_2\ \ \mathbf{B}\,], [\,\mathbf{G}_2\ \ \mathbf{C}\,]\right) = \textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2).$$

Statement (i) follows from submodularity of the rank function. Statements (ii) and (iii) follow from elementary properties of the rank of a matrix and the definition of the Grank. We omit a proof of the lemma here.

Next, we state a lemma that will be useful later on in generating our linear coding solutions.

Lemma 2. *Let $\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2, \mathbf{A}, \mathbf{B}$ be matrices with entries from a finite field $\mathbb{F}$, respectively having of dimensions $P_1 \times Q_1$, $P_2 \times Q_1$, $P_2 \times Q_2, P_1 \times M$ and $P_2 \times M$. Suppose that*

$$\textit{Grank}([\mathbf{H}_1 \ \ \mathbf{A}], [\mathbf{H}_2 \ \ \mathbf{B}], \mathbf{G}_2) > \textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2)$$

Then, there exists a $M \times 1$ column vector $\mathbf{f}$ such that

$$\textit{Grank}([\mathbf{H}_1 \ \ \mathbf{Af}], [\mathbf{H}_2 \ \ \mathbf{Bf}], \mathbf{G}_2) = \textit{Grank}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2) + 1 \tag{12}$$

Furthermore, if

(i) colspan$(\mathbf{B}) \subset$ colspan$([\mathbf{H}_2 \ \ \mathbf{G}_2])$

(ii) colspan$(\mathbf{B}) \not\subset$ colspan$(\mathbf{H}_2)$,

then, choosing a vector $\mathbf{v}$ randomly from the null space of the column space of $[\mathbf{H}_2 \ \ \mathbf{B}]$ and setting $\mathbf{f}$ to be the last M entries of $\mathbf{v}$ satisfies (12) with a probability that approaches 1 as the field size $|\mathbb{F}|$ increases. In this case, the vector $\mathbf{f}$ satisfies the following property: $\mathbf{Bf} \in$ colspan$(\mathbf{H}_2)$. If (i) or (ii) are not satisfied, then picking the entries of $\mathbf{f}$ randomly and uniformly over the field satisfies (12) with a probability that approaches 1 as the field size increases.

Since adding a single column can increase the Grank by at most 1, the vector $\mathbf{f}$ that satisfies (12) maximizes the Grank$([\mathbf{H}_1 \ \ \mathbf{Af}], [\mathbf{H}_2 \ \ \mathbf{Bf}], \mathbf{G}_2)$. The vector $\mathbf{f}$ will be useful in obtaining the code construction in the recursive coding routine. The approach of choosing $\mathbf{f}$ randomly to satisfy (12) follows the spirit of random linear network coding [8].

4.2 Destination reduction

The destination reduction algorithm takes the original problem $\Omega = (\mathcal{G}, \mathcal{S}, \mathcal{T})$ and generates a sequence of $N+1$ ordered two-unicast-Z network problems, for some $N \in \mathbb{Z}^+$, starting with the original problem itself. We denote the sequence of problems as $\mathbb{P} = \left(\Omega^{(0)}, \Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(N)}\right)$, where $\Omega^{(0)} = \Omega$ and $\Omega^{(i)} = (\mathcal{G}, \mathcal{S}, \mathcal{T}^{(i)})$ with $\mathcal{T}^{(i)} = \left\{T_1^{(i)}, T_2^{(i)}\right\}$ being the destination sets for the problem number i. In particular, all the problems have the same underlying graph $\mathcal{G}$ and source set $\mathcal{S}$, but different destination sets, i.e., $\mathcal{T}^{(i)} \neq \mathcal{T}^{(j)}, i \neq j$. The algorithm is formally described in Algorithm 1, in which the key procedure is to sequentially generate $\Omega^{(i+1)}$ from the previous problem $\Omega^{(i)}$. We describe the process informally here.

Recall that in a directed acyclic graph, there is a total ordering Ord on the vertices of the graph. Also recall that Ord induces a partial ordering on the edges, where the set of edges of the same topological order share a common tail node. In brief, the destination reduction algorithm obtains problem $\Omega^{(i+1)}$ from $\Omega^{(i)}$ as follows. We find all the highest topologically ordered edges in the union of the two destination sets. In each destination set, if it contains any of these edges, we replace them with their immediate parent edges. Specifically, given $\Omega^{(i)} = (\mathcal{G}, \mathcal{S}, \mathcal{T}^{(i)})$, let E

Algorithm 1 Destination reduction algorithm

```
procedure REDUCTION($\Omega^{(0)}$)
    $\mathbb{P} \leftarrow ()$
    add $\Omega^{(0)}$ to $\mathbb{P}$
    $i \leftarrow 0$
    $S \leftarrow \{e : \text{Tail}(e) \in \mathcal{S}\}$
    while $T_1^{(i)} \cup T_2^{(i)} \nsubseteq S$ do
        $E \leftarrow \left\{e : \arg\max_{e \in T_1^{(i)} \cup T_2^{(i)}} \text{Ord}(e)\right\}$
        $E_j \leftarrow E \cap T_j^{(i)}, j = 1, 2$
        $v \leftarrow \text{Tail}(E)$
        for $j \leftarrow 1, 2$ do
            if $E_j \neq \varnothing$ then
                $T_j^{(i+1)} \leftarrow \left(T_j^{(i)} \backslash E_j\right) \cup \text{In}(v)$
            else
                $T_j^{(i+1)} \leftarrow T_j^{(i)}$
            end if
        end for
        $\mathcal{T}^{(i+1)} \leftarrow \left\{T_1^{(i+1)}, T_2^{(i+1)}\right\}$
        $\Omega^{(i+1)} \leftarrow (\mathcal{G}, \mathcal{S}, \mathcal{T}^{(i+1)})$
        add $\Omega^{(i+1)}$ into $\mathbb{P}$
        $i \leftarrow i + 1$
    end while
    return $\mathbb{P}$
end procedure
```

denote the set of edges in $T_1^{(i)} \cup T_2^{(i)}$ with the highest topological order. In other words, all edges in $E \subset T_1^{(i)} \cup T_2^{(i)}$ have the same topological order, and a strictly higher topological order with respect to every edge in $T_1^{(i)} \cup T_2^{(i)} \backslash E$. For $j \in \{1, 2\}$, let $E_j = T_j^{(i)} \cap E$. For each destination $j \in \{1, 2\}$, if $T_j^{(i)}$ does not contain any highest topological ordered edge, i.e., if $E_j = \varnothing$, then the destination set remains unchanged in $\Omega^{(i+1)}$, i.e., $T_j^{(i+1)} = T_j^{(i)}$. Otherwise, all edges in E_j are removed in $T_j^{(i)}$ and replaced by $\text{In}(v)$ to produce the new destination set $T_j^{(i+1)}$ in $\mathcal{T}^{(i+1)}$, that is, $T_j^{(i+1)} = (T_j^{(i)} \backslash E_j) \cup \text{In}(v)$.

Before proceeding to describing our coding scheme, we list some useful and instructive properties of the destination reduction algorithm; these properties can be easily checked for the example in Fig. 6.

Property (i) The set of edges $T_j^{(i)} \backslash T_j^{(i+1)}$ has a common tail node v, which also forms the common head node of all the edges in $T_j^{(i+1)} \backslash T_j^{(i)}$. Furthermore, there are only two possibilities: $T_j^{(i+1)} \backslash T_j^{(i)}$ is empty or $T_j^{(i+1)} \backslash T_j^{(i)} = \text{In}(v)$.

Property (ii) An edge in the graph which communicates to at least one edge in T_1 appears in $T_1^{(i)}$

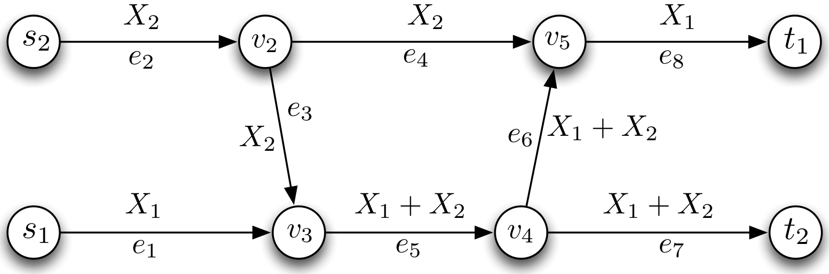
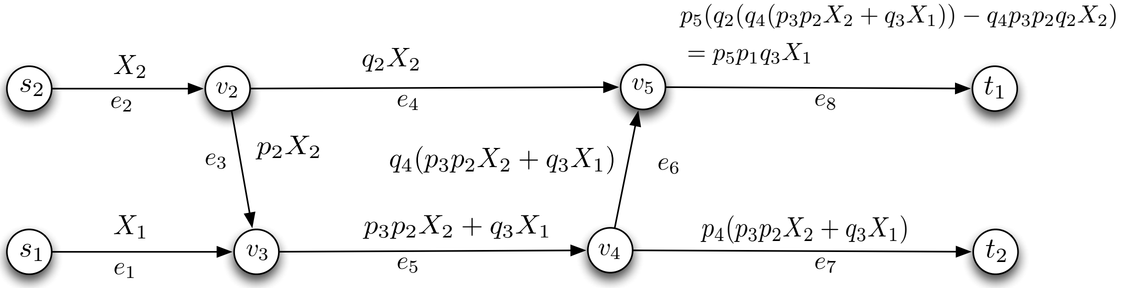


Figure 5: An example network used to demonstrate our algorithm operation



	$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$
Destination $T_1^{(i)}$	e_8	e_4, e_6	e_4, e_5	e_1, e_3, e_4	e_1, e_2
Destination $T_2^{(i)}$	e_7	e_7	e_5	e_1, e_3	e_1, e_2

Figure 6: The destination and recursive coding algorithms shown for the network of Fig. 5. The scalars $p_2, q_2, p_3, p_4, q_4, p_5$ are chosen randomly and independently of each other. Note that the recursive coding algorithm operating at Stage 0 performs alignment step at vertex v_5.

for some value $i \in \{0, 1, 2, \ldots, N\}$. An edge which communicates to at least one edge in T_2 appears in $T_2^{(i)}$ for some value of i. Consider an edge e which communicates to at least one edge in T_1 and at least one edge in T_2. Let k_1 denote the largest number such that the edge e belongs to $T_1^{(k_1)}$. Let k_2 denote the largest number such that the edge e belongs to $T_2^{(k_2)}$. Then $k_1 = k_2$.

Property (iii) Consider a two-unicast-Z problem Ω, where every edge in the graph is connected to at least one of the destinations, that is, there is a path from every edge to at least one edge in $T_1 \cup T_2$. Then, the set of all edges have a lower topological order with respect to $\left(T_1^{(i)} \cup T_2^{(i)}\right) \setminus \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$ in $\mathcal{G}$ is equal to

$$\bigcup_{i+1 \leq k \leq N} T_1^{(k)} \cup T_2^{(k)}.$$

Property (iv) In the $\Omega^{(N)}$, the destination edges are collocated with the source edges. That is $T_1^{(N)} \cup T_2^{(N)} \subseteq S_1 \cup S_2$. Furthermore, if every edge emanating from the source nodes communicates with at least one of the destination nodes, then $T_1^{(N)} \cup T_2^{(N)} = S_1 \cup S_2$.

Property (v) For all i, $0 \leq i \leq N$ and $j = 1, 2$, the destination set $T_j^{(i)}$ forms a cut set between both sources s_1, s_2 and the destination set T_j for the original problem Ω.

Properties (i) and (iv) can be verified by examining the algorithm. We prove Properties (ii), (iii) and (v) in Appendix A. It is instructive to note that Properties (ii) and (iii) imply that the collection of sets

$$\left\{ \left(T_1^{(i)} \cup T_2^{(i)}\right) \setminus \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right) : i = 0, 1, 2, \ldots, N \right\}$$

forms a partition of the set of edges of the graph. Furthermore, this partition is the same as the partition implied by the topological ordering on the edges, i.e., all edges of the same topological order belong to one unique member of this partition. As we observe next in our description of the recursive coding algorithm, the recursion at depth j designs the local coding co-efficients for the edges in the jth member of this partition.

4.3 Recursive coding construction

We describe the recursive coding algorithm formally in Algorithm 2. We present an informal description here. Without loss of generality, we only consider networks where every source edge communicates with at least one edge in $T_1 \cup T_2$. The first step of the recursive coding construction begins with a trivial coding scheme for $\Omega^{(N)}$. In particular, note that Property (iv) states that $T_1^{(N)} \cup T_2^{(N)}$ is equal to $S_1 \cup S_2$. We set the local coding vector for an edge in $T_1^{(N)} \cup T_2^{(N)}$ to be the vector with co-efficient 1 corresponding to the edge and 0 elsewhere. We assume that the local coding vectors for all the edges outside of $T_1^{(N)} \cup T_2^{(N)}$ to be indeterminate at this point; the local coding co-efficients for these edges will be determined using the recursive coding algorithm. Each step of the recursion is referred to as a *stage*. The recursive algorithm has N stages, where at stage i, the algorithm generates the code for $\Omega^{(i)}$ using the coding scheme for the previous stage for $\Omega^{(i+1)}$. In particular, the recursive coding algorithm accomplishes the following: Given a linear coding scheme for the $(i+1)$th stage, that is, for $\Omega^{(i+1)}$, the algorithm at the ith stage constructs a linear coding scheme for $\Omega^{(i)}$. Starting with the trivial coding scheme for $\Omega^{(N)}$, our algorithm recursively constructs coding schemes for the problems $\Omega^{(N-1)}, \Omega^{(N-2)}, \ldots, \Omega^{(1)}$, which eventually leads to a coding scheme for the original problem $\Omega = \Omega^{(0)}$. Next we focus on the coding algorithm at stage i assuming a linear coding scheme for $\Omega^{(i+1)}$ is given in the previous stage.

A solution to the problem $\Omega^{(i+1)}$ will describe local coding co-efficients for all the edges in $\bigcup_{i+1 \leq k \leq N} T_1^{(i+1)} \cup T_2^{(i+1)}$, and leave the co-efficients for the remaining edges to be indeterminate. Given a linear coding solution for the problem $\Omega^{(i+1)}$, the coding solution for the problem $\Omega^{(i)}$ inherits the linear coding co-efficients from the solution to $\Omega^{(i+1)}$ for all edges in $\bigcup_{i+1 \leq k \leq N} T_1^{(i+1)} \cup T_2^{(i+1)}$. To complete the description for a solution to $\Omega^{(i)}$, the algorithm specifies the local coding co-efficients for edges in $\left(T_1^{(i)} \cup T_2^{(i)}\right) \setminus \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$. Because of Properties (ii) and (iii) of the destination reduction algorithm, we note that specifying local coding co-efficients for edge in $\left(T_1^{(i)} \cup T_2^{(i)}\right) \setminus \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$ suffices to specify the global coding-coefficients for these edges as well, since all the edges of which have a lower topological order with respect to this set have been assigned coding co-efficients in the solution to $\Omega^{(i+1)}$.

We use the following notation in our description. For $j = 1, 2$, let

$$U_j^{(i)} = T_j^{(i)} \cap T_j^{(i+1)}, \qquad I_j^{(i)} = T_j^{(i+1)} \setminus U_j^{(i)}, \qquad O_j^{(i)} = T_j^{(i)} \setminus U_j^{(i)}. \tag{13}$$

Recall from Property (i) that all the edges in $T_j^{(i)} \backslash T_j^{(i+1)}$ have a common tail node v, which is also the head of all the edges in $T_j^{(i+1)} \backslash T_j^{(i)}$. The set $I_j^{(i)}$ is therefore contained in the set of *incoming* edges on to this node v. The set $O_j^{(i)}$ is contained in the *outgoing* edges from v. Based on Property (i), we observe that if $I_1^{(i)} \neq \phi, I_2^{(i)} \neq \phi$, then $I_1^{(i)} = I_2^{(i)} = \text{In}(v)$. The set $U_j^{(i)}$ is the set of *unchanged* destination edges between $T_j^{(i+1)}$ and $T_j^{(i)}$. We divide $O_2^{(i)}$ into two disjoint subsets, $A_2^{(i)}$ and $B_2^{(i)}$, such that $O_2^{(i)} = A_2^{(i)} \cup B_2^{(i)}$, where

$$A_2^{(i)} = O_2^{(i)} \cap O_1^{(i)}, \qquad\qquad B_2^{(i)} = O_2^{(i)} \backslash O_1^{(i)}. \tag{14}$$

Note that Property (ii) implies that $B_2^{(i)}$ contains edges that communicate with at least one edge in T_2 but not T_1. Similarly, $A_2^{(i)}$ contains edges that communicate with at least one edge T_1 and at least one edge in T_2. It is useful to note that $A_2^{(i)} \neq \phi \Rightarrow I_1^{(i)} \neq \phi, I_2 \neq \phi \Rightarrow I_1^{(i)} = I_2^{(i)}$. Since $B_2^{(i)}$ and $O_1^{(i)}$ form a partition of the set $\left(T_1^{(i)} \cup T_2^{(i)}\right) \backslash \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$, we specify local linear coding co-efficients for $B_2^{(i)}$ and $O_1^{(i)}$ in the recursive coding algorithm.

Henceforth we will use the following notation. For any set of edges $P \subseteq T_1^{(i)}$, $i \in \{1, 2, \}$, the transfer matrix between source i and P is denoted as $\mathbf{H}_i^P$. For any set of edges $Q \subseteq T_2^{(i)}$,$i \in \{1, 2\}$, the transfer matrix between source i and Q is denoted as $\mathbf{G}_i^Q$. Furthormore, for the sake of consistency, we will assume that the columns of $\mathbf{H}_i^P, \mathbf{G}_i^Q$ are ordered based on the topological orderings of the edges in P and Q. Note that with our notation, $\mathbf{H}_2^{T_1^{(i)} \cap T_2^{(i)}} = \mathbf{G}_2^{T_1^{(i)} \cap T_2^{(i)}}$. With this notation, the local coding vectors at node v generate $\mathbf{H}_j^{O_1^{(i)}}$ from $\mathbf{H}_j^{I_1^{(i)}}$, and $\mathbf{G}_j^{A_2^{(i)}}$ and $\mathbf{G}_j^{B_2^{(i)}}$ from $\mathbf{G}_j^{I_2^{(i)}}$, for $j = 1, 2$. Therefore, the goal of the recursive coding algorithm is to design local coding matrices $\mathbf{F}^{I_1^{(i)}, O_1^{(i)}}, \mathbf{F}^{I_2^{(i)}, B_2^{(i)}}$ at vertex v, with dimensions $|\text{In}(v)| \times |O_1^{(i)}|$, and $|\text{In}(v)| \times |B_2^{(i)}|$ respectively. The global coding co-efficients will be determined as, for $j \in \{1, 2\}$,

$$\mathbf{H}_j^{O_1^{(i)}} = \mathbf{H}_j^{I_1^{(i)}} \mathbf{F}^{I_1^{(i)}, O_1^{(i)}}, \mathbf{G}_j^{A_2^{(i)}} = \mathbf{G}_j^{I_2^{(i)}} \mathbf{F}^{I_2^{(i)}, A_2^{(i)}}, \mathbf{G}_j^{B_2^{(i)}} = \mathbf{G}_j^{I_2^{(i)}} \mathbf{F}^{I_2^{(i)}, B_2^{(i)}} \tag{15}$$

where note that, if $A_2^{(i)}$ is non-empty, then $\mathbf{F}^{I_2^{(i)}, A_2^{(i)}}$ is a sub-matrix of $\mathbf{F}^{I_1^{(i)}, O_1^{(i)}}$. This is because $A_2^{(i)} \subseteq O_1^{(i)}$ and, if $A_2^{(i)}$ is non-empty, $I_1^{(i)} = I_2^{(i)}$.

Informal Description of the Recursive Coding Algorithm: A formal description is described by Algorithm 2. Here, we present an informal description of the recursive coding algorithm for the ith stage, assuming that stage $i+1$ is complete. As previously stated, our goal is to generate $\mathbf{F}^{I_1^{(i)}, O_1^{(i)}}, \mathbf{F}^{I_2^{(i)}, B_2^{(i)}}$ in (15). We do this in two phases, in the first phase, we find $\mathbf{F}^{I_2^{(i)}, B_2^{(i)}}$ and in the second phase, we determine $\mathbf{F}^{I_1^{(i)}, O_1^{(i)}}$. Our strategy is based on the idea of designing the coding co-efficients so that the Grank is maximized at each stage.

4.3.1 Phase 1

If $B_2^{(i)} = \varnothing$, proceed to the next phase. Otherwise, select each entry of the matrix $\mathbf{F}^{I_2^{(i)}, B_2^{(i)}}$ uniformly at random from the underlying finite field $\mathbb{F}$. We note that, over a sufficiently large field,

Algorithm 2 Recursive Coding Algorithm

```
procedure RECURSIVE CODING($\mathbb{P}, \mathbb{F}$)    ▷ $\mathbb{F}$ represents the field over which the coding is performed
    Denote $\mathbb{P}$ by $(\Omega^{(i)}, \Omega^{(i+1)}, \dots, \Omega^{(N)})$.
    Denote $\Omega^{(i)} = (\mathcal{G}, \mathcal{S}, \{T_1^{(i)}, T_2^{(i)}\})$ and use the notation (13),(14).
    Denote $\mathcal{E}^{(k)} = \cup_{j=k}^{N} T_1^{(j)} \cup T_2^{(j)}$ for $k \in \{i, i+1, \dots, N-1\}$

    if length($\mathbb{P}$) $= 1$ then return $\mathbf{I}_{|\mathcal{E}^{(N)}|}$
    end if
    $\mathbf{F} = \mathbf{0}_{|\mathcal{E}^{(i)}| \times |\mathcal{E}^{(i)}|}$
    $\mathbf{F}^{\mathcal{E}^{(i+1)}, \mathcal{E}^{(i+1)}} =$ RECURSIVE CODING($(\Omega^{(i+1)}, \dots, \Omega^{(N)})$)    ▷ The recursion
                    ▷ In the next few steps, we will describe coding co-efficients $\mathbf{F}^{I_j^{(i)}, O_j^{(i)}}, j \in \{1, 2\}$

    if $B_2^{(i)} \neq \varnothing$ then    ▷ Phase 1
        $\mathbf{F}^{I_2^{(i)}, B_2^{(i)}} \leftarrow$ Uniformly random from the field $\mathbb{F}$
    end if
    $\overline{O} = \overline{A} = \phi$    ▷ Temporary variables (sets) used in the for loop next

    for $e \in O_1^{(i)}$ do    ▷ Phase 2: Encoding the edges in $O_1^{(i)}$ one edge at a time.
        $\overline{O} = \overline{O} \cup e, \overline{A} = \overline{O} \cap A_2^{(i)}$ ▷ $\overline{O}$ and $\overline{A}$ respectively represent the subsets of $O_1^{(i)}$ and $A_2^{(i)}$. In the next few steps, we will find the coding co-efficients for the edges in $\overline{O}, \overline{A}$

        if Grank$\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) >$ Grank$\left(\left[\mathbf{H}_1^{U_1^{(i)}} \ \mathbf{H}_1^{\overline{O}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \ \mathbf{H}_2^{\overline{O}}\right], \left[\mathbf{G}_2^{U_2^{(i)}} \ \mathbf{G}_2^{\overline{A}} \ \mathbf{G}_2^{B_2^{(i)}}\right]\right)$

            & colspan$\left(\mathbf{H}_2^{I_1^{(i)}}\right) \not\subset$ colspan$\left(\left[\mathbf{H}_2^{U_1^{(i)}} \ \mathbf{H}_2^{\overline{O}}\right]\right)$
            & colspan$\left(\mathbf{H}_2^{I_1^{(i)}}\right) \subset$ colspan$\left(\left[\mathbf{H}_2^{U_1^{(i)}} \ \mathbf{H}_2^{\overline{O}} \ \mathbf{G}_2^{U_2^{(i)}} \ \mathbf{G}_2^{\overline{A}} \ \mathbf{G}_2^{B_2^{(i)}}\right]\right)$.
        then    ▷ Alignment Step
            $\mathbf{v} \leftarrow$ Random vector in ker($\mathbf{H}_2^{T_1^{(i+1)}}$)
            $\mathbf{F}^{I_1^{(i)}, \{e\}} \leftarrow$ Last $|I_1^{(i)}|$rows of $\mathbf{v}$
        else
            $\mathbf{F}^{I_1^{(i)}, \{e\}} \leftarrow$ Uniformly at random from the field $\mathbb{F}$    ▷ Randomization Step
        end if
    end for

    return $\mathbf{F}$
end procedure
```

our choice of $\mathbf{F}^{I_2^{(i)},B_2^{(i)}}$ maximizes

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \begin{bmatrix}\mathbf{G}_2 & \mathbf{G}_2^{I_2^{(i)}}\mathbf{F}^{I_2^{(i)},B_2^{(i)}}\end{bmatrix}\right)$$

with high probability. This is because choosing the entries of $\mathbf{F}^{I_2^{(i)},B_2^{(i)}}$ uniformly at random maximizes, the rank of $\begin{bmatrix}\mathbf{H}_2^{T_1^{(i+1)}} & \mathbf{G}_2 & \mathbf{G}_2^{I_2^{(i)}}\mathbf{F}^{I_2^{(i),B_2^{(i)}}}\end{bmatrix}$ with a probability that tends to 1 as $|\mathbb{F}|$ increases. After phase 1, we have found coding vectors for edges $B_2^{(i)}$.

4.3.2 Phase 2

Let $O_1^{(i)} = \{e_{i,1}, e_{i,2}, \ldots, e_{i,m}\}$, where $m = |O_1^{(i)}|$. We design the coding co-efficients for the m edges in $O_1^{(i)}$ one-by-one, with the co-efficients designed to maximize the Grank at each step. Our choice of coding co-efficients is motivated by Lemma 2. In particular, the second phase is divided into q steps, where in the jth step, we design the coding co-efficients for edge $e_{i,j}$. Each step is classified as a *alignment step* or a *randomization step* as follows.

Let $O_{1,0}^{(i)} = \varnothing$ and $O_{1,j}^{(i)} = \{e_{i,1}, \ldots, e_{i,j}\}$ be the subset of the first j elements of $O_1^{(i)}$ for $1 \leq j \leq m$. Let $A_{2,j}^{(i)} = A_2^{(i)} \bigcap O_{1,j}^{(i)}$.

Case I (Alignment): The following conditions are satisfied.

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) > \text{Grank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_{1,j-1}^{(i)}}\end{bmatrix}, \begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}}\end{bmatrix}, \begin{bmatrix}\mathbf{G}_1^{U_2^{(i)}} & \mathbf{G}_1^{A_{2,j-1}^{(i)}} & \mathbf{G}_1^{B_2^{(i)}}\end{bmatrix}\right) \tag{16}$$

$$\text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) \not\subset \text{colspan}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}}\end{bmatrix}\right) \tag{17}$$

$$\text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) \subset \text{colspan}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} & \mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{A_{2,j-1}^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) . \tag{18}$$

In this case, we choose a vector $\mathbf{v}$ randomly in the nullspace of the column space spanned by $\mathbf{H}_2^{T_1^{(i+1)}}$ and choose the column vector $\mathbf{F}^{I_1^{(i)},\{e_{ij}\}}$ to be the last $|\mathbf{I}_1^{(i)}|$ rows of column vector $\mathbf{v}$. Note that the sets $O_{1,j}^{(i)}, A_2^{(i)}$ are denoted by temporary variables $\overline{O}, \overline{A}$ in Algorithm 2.

Case II (Randomization): Otherwise, that is, at least one of the conditions (16),(17),(18) is violated. In this case, we select the vector $\mathbf{F}^{I_1^{(i)},\{e_{ij}\}}$ by choosing each of its entries uniformly at random from $\mathbb{F}$.

If (16)-(18) are satisfied, then we refer to our coding step as the *alignment* step. If at least one of (16)-(18) is violated, we refer to the coding step as a *randomization* step. After the q steps of Phase 2, the coding co-efficients for

$$\bigcup_{i \leq k \leq N} T_1^{(i+1)} \cup T_2^{(i+1)},$$

are determined. This completes the description of the recursive coding algorithm. Before proceeding, we discuss some properties of our algorithm.

Discussion

We note that if (16) is satisfied, then, for a sufficiently large field size, we can show using Lemma 2 that

$$\begin{aligned}
&\text{Grank}\left(\left[\mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j}^{(i)}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j}^{(i)}}\right], \left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right)\\
&= \text{Grank}\left(\left[\mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j-1}^{(i)}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j-1}^{(i)}}\right], \left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j-1}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right) + 1
\end{aligned}$$

with a probability that approaches 1 if the size of the field $\mathbb{F}$ grows arbitrarily. We show this below

$$\begin{aligned}
&\text{Grank}\left(\left[\mathbf{H}_1^{T_1^{(i+1)}} \quad \mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j}^{(i)}}\right], \left[\mathbf{H}_2^{T_1^{(i+1)}} \quad \mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j}^{(i)}}\right], \left[\mathbf{G}_2^{T_2^{(i+1)}} \quad \mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j-1}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right)\\
&\overset{(a)}{=} \text{Grank}\left(\left[\mathbf{H}_1^{T_1^{(i+1)}}\right], \left[\mathbf{H}_2^{T_1^{(i+1)}}\right], \left[\mathbf{G}_2^{T_2^{(i+1)}}\right]\right)\\
&\overset{(b)}{>} \text{Grank}\left(\left[\mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j-1}^{(i)}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j-1}^{(i)}}\right], \left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j-1}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right)
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow \text{Grank}\left(\left[\mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j}^{(i)}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j}^{(i)}}\right], \left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right)\\
&= \text{Grank}\Bigg(\left[\mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j-1}^{(i)}} \quad \mathbf{H}_1^{T_1^{(i+1)}} \mathbf{F}^{T_1^{(i+1)},\{e_{i,j}\}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j-1}^{(i)}} \quad \mathbf{H}_2^{T_1^{(i+1)}} \mathbf{F}^{T_1^{(i+1)},\{e_{i,j}\}}\right]\\
&\qquad \left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j-1}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}} \quad \mathbf{G}_2^{T_2^{(i+1)}} \mathbf{F}^{T_1^{(i+1)},\{e_{i,j}\}}\right]\Bigg)\\
&\overset{(c)}{=} \text{Grank}\left(\left[\mathbf{H}_1^{U_1^{(i)}} \quad \mathbf{H}_1^{O_{1,j-1}^{(i)}}\right], \left[\mathbf{H}_2^{U_1^{(i)}} \quad \mathbf{H}_2^{O_{1,j-1}^{(i)}}\right], \left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_{2,j-1}^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right) + 1
\end{aligned}$$

where (a) follows from Statement (iii) of Lemma 1, (b) follows from (16) and (c) follows from Lemma 2.

It is also instructive to note that, if $p > q$,

$$\text{Grank}(\mathbf{H}_1^{T_1^{(p)}}, \mathbf{H}_2^{T_1^{(p)}}, \mathbf{G}_2^{T_2^{(p)}}) \geq \text{Grank}(\mathbf{H}_1^{T_1^{(q)}}, \mathbf{H}_2^{T_1^{(q)}}, \mathbf{G}_2^{T_2^{(q)}}).$$

That is, the Grank of a stage q is no smaller than stage p which is downstream with respect to a stage q.

We note that our algorithm performs better than optimal routing (multi-commodity flow) for certain networks, for e.g., Fig. 5. On the other hand, for certain networks our simulations have revealed that routing outperforms our algorithm.

Complexity

First consider the destination reduction algorithm. Since each iteration corresponds to one vertex, i.e., the common tail node of the last topologically ordered edges, the algorithm terminates in $\mathcal{O}(|\mathcal{V}|)$ steps.

Now consider the recursive coding algorithm. The algorithm traverses through all the intermediate nodes in their topological order and performs either random coding or alignment step for outgoing edges. At each node v, the complexity of the coding operations is dominated by the complexity of the alignment step, if it is performed, which is bounded by $\mathcal{O}(d^3)$, where d is the in-degree of node v. Therefore, we have the following property for the recursive coding algorithm.

Lemma 3. *For the class of directed acyclic graphs with in-degree bounded by D, the complexity of the recursive coding algorithm is $\mathcal{O}(|\mathcal{V}|D^3)$.*

5 Alternative proof of the max-flow min-cut theorem

In this section, we give an alternate proof of the max-flow min-cut theorem. The intuition of our proof is provided in Section 2. Our proof is based on induction on the number of edges of the graph. We begin with some notation that we will use in this section.

We denote the single unicast network graph as $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with n edges the subscript n is a notation that is useful in our proof. As per the notation introduced in Section 3, we denote the unicast problem as $\Omega_n = (\mathcal{G}_n, s, T_n)$, where $s \in \mathcal{V}_n$ is the source and $T_n \subset \mathcal{E}_n$ is the set of destination edges. A linear coding solution for Ω_n can be described by the local coding matrix $\mathbf{F}_n$, which results in a linear network transfer $\mathbf{M}_n = (\mathbf{I}_n - \mathbf{F}_n)^{-1}$. The source to destination linear transfer matrix $\mathbf{H}_n$ can be obtained as the submatrix of $\mathbf{M}_n$, whose rows correspond to the source edges and whose columns correspond to the destination edges. Mathematically, it can be done by multiplying $\mathbf{M}_n$ with with an incident matrix $\mathbf{A}_n$ of size $S \times n$ and then an exit matrix $\mathbf{B}_n$ of size $D \times n$, where S is out-degree of the source node and D is the cardinality of the destination edge set. Each row of $\mathbf{A}_n$ is a length n unit vector indicating the corresponding source edge coming out of the source. Likewise, each row $\mathbf{B}_n$ is a length n unit vector indicating the index of the corresponding destination edge[5]. To the end, the $(s - T_n)$ source to destination linear transfer matrix is given by $\mathbf{H}_n = \mathbf{A}_n (\mathbf{I}_n - \mathbf{F}_n)^{-1} \mathbf{B}_n^T$. Our goal is to show that the rank of $\mathbf{H}_n$ can be made equal to the min-cut between the source and the destination by choosing $\mathbf{F}_n$ appropriately. We use the following notation for the min-cut: for any problem $\Omega = (\mathcal{G}, s, T)$, we denote the min-cut between the source node s and the destination edges T as $c_{\mathcal{G}}(s, T)$.

Without loss of generality, we assume $\mathbf{A}_n \cdot \mathbf{B}_n = \mathbf{0}$, that is, the source node s is not a tail node of any destination edges in T in the graph $\mathcal{G}_n$. If there is any destination edge that is coming directly from the source node, then this edge can be removed; it suffices to show the max-flow min-cut theorem can be proved on the remaining graph. Next we introduce a few definitions and lemmas which will be useful in our proof.

5.1 Preliminary Lemmas

Definition 2 (Atomic matrix)**.** *An atomic matrix of size $n \times n$ is an upper-triangular matrix, where all the off-diagonal elements are zero, except those elements in a single column. Given an upper-triangular matrix $\mathbf{U}$, the i-th atomic matrix of $\mathbf{U}$, denoted as $\mathbf{U}^{[i]}$, is the atomic matrix formed by setting all the off-diagonal elements of $\mathbf{U}$ to zero, except those in column i.*

[5] Note the difference the incident/exit matrices and the input/output matrices $\mathbf{A}$ and $\mathbf{B}$ defined in [2]. The latter are not restricted to unit vector in rows and encompass the encoding and decoding operations at the source and receiver respectively. Here, we focus on the transfer matrix observed for the network. Hence, we are concerned with only incident and exit matrices.

A standard property in matrix algebra captures the relation between an upper-triangular matrix and its atomic matrices.

Lemma 4 (Atomic decomposition)**.** *An $n \times n$ triangular matrix $\mathbf{U}$ with all one diagonal elements can be written as the product of its atomic matrices in the reverse index order, i.e.* $\mathbf{U} = \mathbf{U}^{[n]} \cdot \mathbf{U}^{[n-1]} \cdots \mathbf{U}^{[1]}$.

Since the local coding matrix $\mathbf{F}_n$ is an $n \times n$ strict upper-triangular matrix, the quantity $(\mathbf{I}_n - \mathbf{F}_n)$ is also an upper-triangular matrix but with all diagonal elements begin equal to 1. We are interested in decomposing the inverse of atomic matrix $(\mathbf{I}_n - \mathbf{F}_n)^{[i]}$ in order to understand the linear network transfer matrix. For that, we start with the following property.

Property 1.

$$\left((\mathbf{I}_n - \mathbf{F}_n)^{[i]}\right)^{-1} = (\mathbf{I}_n + \mathbf{F}_n)^{[i]} \ . \tag{19}$$

For simplicity, we denote $\mathbf{E}_n = (\mathbf{I}_n + \mathbf{F}_n)$ and define i-th atomic matrix $\mathbf{E}_n^{[i]}$ to be the i-th *edge coding matrix.* Note that the i-th column entries of $\mathbf{E}_n^{[i]}$ above the diagonal represent exactly the local coding vector of the i-th edge of the network. With property (19), we note the following.

Lemma 5 (Network transfer matrix decomposition)**.** *The network transfer matrix* $(\mathbf{I}_n - \mathbf{F}_n)^{-1}$ *can be decomposed into the product of its edge matrices in the forward topological order, i.e.* $(\mathbf{I}_n - \mathbf{F}_n)^{-1} = \prod_{i=1}^{n} \mathbf{E}_n^{[i]}$

The proof is simple and omitted. The above lemma is interesting because it decomposes the network transfer matrix into the local coding based into a prodoct of n matrices, where each matrix in the product captures the influence of the local coding vector corresponding to a single edge. The edge reduction lemma of [45] can be shown simply using the above decomposition. We now proceed to a proof of the max flow min-cut theorem.

5.2 The proof

Now we are ready to prove the max flow min cut theorem. It is sufficient to show the linear achievability of a flow that equals to the min cut. That is equivalent to the following proposition.

Proposition 1 (Achievability of Min-Cut)**.** *For all $k \in \mathbb{Z}^+$, given an arbitrary directed acyclic graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E})$ with k edges, and a unicast problem $\Omega_k = (\mathcal{G}_k, s, T)$, there exists a $k \times k$ local coding matrix $\mathbf{F}_k$, such that the rank of the transfer matrix from s to T equals to the min cut between s and T, i.e.*

$$\textit{rank}\,(\mathbf{H}) = \textit{rank}\left(\mathbf{A}\,(\mathbf{I}_k - \mathbf{F}_k)^{-1}\,\mathbf{B}^T\right) = c_{\mathcal{G}_k}(s, T) \ . \tag{20}$$

Note that this is equivalent of showing that there exists an edge coding matrices $\mathbf{E}_k^{[i]}$ for the i-th edge, such that

$$\operatorname{rank}(\mathbf{H}) = \operatorname{rank}\left(\mathbf{A}\left(\mathbf{I}_k - \mathbf{F}_k^{-1}\right)\mathbf{B}^T\right) = \operatorname{rank}\left(\mathbf{A} \cdot \prod_{i=1}^{n} \mathbf{E}_k^{[i]} \cdot \mathbf{B}^T\right) = c_{\mathcal{G}_k}(s, T) \ . \tag{21}$$

We prove Proposition 1 using mathematical induction on $k \in \mathbb{Z}^+$. When $k = 1$, the claim is trivially true. For the inductive step, we start with the following inductive assumption and claim

Assumption 1 (Inductive assumption)**.** *Proposition 1 holds for all k such that* $1 \leq k \leq n-1$.

Using the above assumption, we prove Proposition 1 for the case where $k = n$, for an arbitrary, unicast problem Ω_n. It will be convenient to denote $\mathcal{E}_n = \mathcal{E}, T_n = T, \mathbf{H}_n = \mathbf{H}, \mathbf{A}_n = \mathbf{A}, \mathbf{B}_n = \mathbf{B}$. Let e_n be an edge with the highest topological order in graph $\mathcal{G}_n$. Let $\mathcal{G}_{n-1} = (\mathcal{V}, \mathcal{E} - \{e_n\})$. In the problem Ω_n, if e_n is not a destination edge, i.e. $e_n \notin T_n$, then e_n does not communicates with any destination edges, i.e. $e_n \not\rightsquigarrow e$, $\forall e \in T$, since e_n has the highest topological order in $\mathcal{G}_n$. Consequently, erasing e_n do not affect the capacity and the min cut from s to T_n, i.e. $c_{\mathcal{G}_n}(s, T_n) = c_{\mathcal{G}_{n-1}}(s, T_n)$.

Next we focus on the case when $e_n \in T_n$. Note that e_n is not emitted directly from the source node s and is a destination edge. As a result, we can decompose the incident and exit matrices as follows,

$$\mathbf{A}_n = \begin{bmatrix} \mathbf{A}_{n-1} & \mathbf{0} \end{bmatrix}, \qquad\qquad \mathbf{B}_n = \begin{bmatrix} \mathbf{B}^*_{n-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{22}$$

where $\mathbf{A}_{n-1}$ is the incident matrix from the source to subgraph $\mathcal{G}_{n-1}$, while $\mathbf{B}^*_{n-1}$ is the exit matrix from $\mathcal{G}_{n-1}$ to the first $D-1$ destination edges. The zero column following $\mathbf{A}_{n-1}$ indicates that e_n is not emitted from the source node, whereas the unit vector in the last column of $\mathbf{B}_n$ indicates that e_n is a destination edge. Subsequently, consider the decomposition of the $s - T_n$ transfer matrix on $\mathcal{G}_n$. In the following sequence of equations, we use the notation $\mathbf{e}_n$ to denote the first $n-1$ entries of the nth column of matrix $\mathbf{F}_n$; note that the last entry of the nth column is 0. We can write the transfer matrix as,

$$\begin{align}
\mathbf{H}_n &= \mathbf{A}_n \left(\mathbf{I}_n - \mathbf{F}_n\right)^{-1} \mathbf{B}_n^T = \mathbf{A}_n \cdot \prod_{i=1}^{n} \mathbf{E}_n^{[i]} \cdot \mathbf{B}_n^T \notag \\
&= \begin{bmatrix} \mathbf{A}_{n-1} & \mathbf{0} \end{bmatrix} \cdot \prod_{i=1}^{n} \mathbf{E}_i^{[i]} \cdot \begin{bmatrix} \mathbf{B}^*_{n-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^T \tag{23} \\
&= \begin{bmatrix} \mathbf{A}_{n-1} & \mathbf{0} \end{bmatrix} \cdot \left(\prod_{i=1}^{n-1} \mathbf{E}_n^{[i]} \right) \cdot \mathbf{E}_n^{[n]} \cdot \begin{bmatrix} \mathbf{B}^*_{n-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^T \notag \\
&\overset{(a)}{=} \begin{bmatrix} \mathbf{A}_{n-1} & \mathbf{0} \end{bmatrix} \cdot \left[\begin{array}{c|c} \left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1} & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array} \right] \cdot \mathbf{E}_n^{[n]} \cdot \begin{bmatrix} \mathbf{B}^*_{n-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^T \notag \\
&\overset{(b)}{=} \begin{bmatrix} \mathbf{A}_{n-1} & \mathbf{0} \end{bmatrix} \cdot \left[\begin{array}{c|c} \left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1} & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array} \right] \cdot \left[\begin{array}{c|c} \mathbf{I}_{n-1} & \mathbf{e}_n \\ \hline \mathbf{0} & 1 \end{array} \right] \cdot \begin{bmatrix} \mathbf{B}^*_{n-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^T \notag \\
&= \begin{bmatrix} \mathbf{A}_{n-1} \left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{B}^{*T}_{n-1} & \mathbf{e}_n \\ \mathbf{0} & 1 \end{bmatrix} \notag \\
&= \mathbf{A}_{n-1} \left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1} \begin{bmatrix} \mathbf{B}^{*T}_{n-1} & \mathbf{e}_n \end{bmatrix} . \tag{24}
\end{align}$$

In this process, for step (a), we apply Lemma 5 on the product of first $n-1$ to obtain the network transfer matrix of the subgraph $\mathcal{G}_{n-1}$, while in (b) we simply write out the matrix $\mathbf{E}_n$ explicitly using the local coding vector $\mathbf{e}_n$ on the edge e_n and idenity matrix of $(n-1) \times (n-1)$. To the end, we decompose the source-destination transfer matrix into two parts that can be intuitively understood. The first part, $\mathbf{H}_{n-1} = \mathbf{A}_n \left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1} \mathbf{B}^{*T}_{n-1}$ is simply the source to destination transfer matrix of the subgraph $\mathcal{G}_{n-1}$. The second part is the contribution of the last topologically ordered edge e_n. Therefore,

$$\mathbf{H}_n = \begin{bmatrix}\mathbf{H}_{n-1} & \mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1}\mathbf{e}_n\end{bmatrix}. \tag{25}$$

Now consider the last edge e_n. If it is not a part of any $s - T_n$ min cut on graph $\mathcal{G}_n$, then removal of e_n does not affect the min cut of the graph, i.e. $c_{\mathcal{G}_{n-1}}(s,t) = c_{\mathcal{G}_n}(s,t)$. In this case, leveraging the inductive assumption on the subgraph $\mathcal{G}_{n-1}$, we claim that there exists local coding matrix $\mathbf{F}^*_{n-1}$ for the graph $\mathcal{G}_{n-1}$ such that,

$$\text{rank}\left(\mathbf{H}_{n-1}\right) = \text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1}\mathbf{B}^{*T}_{n-1}\right) = c_{\mathcal{G}_{n-1}}(s,t) = c_{\mathcal{G}_n}(s,t)\ . \tag{26}$$

To put it simply, when e_n is not a part of any min cut set, we can ignore the this edge and achieve a flow equal to the min cut, using only the remaining subgraph $\mathcal{G}_{n-1}$. To do this, we can simply choose the existing local coding matrix $\mathbf{F}^*_{n-1}$ which satisfies (26) and set the local coding vector at $\mathbf{e}_n$ to be zero, i.e. $\mathbf{e}_n = \mathbf{0}$. Doing this eliminates the last column in (25) and guarantees that

$$\text{rank}\left(\mathbf{H}_n\right) = \text{rank}\left(\mathbf{H}_{n-1}\right) = c_{\mathcal{G}_{n-1}}(s,T_n) = c_{\mathcal{G}_n}(s,T_n) \tag{27}$$

It remains to prove the claim in the case when e_n belongs to some min-cut set for the graph $\mathcal{G}_n$. Let $v = \text{Tail}(e_n)$ be the tail node of theg edge e_n. We examine two unicast problems on $\mathcal{G}_{n-1}$, $\Omega_{n-1} = (\mathcal{G}_{n-1}, s, T_{n-1})$ and $\Omega^*_{n-1} = \left(\mathcal{G}_{n-1}, s, T^*_{n-1}\right)$, where T_{n-1} and T^*_{n-1} are given by,

$$T_{n-1} = T_n \backslash \{e_n\} \cup \text{In}(v) \quad , \qquad\qquad T^*_{n-1} = T_n \backslash \{e_n\} \tag{28}$$

Note that both Ω_{n-1} and Ω^*_{n-1} are unicast problems whose underlying graph has exactly $n-1$ edges. For the source-destination min cuts, we have the following lemma.

Lemma 6. *When e_n is a edge in some min cut between s and T_n on $\mathcal{G}_n$, the min cut values on $\mathcal{G}_n$ and $\mathcal{G}_{n-1}$ satisfy the following*

$$c_{\mathcal{G}_{n-1}}(s, T_{n-1}) \geq c_{\mathcal{G}_n}(s, T_n) \tag{29}$$

$$c_{\mathcal{G}_{n-1}}(s, T^*_{n-1}) = c_{\mathcal{G}_n}(s, T_n) - 1 \tag{30}$$

The proof follows from elementary graph theoretic arguments and is omitted. It is straightforward to see that the $s - T_{n-1}$ transfer matrix for Ω^*_{n-1} is exactly given by $\mathbf{H}_{n-1}$, which is the first part of the matrix $\mathbf{H}_n$. For Ω_{n-1}, the destination edge set is formed by replacing e_n with its parens edges, i.e. incoming edges to v. Let these edge be $e_{i_1}, \ldots, e_{i_h}$, where $i_1, \ldots, i_h$ are the edge indices, the exit matrix of Ω_{n-1} is given by

$$\mathbf{B}^T_{n-1} = \left[\ \mathbf{B}^{*T}_{n-1}\ \middle|\ \mathbf{u}_{i_1}\ \middle|\ \ldots\ \middle|\ \mathbf{u}_{i_h}\ \right]\ , \tag{31}$$

where $\mathbf{u}_{i_j},\ 1 \leq j \leq h$ is a length n unit vector with 1 at position i_j and zero elsewhere. Hence, the transfer matrix for Ω_{n-1} is,

$$\begin{aligned}\mathbf{H}_{n-1} &= \mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1}\mathbf{B}^T_{n-1} \\ &= \begin{bmatrix}\mathbf{H}^T_{n-1} & \mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1}\begin{bmatrix}\mathbf{u}_{i_1} & \ldots & \mathbf{u}_{i_h}\end{bmatrix}\end{bmatrix}.\end{aligned}$$

Next, we invoke the inductive assumptions on the problems Ω_{n-1} and Ω^*_{n-1}. We can conclude that, there exists a local coding matrix $\mathbf{F}_{n-1}$ for Ω_{n-1} and a local coding matrix $\mathbf{F}^*_{n-1}$ for Ω^*_{n-1}, such that,

$$\text{rank}\left(\mathbf{H}_{n-1}\right) = \text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1}\mathbf{B}^T_{n-1}\right) = c_{\mathcal{G}_{n-1}}(s, T_{n-1}) \geq c_{\mathcal{G}_n}(s, T_n) \tag{32}$$

$$\text{rank}\left(\mathbf{H}^*_{n-1}\right) = \text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}^*_{n-1}\right)^{-1}\mathbf{B}^{*T}_{n-1}\right) = c_{\mathcal{G}_{n-1}}(s, T^*_{n-1}) = c_{\mathcal{G}_n}(s, T_n) - 1 \tag{33}$$

Finally, the following lemma helps us to find the local coding matrix for the original graph $\mathcal{G}_n$

Lemma 7. *Given a local coding matrix $\mathbf{F}_{n-1}$ that satisfying* (32) *and a local coding matrix $\mathbf{F}^*_{n-1}$ satisfying* (33)*, for a sufficiently large field $\mathbb{F}$, there exists $p,\ q \in \mathbb{F}$ and a local coding vector $\mathbf{e}_n$ such that*

$$\textit{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \left(p\mathbf{F}_{n-1} + q\mathbf{F}^*_{n-1}\right)\right)^{-1}\begin{bmatrix}\mathbf{B}^{*T}_{n-1} & \mathbf{e}_n\end{bmatrix}\right) = c_{\mathcal{G}_n}(s, T_n) \tag{34}$$

The proof follows from the spirit of the algebraic framework of network coding introduced in [2] and is presented in Appendix B. Therefore, from the two local coding matrices $\mathbf{F}_n$ and $\mathbf{F}^*_{n-1}$, we can choose the local coding vectors for the last k edges randomly, and construct the local coding matrix $\mathbf{F}_n$ for $\mathcal{G}_n$ as follows,

$$\mathbf{F}_n = \begin{bmatrix} p\mathbf{F}_{n-1} + q\mathbf{F}^*_{n-1} & \mathbf{e}_n \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{35}$$

The new local coding matrix $\mathbf{F}_n$ will satisfy the original claim in (20) which provides an achievable flow that equals the min cut on any arbitray graph of n edges.

Remark 4. *Note that in the process of searching for the local coding matrix $\mathbf{F}_n$ which satisfies* (20) *and achieves the maximum flow, we essentially rely on recursive construction the local coding matrices throught the sequence $\{\mathbf{F}_{n-1}, \mathbf{F}_{n-2}, \ldots, \mathbf{F}_1\}$ using* (35)*. These local coding matrices in turn correspond to maximum flow achieving coding matrices for smaller problems we constructed as subproblems, i.e. $\{\Omega_{n-1}, \Omega_{n-2}, \ldots \Omega_1\}$. This sequence of problems would, in fact, result in running the destination reduction algorithm of Section 4 on the original problem Ω (with the second destination set equal to the null set).*

6 Achievability of Rate Pair $(1, 1)$

We prove for the special case that the rate pair $(1, 1)$ is always achievable through our algorithm unless there is a single edge GNS cut. We begin by stating our result formally.

Theorem 1. *Consider a two-unicast-Z problem $\Omega = (\mathcal{G}, \mathcal{S}, \mathcal{T})$ where, for $i \in \{1, 2\}$, source s_i is connected to at least one edge in destination T_i, and the GNS-cut set bound is at least* 2*. Then, the coding matrix* $\mathbf{F}$ *returned by* RECURSIVECODING (REDUCTION (Ω) , $\mathbb{F}$) *achieves the rate pair* $(1, 1)$ *with a probability that tends to* 1 *as the field size* $\mathbb{F}$ *increases.*

We first state the a few lemmas on the transfer matrices produced by the recursive coding algorithm. They are useful for the proof of Theorem 1.

Lemma 8. *Let $\mathbf{H}_1, \mathbf{H}_2$ and $\mathbf{G}_2$ be matrices of dimensions $P_1 \times Q_1$, $P_2 \times Q_1$ and $P_2 \times Q_1$ respecitvely. If $\mathbf{H}_1 \neq \mathbf{0}$ and $\mathbf{G}_2 \neq \mathbf{0}$, then Grank $(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2) = 1$ if and only if the following holds,*

$$\textit{rank}\left(\mathbf{H}_2\right) = \textit{rank}\left(\begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\end{bmatrix}\right) = \textit{rank}\left(\mathbf{G}_2\right) = \textit{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right) = 1.$$

Lemma 9. *In the recursive coding algorithm, for any $i \in \{0, 1, 2, \ldots, N\}$, the matrix* $\begin{bmatrix} \mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}} \end{bmatrix}$ *does not contain an all zeroes column with a probability that tends to 1 as the size of field $|\mathbb{F}|$ increases.*

Lemma 10. *Let $Q^{(i)} = T_2^{(i)} \backslash T_1^{(i)}$. Given that the recursive algorithm performs the alignment step from some stage $k+1$ to stage k, then, with a probability that tends to 1 as the field size $|\mathbb{F}|$ increases, $\mathbf{G}_2^{Q^{(i)}} \neq \mathbf{0}$ for every stage i in $\{0, 1, 2 \ldots, k\}$.*

It is straightforward to verify Lemma 8 from the definition of Grank. The proofs of Lemmas 9 and 10 are provided in Appendix E and D respectively.

Remark 5. *Henceforth, all the statements of the proof hold true in a probabilistic sense. That is, the statements are true with a probability that tend to 1 as the field size $|\mathbb{F}|$ tends to infinity. To avoid laborious notation and wording, we omit mentioning this explicitly in our proof.*

6.1 Proof Overview

We first provide a brief overview of the proof of Theorem 1 and summarize the basic ideas of the proof. Note that we only prove the necessity part of the theorem in this paper, as the sufficiency follows directly from the GNS outer bound [26]. In particular, it suffices to show that the recursive coding algorithm will generate source-to-destination transfer matrices which give an achievable rate region that contains the point $(1, 1)$. In the proof, we show that this is true for an arbitrary two-unicast-Z problem $\Omega = (\mathcal{G}, \mathcal{S}, \mathcal{T})$ with underlying graph $\mathcal{G}$. We shall use the achievable region given in Theorem 1 in [45], which is simplified to the following in the case of two-unicast-Z networks,

$$R_1 \leq \text{rank}\left(\mathbf{H}_1^{T_1^{(0)}}\right) \ , \tag{36}$$

$$R_2 \leq \text{rank}\left(\mathbf{G}_2^{T_2^{(0)}}\right) \ , \tag{37}$$

$$R_1 + R_2 \leq \text{Grank}\left(\mathbf{H}_1^{T_1^{(0)}}, \mathbf{H}_2^{T_1^{(0)}}, \mathbf{G}_2^{T_2^{(0)}}\right) . \tag{38}$$

In order to show that the above achievable region contains the point $(1, 1)$, we prove three claims on the transfer matrices, each corresponds to an upper bound in (36) to (38). Recall that the algorithm is divided into stages from N to 0. At stage i, the algorithm specifies coding co-efficients for the edges in the destination sets of $\Omega^{(i)}$, specifically, for edges in $T_1^{(i)} \cup T_2^{(i)} \backslash \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$, by using the coding co-efficients in the previous stage $i+1$. Also, recall that in the process of coding for stage i, for destination $T_j^{(i)}$, $U_j^{(i)}$ denotes the unchanged edges; $I_j^{(i)}$ denotes the incoming edges which are coded in stage $i+1$ and are removed from the destination set; $O_j^{(i)}$ denotes the outgoing edges which are to be coded from $I_j^{(i)}$ edges and enter the destination set. Furthermore, $O_2^{(i)}$ is the union of two disjoint set $B_2^{(i)}$ and $A_2^{(i)}$. The former set of edges are coded in phase 1 of the algorithm while the latter edges are coded in phase 2 as a subset of $O_1^{(i)}$. To prove Theorem 1, we show that

$$\text{rank}\left(\mathbf{H}_1^{T_1^{(i)}}\right) \geq 1 \ , \qquad \text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) \geq 1 \ , \qquad \text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) \geq 2 \ , \tag{39}$$

for every stage i, $0 \leq i \leq N$, unless there is a single edge GNS cut in the graph. Note that the above statement is trivially true for the initial stage, i.e. $i = N$. Therefore, to prove the theorem, we assume that (39) holds true for some stage $i+1$ and we show that the conditions remain true in stage i unless there is a single edge GNS cut set. We outline our proof steps below.

- In Claims 1 and 2, we show that the transfer matrices $\mathbf{H}_1^{T_1^{(i)}}$ and $\mathbf{G}_2^{T_2^{(i)}}$ are non-zero, assuming that $\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}$ are non-zero. We prove this claim by considering the the cases of randomization and alignment steps separately. Since in the previous stage, the matrices $\mathbf{H}_1^{T_1^{(i+1)}}$ and $\mathbf{G}_2^{T_2^{(i+1)}}$ are non-zero by assumption, and since random linear combination of a set of non-zero vector does not result in a zero vector, a randomization step preserves the desired non-zero property of the respective matrices at stage i. In case that the stage contains an alignment step, we make use of the fact that conditions (16)-(18) have to be fulfilled and the generated column has to satsify Lemma 2. We verify that these conditions imply that neither $\mathbf{G}_2^{T_2^{(i)}}$ nor $\mathbf{H}_1^{T_1^{(i)}}$ are set to 0.
- We show in Claim 3 that in stage i, the Grank of the transfer matrices is always lower bounded by 2 only if there does not exist a single edge whose removal disconnects $(s_1, T_1), (s_2, T_2)$ and (s_2, T_1). Specifically, we assume that the claim holds for stage $i+1$ and we show that it remains true for the next stage i. To do that, we first show that if any alignment step takes place between the two stages, the rank of the resulting matrix $\begin{bmatrix} \mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}} \end{bmatrix}$ is strictly greater than the rank of $\mathbf{H}_2^{T_1^{(i)}}$. This implies that the Grank is strictly larger than the rank of $\begin{bmatrix} \mathbf{H}_2^{(i)} & \mathbf{G}_2^{(i)} \end{bmatrix}$, which is at least 1. As a consequence, we show that the Grank is at least 2.

  Subsequently, if the algorithm results in Grank less than 2, it can only carry out randomization steps at that stage. In this case, we use Lemmas 8, 9 and 10 to show that if the Grank at stage i reduces to 1, then (a) the edges in $U_2^{(i)}$ cannot be communicated from source s_2 (b) the sets $U_1^{(i)}$ and $B_2^{(i)}$ are empty, and (c) there can be only one newly coded edge, common to both destination set, i.e. $\left|O_1^{(i)}\right| = \left|A_2^{(i)}\right| = 1$. $(a), (b)$, and (c) imply that $T_2^{(i)}$ contains only one edge that s_2 communicates with, while $T_1^{(i)}$ contains this particular edge as the only edge in the set. Consequently, applying Property (v) in Section 4, we know that $O_1^{(i)}$ is a cut set between source s_1, s_2 and T_1. Furthermore, because of (a) and the fact that $T_2^{(i)}$ is a cut set between source s_1, s_2 and T_2, we establish that $A_2^{(i)} = T_2^{(i)} \backslash U_2^{(i)}$ is a cut set between s_2 and T_2. As a result, we infer that the removal of the edge in $O_1^{(i)}$ simultaneously disconnects the source-destination pairs $(s_1, T_1), (s_2, T_2)$ and (s_2, T_1). In other words, we infer that the edge in $O_1^{(i)}$ forms a GNS-cut set to complete the proof.

We now formally present proofs of Claims 1,2 and 3 which combine to serve as a proof of Theorem 1.

6.2 Proof of Theorem 1

Claim 1. *For all* $0 \leq i \leq N$, $\mathbf{H}_1^{T_1^{(i)}} \neq \mathbf{0}$.

Proof. Consider first the case of $i = N$. Since s_1 communicates with t_1, there exists some source edge $(s_1, v_1) \in T_1^{(N)}$. Recall that in the trivial solution for $\Omega^{(N)}$, we simply set the column corresponding the each edge in $T_1^{(N)}$ to a unit vector indicating the source edge it represents. Since $(s_1, v_1) \in T_1^{(N)}$ is a source edge at s_1, it gives rise to a non-zero column in $\mathbf{H}_1^{T_1^{(N)}}$. Thus, the claim holds for stage N.

Suppose the claim of $\mathbf{H}_1^{T_1^{(i+1)}} \neq \mathbf{0}$ is true at some stage $i + 1$, consider the stage i. Note that only phase 2 of the algorithm affects $\mathbf{H}_1^{T_1^{(i)}}$. If $\mathbf{H}_1^{U_1^{(i)}} \neq \mathbf{0}$, since $U_1^{(i)} \subseteq T_1^{(i)}$, the claim holds for $\mathbf{H}_1^{T_1^{(i)}}$. Otherwise, we must have $\mathbf{H}_1^{I_1^{(i)}} \neq \mathbf{0}$. In this case, consider some step j in phase 2.

- If it is a randomization step, then $\mathbf{H}_1^{T_1^{(i)}} \neq \mathbf{0}$.
- If it is an alignment step, by Lemma 2, the newly generated column must satisfy,

$$\mathbf{H}_2^{I_1^{(i)}} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}} \in \text{colspan}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} \end{bmatrix}\right),$$

$$\begin{bmatrix} \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{I_1^{(i)}} \end{bmatrix} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}} \notin \text{colspan}\left(\begin{bmatrix} \mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_{1,j-1}^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} \end{bmatrix}\right).$$

  This is only true when $\mathbf{H}_1^{I_1^{(i)}} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}} \neq \mathbf{0}$. By the construction of the algorithm, $\mathbf{H}_1^{I_1^{(i)}} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}}$ is a column in $\mathbf{H}_1^{T_1^{(i)}}$.

Therefore, $\mathbf{H}_1^{T_1^{(i)}} \neq 0$. This completes the proof. □

Claim 2. *For all* $0 \leq i \leq N$, $\mathbf{G}_2^{T_2^{(i)}} \neq \mathbf{0}$.

Proof. For $i = N$, since s_2 communicates with t_2, there exists a source edge $(s_2, v_2) \in T_2^{(N)}$. Similar to the previous case, in the trivial solution for $\Omega^{(N)}$, we simply set the column corresponding the each edge in $T_2^{(N)}$ to a unit vector indicating the source edge it represents. Since $(s_2, v_2) \in T_2^{(N)}$ is a source edge at s_2, it gives rise to a non-zero column in $\mathbf{G}_2^{T_2^{(N)}}$. Thus, the claim holds for stage N.

Assume the claim is true for stage $i + 1$ and consider stage i. If $\mathbf{G}_2^{U_2^{(i)}} \neq \mathbf{0}$, then the claim holds for $\mathbf{G}_2^{T_2^{(i)}}$, as $U_2^{(i)} \subseteq T_2^{(i)}$. Otherwise, we must have $\mathbf{G}_2^{I_2^{(i)}} \neq \mathbf{0}$. In this case, if random linear coding is performed, either in phase 1 (for $B_2^{(i)}$ edges) or for an edge in $A_2^{(i)}$ in phase 2, then the resulting column is non-zero and hence $\mathbf{G}_2^{T_2^{(i)}} \neq \mathbf{0}$.

Therefore, we only need to consider the case where alignment occurs in some step j in phase 2. In this case, because of the conditions under which the algorithm performs alignment, we have

$$\text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) \not\subseteq \text{colspan}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} \end{bmatrix}\right)$$

$$\text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) \subseteq \text{colspan}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} & \mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{A_{2,j-1}^{(i)}} & \mathbf{G}_2^{B_2^{(i)}} \end{bmatrix}\right).$$

Since $A^{(i)}_{2,j-1} \subseteq O^{(i)}_{1,j-1}$ by the construction of the algorithm, $\mathbf{G}_2^{A^{(i)}_{2,j-1}}$ is a submatrix of $\mathbf{H}_2^{O^{(i)}_{1,j-1}}$. Subsequently, for the above alignment conditions to hold we must have

$$\left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right] \neq \mathbf{0}. \tag{40}$$

Since $\left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]$ is a submatrix of $\mathbf{G}_2^{T_2^{(i)}}$, the alignment step does not set $\mathbf{G}_2^{T_2^{(i)}}$ to $\mathbf{0}$ and the claim is true for all $i, 0 \leq i \leq N$. $\square$

From the two claims, we have established that for all i,

$$\text{rank}\left(\mathbf{H}_1^{T_1^{(i)}}\right) \geq 1 \ , \qquad \qquad \text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) \geq 1 \ . \tag{41}$$

Claim 3. *For all* $0 \leq i \leq N$*, Grank* $\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) \geq 2$ *if and only if the graph* $\mathcal{G}$ *does not contain a single edge GNS cut set.*

Proof. First note that Claim 1 implies that Grank $\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) \geq \text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) \geq 1$. Observe that (41), Grank $\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = 1$ if and only if the following hold:

$$\text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = \text{rank}\left(\left[\mathbf{H}_2^{T_1^{(i)}} \quad \mathbf{G}_2^{T_2^{(i)}}\right]\right) = \text{rank}\left(\mathbf{H}_2^{T_1^{(i)}}\right) = \text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}}\end{bmatrix}\right) = 1. \tag{42}$$

Next, we prove Claim 3. First we show that the claim is true for stage N.

Suppose that Grank $\left(\mathbf{H}_1^{T_1^{(N)}}, \mathbf{H}_2^{T_1^{(N)}}, \mathbf{G}_2^{T_2^{(N)}}\right) = 1$. By (42),

$$\text{rank}\left(\mathbf{G}_2^{T_2^{(N)}}\right) = 1 = \text{rank}\left(\mathbf{H}_2^{T_1^{(N)}}\right) \ , \qquad \text{colspan}\left(\mathbf{H}_2^{T_1^{(N)}}\right) = \text{colspan}\left(\mathbf{G}_2^{T_2^{(N)}}\right) \ .$$

By the construction of the algorithm, at stage N, we assign a different unit vector for each of the different source edge. Therefore, to satisfy the above equations, we must have $T_1^{(N)} = T_2^{(N)}$ and $\left|T_1^{(N)}\right| = 1$. In this case, removal the single edge in $T_1^{(N)}$ will disconnect $(s_1, T_1), (s_2, T_2)$ and (s_2, T_1) simultaneously. Therefore, the claim holds for stage N.

Suppose that the claim holds for stage $i+1$, i.e. Grank $\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) \geq 2$. We show the claim for stage i.

Alignment steps do not violate Claim 3.

We first show that if the recursive coding algorithm performs some alignment step in Phase 2 in stage i, then Grank $\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) \geq 2$ and therefore, the claim holds. Without loss of generality, assume that the alignment step is performed for some edge $e_{i,j} \in O_1^{(i)}$. Then, by Lemma 2, the generated column $\begin{bmatrix} \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{I_1^{(i)}} \end{bmatrix} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}}$ will satisfy,

$$\begin{aligned} \mathbf{H}_2^{I_1^{(i)}} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}} &\in \text{colspan}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} \end{bmatrix}\right) , \\ \begin{bmatrix} \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{I_1^{(i)}} \end{bmatrix} \mathbf{F}^{I_1^{(i)},\{e_{i,j}\}} &\notin \text{colspan}\left(\begin{bmatrix} \mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_{1,j-1}^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}} \end{bmatrix}\right) . \end{aligned}$$

As a result,

$$\text{rank}\left(\begin{bmatrix} \mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_{1,j}^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j}^{(i)}} \end{bmatrix}\right) > \text{rank}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j}^{(i)}} \end{bmatrix}\right) . \tag{43}$$

Therefore, at stage i we have,

$$\begin{aligned} &\text{rank}\left(\begin{bmatrix} \mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}} \end{bmatrix}\right) - \text{rank}\left(\mathbf{H}_2^{T_1^{(i)}}\right) \\ &= \text{rank}\left(\begin{bmatrix} \mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_{1,j}^{(i)}} & \mathbf{H}_1^{O_1^{(i)} \setminus O_{1,j}^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j}^{(i)}} & \mathbf{H}_2^{O_1^{(i)} \setminus O_{1,j}^{(i)}} \end{bmatrix}\right) - \text{rank}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j}^{(i)}} & \mathbf{H}_2^{O_1^{(i)} \setminus O_{1,j}^{(i)}} \end{bmatrix}\right) \\ &\geq \text{rank}\left(\begin{bmatrix} \mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_{1,j}^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j}^{(i)}} \end{bmatrix}\right) - \text{rank}\left(\begin{bmatrix} \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j}^{(i)}} \end{bmatrix}\right) > 0 . \end{aligned}$$

Hence $\text{rank}\left(\begin{bmatrix} \mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}} \end{bmatrix}\right) > \text{rank}\left(\mathbf{H}_2^{T_1^{(i)}}\right)$. Since (41) holds for all stages, by Lemma 8, we have

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) \geq 2.$$

That implies, if any alignment step takes place at stage i, then the claim holds for stage i.

Randomization steps do not violate Claim 3.

It remains to show the claim holds if all the coding steps performed at stage i are randomization steps. Consider the stage $i+1$ where the claim is assumed to be true. That is, we have

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) \geq 2.$$

We consider two possible cases:

(I) $\text{rank}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) \geq 2$, and

(II) $\text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) > \text{rank}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) = 1.$

In each case, we show that if

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = 1 \tag{44}$$

then, the following hold:

$$\mathbf{G}_2^{U_2^{(i)}} = \mathbf{0}\,, \qquad U_1^{(i)} = B_2^{(i)} = \varnothing\,, \qquad O_1^{(i)} = A_2^{(i)}\,, \qquad \left|O_1^{(i)}\right| = 1\,. \tag{45}$$

Once we show this, the claim can be proved as follows. Given that $U_1^{(i)}$ is empty at stage i, clearly $T_1^{(i)} = O_1^{(i)}$. Therefore, based on Property (v) of the algorithm, we infer that s_1 and s_2 communicate with T_1 only through the single edge in $O_1^{(i)}$. In other words, removing the edge in $O_1^{(i)}$ disconnects T_1 from the two sources. We argue here that (45) implies that removing the edge disconnects s_2 from T_2 as well. Since $B_2^{(i)}$ is empty, we have $T_2^{(i)} \backslash T_1^{(i)} = U_2^{(i)}$. Because $\mathbf{G}_2^{U_2^{(i)}} = \mathbf{0}$, we infer from Lemma 10 that there is no alignment step in any stage $k \geq i$. Therefore, the matrix $\mathbf{G}_2^{U_2^{(i)}}$ is generated through performing random linear coding at every edge that lies between the sources and the edges in $T_1^{(i)} \cup T_2^{(i)}$. The fact that $\mathbf{G}_2^{U_2^{(i)}}$ is a zero matrix in conjunction with the classical result of [2] which indicates that random linear network coding achieves a rank that is equal to the min-cut to any set of destination edges together imply that s_2 does not communicate with $U_2^{(i)}$. Based on Property (v) in Section 4, $T_2^{(i)}$ is a cut set between s_1, s_2 and T_2. But since s_2 does not communicate with $U_2^{(i)}$, we conclude that s_2 communicates with T_2 only through $T_2^{(i)} \backslash U_2^{(i)} = A_2^{(i)} = O_1^{(i)}$, i.e., s_2 communicates with T_2 through the single edge in $O_1^{(i)}$. Consequently, the removal of the single edge in $O_1^{(i)}$ disconnects s_2 from T_2 as well. Thus, $O_1^{(i)}$ is a single-edge GNS cut set.

To complete the proof we show that if (44) holds, then (45) also holds for each of the two cases.

Case (I): $\text{rank}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) = \text{rank}\left(\left[\mathbf{G}_2^{U_2^{(i)}} \;\; \mathbf{G}_2^{I_2^{(i)}}\right]\right) \geq 2$. Suppose that $\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = 1$. Clearly, in this case, $A_2^{(i)}$ and $B_2^{(i)}$ cannot be both empty; otherwise $\mathbf{G}_2^{T_2^{(i)}} = \mathbf{G}_2^{T_2^{(i+1)}}$ and $\text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = 2$, which contradicts (42). Now, consider the possible ranks of the matrix $\mathbf{G}_2^{U_2^{(i)}}$. From (42) we have, $\text{rank}\left(\mathbf{G}_2^{U_2^{(i)}}\right) \leq \text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = 1$. We first show that the rank of $\mathbf{G}_2^{U_2^{(i)}}$ is 0. If $\text{rank}\left(\mathbf{G}_2^{U_2^{(i)}}\right) = 1$, then $\text{colspan}\left(\mathbf{G}_2^{I_2^{(i)}}\right) \not\subseteq \text{colspan}\left(\mathbf{G}_2^{U_2^{(i)}}\right)$, as $\text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) \geq 2$. If $A_2^{(i)} \cup B_2^{(i)}$ is non-empty, then $\mathbf{G}_2^{A_2^{(i)}}$ and/or $\mathbf{G}_2^{B_2^{(i)}}$ will be created from random coding of columns of $\mathbf{G}_2^{I_2^{(i)}}$. As a result, we have,

$$\text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = \text{rank}\left(\left[\mathbf{G}_2^{U_2^{(i)}} \quad \mathbf{G}_2^{A_2^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]\right) > \text{rank}\left(\mathbf{G}_2^{U_2^{(i)}}\right) = 1,$$

which contradicts (42). But $A_2^{(i)}$ or $B_2^{(i)}$ cannot both be empty. Hence, we cannot have $\text{rank}\left(\mathbf{G}_2^{U_2^{(i)}}\right) = 1$. Therefore, $\text{rank}\left(\mathbf{G}_2^{U_2^{(i)}}\right) = 0$. Therefore, we have $\text{rank}\left(\mathbf{G}_2^{I_2^{(i)}}\right) \geq 2$. Since $I_2^{(i)}$ is non-empty, note that columns of $\left[\mathbf{H}_2^{O_1^{(i)}} \quad \mathbf{G}_2^{A_2^{(i)}} \quad \mathbf{G}_2^{B_2^{(i)}}\right]$ are generated by random coding from columns of $\mathbf{G}_2^{I_2^{(i)}}$. If the set

$O_1^{(i)} \cup A_2^{(i)} \cup B_2^{(i)} = O_1^{(i)} \cup B_2^{(i)}$ (since $A_2^{(i)} \subseteq O_1^{(i)}$) contains at least two elements, then

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{T_1^{(i)}} & \mathbf{G}_2^{T_2^{(i)}}\end{bmatrix}\right) \geq \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{O_1^{(i)}} & \mathbf{G}_2^{A_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) \geq 2\ ,$$

which contradicts (42). We have already shown that $A_2^{(i)} \cup B_2^{(i)}$ is non-empty. Thus, we conclude $O_1^{(i)} \cup B_2^{(i)} = 1$. Since $O_1^{(i)}$ and $B_2^{(i)}$ are disjoint, there are only two possiblities: $\left|O_1^{(i)}\right| = 1, A_2^{(i)} = O_1^{(i)}, B_2^{(i)} = 0$ or $O_1^{(i)} = 0, B_2^{(i)} = 1$. We show next that the latter is impossible, and show that the former condition implies that $U_1^{(i)}$ is empty.

1. $\left|B_2^{(i)}\right| = 1$, $\left|O_1^{(i)}\right| = \left|A_2^{(i)}\right| = 0$. From (42), we have $\operatorname{rank}\left(\mathbf{H}_2^{T_1^{(i)}}\right) = 1$. Since $\operatorname{rank}\left(\mathbf{G}_2^{I_2^{(i)}}\right) = 2$, $\operatorname{colspan}\left(\mathbf{G}_2^{I_2^{(i)}}\right) \not\subset \operatorname{colspan}\left(\mathbf{H}_2^{T_1^{(i)}}\right)$. As $\mathbf{G}_2^{B_2^{(i)}}$ is generated from $\mathbf{G}_2^{I_2^{(i)}}$ by random coding, we have $\operatorname{colspan}\left(\mathbf{G}_2^{B_2^{(i)}}\right) \not\subset \operatorname{colspan}\left(\mathbf{H}_2^{T_1^{(i)}}\right)$. Consequently,

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{T_1^{(i)}} & \mathbf{G}_2^{T_2^{(i)}}\end{bmatrix}\right) \geq \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{T_1^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) \geq 2\ ,$$

which contradicts (42).

2. $\left|B_2^{(i)}\right| = 0$, $\left|O_1^{(i)}\right| = \left|A_2^{(i)}\right| = 1$. Since we have already shown that $\operatorname{rank}\left(\mathbf{G}_2^{U_2^{(i)}}\right) = 0$, to show (45), it remains to prove that $U_1^{(i)}$ is empty. We do this by showing the column spaces of both $\mathbf{H}_2^{U_1^{(i)}}$ and $\mathbf{H}_1^{U_1^{(i)}}$ are the null space. By the construction of the algorithm, $I_1^{(i)} = I_2^{(i)}$; since $B_2^{(i)} = \phi$, we have $\mathbf{H}_2^{I_1^{(i)}} = \mathbf{G}_2^{I_2^{(i)}}$. From (42), we have,

$$\operatorname{rank}\left(\mathbf{H}_2^{U_1^{(i)}}\right) \leq \operatorname{rank}\left(\mathbf{H}_2^{T_1^{(i)}}\right) = 1.$$

Since $\mathbf{H}_2^{O_1^{(i)}}$ is generated by random coding from $\mathbf{G}_2^{I_2^{(i)}}$ whose rank is 2, it is clear that $\operatorname{colspan}\left(\mathbf{H}_2^{O_1^{(i)}}\right) \not\subset \operatorname{colspan}\left(\mathbf{H}_2^{U_1^{(i)}}\right)$. As a result, if $\mathbf{H}_2^{U_1^{(i)}} \neq \mathbf{0}$ then

$$\operatorname{rank}\left(\mathbf{H}_2^{T_1^{(i)}}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}\right) \geq 2\ ,$$

which contradicts (42). Hence $\mathbf{H}_2^{U_1^{(i)}} = \mathbf{0}$ and $\operatorname{rank}\left(\mathbf{H}_2^{O_1^{(i)}}\right) = 1$. Now, if $\mathbf{H}_1^{U_1^{(i)}} \neq \mathbf{0}$, we have,

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}\right) \geq \operatorname{rank}\left(\mathbf{H}_1^{U_1^{(i)}}\right) + \operatorname{rank}\left(\mathbf{H}_2^{O_1^{(i)}}\right) \geq 2\ ,$$

which also contradicts (42). Thus, $\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}}\end{bmatrix} = \mathbf{0}$. But by Lemma 9, $\begin{bmatrix}\mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}}\end{bmatrix}$ does not contain the all zeroes column. That implies that $U_1^{(i)} = \varnothing$ and subsequently, $T_1^{(i)} = O_1^{(i)}$. Therefore, we have shown (45) as required.

In summary, if $\text{rank}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) \geq 2$, then $\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = 1$ holds if and only if there exists a single edge GNS cut set in graph $\mathcal{G}$.

Case (II): $\text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) > \text{rank}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) = 1$. Again we start by assuming $\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = \text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = 1$ in stage i. We show that (45) is true in three steps.

(i) We first show that $O_1^{(i)}$ is non-empty, $\text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}}\end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \geq 2$ and that $\text{colspan}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = \text{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}\right)$.

(ii) Next we use the result of (i) to show that $U_1^{(i)}$ is empty and $\left|O_1^{(i)}\right| = 1$.

(iii) Finally we use the results of (i) and (ii) to show that $\mathbf{G}_2^{U_2^{(i)}} = \mathbf{0}$, $B_2^{(i)} = \phi$, and $A_1^{(i)} = O_1^{(i)}$.

We describe the three steps in detail next.

(i) We show that $O_1^{(i)}$ is non-empty, $\text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}}\end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \geq 2$ and that $\text{colspan}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = \text{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}\right)$.

Because $\text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = \text{rank}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) = 1$, and because $\mathbf{G}_2^{T_2^{(i)}}$ is a linear combination of columns of $\mathbf{G}_2^{T_2^{(i+1)}}$, the column space of $\mathbf{G}_2^{T_2^{(i+1)}}$ is the same as that of $\mathbf{G}_2^{T_2^{(i)}}$. Therefore, if $O_1^{(i)}$ is empty, we have

$$\text{rank}\left(\mathbf{H}_2^{T_1^{(i)}} \quad \mathbf{G}_2^{T_2^{(i)}}\right) = \text{rank}\left(\mathbf{H}_2^{T_1^{(i+1)}} \quad \mathbf{G}_2^{T_2^{(i+1)}}\right),$$

which implies the following:

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = \text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) \geq 2 \; .$$

This contradicts the assumption that the Grank at stage i is 1. Thus $O_1^{(i)}$ is non-empty.

We now show that $\text{colspan}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = \text{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}\right)$. From (42), we have,

$$\text{rank}\left(\begin{bmatrix}\mathbf{H}_2^{T_1^{(i)}} & \mathbf{G}_2^{T_2^{(i)}}\end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_1^{(i)}} & \mathbf{G}_2^{T_2^{(i)}}\end{bmatrix}\right) = 1 \; .$$

Thus $\text{colspan}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}\right) \subseteq \text{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}\right)$. However, $\mathbf{H}_2^{O_1^{(i)}}$ is generated by random coding from $\mathbf{H}_2^{I_1^{(i)}}$. Hence, we must have,

$$\text{colspan}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \subseteq \text{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = \text{colspan}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) \; . \tag{46}$$

As a result, $\operatorname{rank}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \leq 1$. Furthermore, the rank of $\mathbf{H}_2^{T_1^{(i+1)}}$ is lower bounded by the rank of $\mathbf{H}_2^{T_1^{(i)}}$, which is 1 by (42). Therefore,

$$\operatorname{rank}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = 1\,, \quad \operatorname{colspan}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = \operatorname{colspan}\left(\mathbf{G}_2^{T_2^{(i+1)}}\right) = \operatorname{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}\right). \tag{47}$$

We now show that $\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \geq 2$.

Equation (47) implies that

$$rank\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) = 1, \ \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{T_1^{(i+1)}} & \mathbf{G}_2^{T_2^{(i+1)}}\end{bmatrix}\right) = 1. \tag{48}$$

By assumpion, we have,

$$\begin{aligned}\operatorname{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) &= \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}}\end{bmatrix}\right) + \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_2^{T_1^{(i+1)}} & \mathbf{G}_2^{T_2^{(i+1)}}\end{bmatrix}\right) - \operatorname{rank}\left(\mathbf{H}_2^{T_1^{(i+1)}}\right) \\ &\geq 2\,. \end{aligned} \tag{49}$$

Combining (48) and (49) we get

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}}\end{bmatrix}\right) \geq 2 \tag{50}$$

We have showed (i) and we proceed to (ii).

(ii) We show that $U_1^{(i)}$ is empty, and $\left|O_1^{(i)}\right| = 1$.

Since $\operatorname{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = 1$ we have

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}}\end{bmatrix}\right) \leq \operatorname{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) = 1\,.$$

Since we have shown that

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \geq 2,$$

we must have

$$\operatorname{colspan}\left(\begin{bmatrix}\mathbf{H}_1^{I_1^{(i)}} \\ \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \not\subset \operatorname{colspan}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}}\end{bmatrix}\right) \ . \tag{51}$$

Since $\begin{bmatrix}\mathbf{H}_1^{O_1^{(i)}}\\ \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}$ is generated from $\begin{bmatrix}\mathbf{H}_1^{I_1^{(i)}}\\ \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}$ by random coding, we have colspan $\left(\begin{bmatrix}\mathbf{H}_1^{O_1^{(i)}}\\ \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}\right) \not\subseteq$ colspan $\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}}\\ \mathbf{H}_2^{U_1^{(i)}}\end{bmatrix}\right)$. Consequently,

$$\text{Grank}\left(\mathbf{H}_1^{T_1^{(i)}}, \mathbf{H}_2^{T_1^{(i)}}, \mathbf{G}_2^{T_2^{(i)}}\right) \geq \text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}} & \mathbf{H}_1^{O_1^{(i)}}\\ \mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}\right) \geq 2\,,$$

which contradicts the assumption. Therefore, we conclude,

$$\text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{U_1^{(i)}}\\ \mathbf{H}_2^{U_1^{(i)}}\end{bmatrix}\right) = 0\,, \qquad \text{rank}\left(\begin{bmatrix}\mathbf{H}_1^{I_1^{(i)}}\\ \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \geq 2\,. \tag{52}$$

But by Lemma 9, there is no all zero submatrix of $\begin{bmatrix}\mathbf{H}_1^{T_1^{(i)}}\\ \mathbf{H}_2^{T_1^{(i)}}\end{bmatrix}$. Hence, we conclude that $U_1^{(i)} = \varnothing$.

The fact that rank $\left(\begin{bmatrix}\mathbf{H}_1^{I_1^{(i)}}\\ \mathbf{H}_2^{I_1^{(i)}}\end{bmatrix}\right) \geq 2$ combined with (42) can be used to show that $\left|O_1^{(i)}\right| < 2$. In particular if $\left|O_1^{(i)}\right| \geq 2$, then we have rank $\left(\begin{bmatrix}\mathbf{H}_1^{O_1^{(i)}}\\ \mathbf{H}_2^{O_1^{(i)}}\end{bmatrix}\right) \geq 2$ and thus rank $\left(\begin{bmatrix}\mathbf{H}_1^{T_1^{(i)}}\\ \mathbf{H}_2^{T_1^{(i)}}\end{bmatrix}\right) \geq 2$, which violates (42). Since we have already shown that $O_1^{(i)}$ is non-empty, we conclude that $\left|O_1^{(i)}\right| = 1$.

(iii) $\mathbf{G}_2^{U_2^{(i)}} = \mathbf{0}$, $B_2^{(i)}$ is empty and $A_1^{(i)} = O_1^{(i)}$.

Suppose that $\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix} \neq \mathbf{0}$. We show that this implies that there is an alignment step at stage i contradicting our assumption that the stage contains only randomization steps. This will imply that $\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix} = \mathbf{0}$.

Since, by Claim 2,

$$\text{rank}\left(\mathbf{G}_2^{T_2^{(i)}}\right) = \text{rank}\left(\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{A_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) = 1,$$

we have

$$\text{colspan}\left(\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) = \text{colspan}\left(\mathbf{G}_2^{T_2^{(i)}}.\right)$$

However, from (i) and the fact that $U_1^{(i)}$ is empty, we have

$$\text{colspan}\left(\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) = \text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right).$$

As a result, we have the following at the beginning of recursive algorithm Phase 2 between stage $i+1$ and i,

$$\begin{aligned} \text{Grank}\left(\mathbf{H}_1^{T_1^{(i+1)}}, \mathbf{H}_2^{T_1^{(i+1)}}, \mathbf{G}_2^{T_2^{(i+1)}}\right) &> \text{Grank}\left(\mathbf{H}_1^{I_1^{(i)}}, \mathbf{H}_2^{I_1^{(i)}}, \mathbf{G}_2^{U_2^{(i)}}\right) \\ \text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) &\not\subset \varnothing \\ \text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) &\subseteq \text{colspan}\left(\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix}\right) . \end{aligned}$$

In other words, the conditions for alignment step are satsfied. Thus, the algorithm will carry out an alignment step, which is a contradiction. Therefore, we conclude that $\begin{bmatrix}\mathbf{G}_2^{U_2^{(i)}} & \mathbf{G}_2^{B_2^{(i)}}\end{bmatrix} = \mathbf{0}$. Given that $\mathbf{G}_2^{B_2^{(i)}}$ is obtained by random coding from $\mathbf{G}_2^{I_2^{(i)}}$, we infer that $B_2^{(i)} = \varnothing$; otherwise we would have $\mathbf{G}_2^{B_2^{(i)}} \neq 0$. Furthermore, with rank $\left(\mathbf{G}_2^{T_2^{(i)}}\right) = 1$, we must have rank $\left(\mathbf{G}_2^{A_2^{(i)}}\right) = 1$ and thus $A_2^{(i)}$ is not empty. But $A_2^{(i)}$ is a subset of $O_1^{(i)}$ and $\left|O_1^{(i)}\right| = 1$, so we get $A_2^{(i)} = O_1^{(i)}$.

Therefore, we have $\mathbf{G}_2^{U_2^{(i)}} = \mathbf{0}$, $B_2^{(i)} = \varnothing$ and $A_2^{(i)} = O_1^{(i)}$.

This completes the proof. □

7 Conclusion

The techniques of routing and random network coding have served as pillars of our encoding function design in networks. These techniques are loosely analogous to wireless network achievability techniques of orthogonalization and random coding combined with treating interference as noise respectively. The paradigms of orthogonalization and random coding were challenged by interference alignment in [28]. An important milestone in the development of interference alignment for wireless networks was the development of numerical alignment algorithms [51, 52]. In this paper, we have undertaken an analogous effort for alignment in wireline network coding.

Our paper leads to several open problems. We initiate the study of two-unicast-Z networks. For two-unicast networks, we know that linear coding is insufficient for capacity in general, and that edge-cut outerbounds [26, 53] are loose. In contrast, it is an open question whether even scalar linear network is sufficient for two-unicast-Z networks; similarly, it is not known whether the GNS-cut set bound is loose for two-unicast-Z networks. Second, our approach to maximizing the sum-rate is rather myopic, since we greedily optimize the network coding co-efficients one edge at a time. In comparison, linear programming based algorithms have been formulated for routing, and for network coding restricted to binary field. Development of similar formulations for optimizing the sum-rate of the two-unicast-Z network is a promising research direction. Finally, an interesting question is how our approach compares to other approaches when translated to the index coding problem through the construction of [37].

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

[2] R. Koetter and M. Médard, “An algebraic approach to network coding,” in *IEEE/ACM Trans. Networking*, vol. 11, pp. 782–295, 2003.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, 2003.

[4] T. C. Hu, “Multi-commodity network flows,” *Operations Research*, vol. 11, no. 3, pp. 344–360, 1963.

[5] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*, vol. 24. Springer, 2003.

[6] C. Meng, M. Chen, and A. Markopoulou, “On routing-optimal network for multiple unicasts,” *IEEE International Symposium on Information Theory (ISIT),*, Aug 2014. Extended version on axiv.org.

[7] Z. Li and B. Li, “Network coding: The case of multiple unicast sessions,” in *Allerton Conference on Communications*, vol. 16, 2004.

[8] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Transactions on Information Theory*, vol. 52, pp. 4413 –4430, oct. 2006.

[9] E. Erez and M. Feder, “Capacity region and network codes for two receivers multicast with private and common data,” *Workshop on Coding, Cryptography and Combinatorics*, 2003.

[10] R. W. Yeung, S.-y. Li, and N. Cai, *Network Coding Theory (Foundations and Trends(R) in Communications and Information Theory)*. Hanover, MA, USA: Now Publishers Inc., 2006.

[11] M. Kim, M. Médard, U.-M. O’Reilly, and D. Traskov, “An evolutionary approach to inter-session network coding,” in *INFOCOM 2009, IEEE*, pp. 450–458, IEEE, 2009.

[12] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Transactions on Information Theory*, vol. 51, pp. 2745– 2759, Aug. 2005.

[13] A. Blasiak, R. Kleinberg, and E. Lubetzky, “Lexicographic products and the power of non-linear network coding,” in *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pp. 609–618, Oct 2011.

[14] S. Kamath, D. N. Tse, and C.-C. Wang, “Two-unicast is hard,” in *Information Theory (ISIT), 2014 IEEE International Symposium on*, pp. 2147–2151, IEEE, 2014.

[15] R. Dougherty, C. Freiling, and K. Zeger, “Networks, matroids, and non-shannon information inequalities,” *IEEE Transactions on Information Theory*, vol. 53, pp. 1949–1969, June 2007.

[16] A. R. Lehman and E. Lehman, “Complexity classification of network information flow problems,” in *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 142–150, Society for Industrial and Applied Mathematics, 2004.

[17] M. Médard, M. Effros, D. Karger, and T. Ho, “On coding for non-multicast networks,” in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 41, pp. 21–29, The University; 1998, 2003.

[18] R. Dougherty, C. Freiling, and K. Zeger, "Linear network codes and systems of polynomial equations," *Information Theory, IEEE Transactions on*, vol. 54, pp. 2303–2316, May 2008.

[19] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," *IEEE Transactions on Information Theory*, vol. 57, pp. 1008–1014, Feb 2011.

[20] V. Cadambe, S. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of mds codes in distributed storage," *IEEE Transactions on Information Theory*, vol. 59, pp. 2974–2987, May 2013.

[21] S. Jafar, "Topological interference management through index coding," *IEEE Transactions on Information Theory*, vol. 60, pp. 529–568, Jan 2014.

[22] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, pp. 2856–2867, May 2014.

[23] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," in *2006 IEEE International Symposium on Information Theory*, pp. 1758–1762, July 2006.

[24] T. Ho, Y. Chang, and K. J. Han, "On constructive network coding for multiple unicasts," in *44th Allerton Conference on Communication, Control and Computing*, p. 76, 2006.

[25] C.-C. Wang and N. B. Shroff, "Pairwise intersession network coding on directed networks.," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3879–3900, 2010.

[26] S. Kamath, D. Tse, and V. Anantharam, "Generalized network sharing outer bound and the two-unicast problem," in *International Symposium Network Coding (NetCod)*, pp. 1 –6, July 2011.

[27] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," in *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 1878 –1882, June 2010.

[28] V. Cadambe and S. Jafar, "Interference alignment and the degrees of freedom of the K user interference channel," *IEEE Trans. on Information Theory*, vol. 54, pp. 3425–3441, Aug. 2008.

[29] C. Meng, A. Das, A. Ramakrishnan, S. Jafar, A. Markopoulou, and S. Vishwanath, "Precoding-based network alignment for three unicast sessions," *IEEE Transactions on Information Theory*, vol. 61, pp. 426–451, Jan 2015.

[30] T. Bavirisetti, A. Ganesan, K. Prasad, and B. Rajan, "Precoding-based network alignment using transform approach for acyclic networks with delay," *IEEE Transactions on Information Theory*, vol. 60, pp. 6276–6302, Oct 2014.

[31] Y. Wu and A. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *IEEE International Symposium on Information Theory*, pp. 2276 –2280, 28 2009-july 3 2009.

[32] V. R. Cadambe, C. Huang, S. A. Jafar, and J. Li, "Optimal repair of MDS codes in distributed storage via subspace interference alignment," 2011. Availalbe on http://arxiv.org/abs/1106.1250.

[33] D. Papailiopoulos, A. Dimakis, and V. Cadambe, "Repair optimal erasure codes through Hadamard designs," *IEEE Transactions on Information Theory*, vol. 59, pp. 3021–3037, May 2013.

[34] H. Maleki, V. Cadambe, and S. Jafar, "Index coding - an interference alignment perspective," *IEEE Transactions on Information Theory*, vol. 60, pp. 5402–5432, Sept 2014.

[35] Y. Birk and T. Kol, "Informed-source coding-on-demand (iscod) over broadcast channels," in *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1257 –1264 vol.3, mar-2 apr 1998.

[36] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *Information Theory, IEEE Transactions on*, vol. 56, no. 7, pp. 3187–3195, 2010.

[37] M. Effros, S. E. Rouayheb, and M. Langberg, "An equivalence between network coding and index coding," *arXiv preprint arXiv:1211.6660*, 2012.

[38] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Index coding via linear programming," *arXiv preprint arXiv:1004.1379*, 2010.

[39] K. Shanmugam, A. G. Dimakis, and M. Langberg, "Local graph coloring and index coding," *CoRR*, vol. abs/1301.5359, 2013. http://arxiv.org/abs/1301.5359.

[40] F. Arbabjolfaei, B. Bandemer, Y.-H. Kim, E. Sasoglu, and L. Wang, "On the capacity region for index coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 962–966, IEEE, 2013.

[41] F. Arbabjolfaei and Y.-H. Kim, "Local time sharing for index coding," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, pp. 286–290, IEEE, 2014.

[42] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[43] R. Peeters, "Orthogonal representations over finite fields and the chromatic number of graphs," *Combinatorica*, vol. 16, no. 3, pp. 417–431, 1996.

[44] M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. 29, pp. 439–441, May 1983.

[45] W. Zeng, V. Cadambe, and M. Medard, "An edge reduction lemma for linear network coding and an application to two-unicast networks," in *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 509–516, 2012.

[46] D. B. West *et al.*, *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001.

[47] P. Elias, A. Feinstein, and C. Shannon, "A note on the maximum flow through a network," *IEEE Transactions on Information Theory*, vol. 2, no. 4, pp. 117–119, Dec 1956.

[48] A. E. Gamal and M. Costa, "The capacity region of a class of deterministic interference channels," *IEEE Trans. Inform. Theory*, vol. 2, pp. 343–346, March 1982.

[49] S. Huang and A. Ramamoorthy, "An achievable region for the double unicast problem based on a minimum cut analysis," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2890–2899, 2013.

[50] X. Xu, Y. Zeng, Y. L. Guan, and T. Ho, "An achievable region for double-unicast networks with linear network coding," *IEEE Transactions on Communications*, vol. 62, pp. 3621–3630, Oct 2014.

[51] K. Gomadam, V. Cadambe, and S. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Transactions on Information Theory*, vol. 57, pp. 3309 –3322, June 2011.

[52] S. Peters and R. Heath, "Interference alignment via alternating minimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2445 –2448, April 2009.

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

A Proof of Properties (ii), (iii) and (v)

We first state and prove Lemma 11 which describes a useful property of the destination reduction algorithm. Then we prove Properties *(ii)* and *(iii)*.

Lemma 11. *For any two positive integers p, q with $p < q$, the set of all edges in $T_1^{(q)} \cup T_2^{(q)}$ has a strictly lower topological order with respect to the edges in $\left(T_1^{(p)} \cup T_2^{(p)}\right) \setminus \left(T_1^{(p+1)} \cup T_2^{(p+1)}\right)$.*

Proof. For any integer m, let us denote by $e^{(m)}$, an edge of the highest topological order in set $T_1^{(m)} \cup T_2^{(m)}$. From the destination reduction algorithm, we note that for any integer m, every edge in $\left(T_1^{(m)} \cup T_2^{(m)}\right) \setminus \left(T_1^{(m+1)} \cup T_2^{(m+1)}\right)$ has the same topological order as $e^{(m)}$. Furthermore, from the algorithm, we note that $e^{(m)}$ has a strictly higher topological order with respect to every edge in $T_1^{(m+1)} \cup T_2^{(m+1)}$; specifically, $e^{(m)}$ has a higher topological order with respect to $e^{(m+1)}$. This is because, in the process of obtaining $\Omega^{(m+1)}$ from $\Omega^{(m)}$, we remove the edges with the highest topological order from the two destination sets $T_1^{(m)} \cup T_2^{(m)}$ and replace them with the immediate parent edges, which have a strictly lower topological order. By the transitive nature of partial ordering, we infer that $e^{(p)}$ has a higher topological order with respect to $e^{(q)}$. Therefore $e^{(p)}$ has a higher topological order with respect to every edge in $T_1^{(q)} \cup T_2^{(q)}$. Furthermore, since every edge in $\left(T_1^{(p)} \cup T_2^{(p)}\right) \setminus \left(T_1^{(p+1)} \cup T_2^{(p+1)}\right)$ has the same topological order as $e^{(p)}$, we conclude that every edge in this set has a strictly higher topological order with respect to every edge in $T_1^{(q)} \cup T_2^{(q)}$. □

Proof of Property (ii): For $j \in \{1, 2\}$, let K_j be the set of edges that communicate with destination T_j, but do not belong to $\bigcup_{0 \le k \le N} T_j^{(k)}$. We show that K_1 and K_2 are empty by contradiction.

Suppose that K_1 is non-empty. Let e be the highest topologically ordered edge in K_1. Since there is a path from e to destination 1, there is at least on edge e' which is a member of $\text{Out}(\text{Head}(e))$, such that there is a path from e' to destination 1. Furthermore, since e' does not belong to K_1, there exists an integer k such that e' lies in $T_1^{(k)} \cup T_2^{(k)}$ for some value of k. Let k^* denote the largest among such integers. Therefore, we note that e' does not lie in $T_1^{(k^*+1)} \cup T_2^{(k^*+1)}$. By examining the destination reduction algorithm, we infer that e' is among the edges of the highest topological order in $T_1^{(k^*)} \cup T_2^{(k^*)}$. In the process of obtaining $\Omega^{(k^*+1)}$ from $\Omega^{(k^*)}$, all the edges in $\text{In}(\text{Tail}(e'))$ are added to $T_1^{(k^*+1)}$ or $T_2^{(k^*+1)}$ or both sets, depending on whether e' belongs to $T_1^{k^*}$, or $T_2^{(k^*)}$ or both sets. In particular, edge e is added to $T_1^{(k^*+1)} \cup T_2^{(k^*+1)}$. Therefore edge e does not lie in K_1 which contradicts our earlier assumption. Therefore the set K_1 is empty. We can similarly show that K_2 is empty.

Therefore, we have shown that if an edge communicates with destination j for $j \in \{1, 2\}$, then it belongs to $T_j^{(k)}$ for some integer k. Let k_1 be the largest integer such that e belongs to $T_1^{(k_1)}$. Let k_2 be the largest integer such that e belongs to $T_2^{(k_2)}$. To complete the proof, we show that $k_1 = k_2$. As a contradiction, suppose that $k_1 \neq k_2$. Without loss of generality, assume that $k_1 > k_2$. Then, we note that edge e belongs to $T_2^{(k_2)}$, but does not belong to $T_2^{(k_2+1)}$. By examination of the destination reduction algoritm, we infer that this means that edge e is among the highest topologically ordered edges in $T_2^{(k_2)}$,. and that edge e has a higher topological order with respect to all edges in $T_1^{(k_1+1)} \cup T_2^{(k_2+1)}$.

Since $k_1 > k_2$ and $e \in \left(T_1^{(k_2+1)} \cup T_2^{(k_2+1)}\right) \setminus \left(T_1^{(k)} \cup T_2^{(k)}\right)$, Lemma 11 implies that e has a strictly higher topological order with respect to all the edges in $T_1^{(k_1)} \cup T_2^{(k_1)}$. Therefore e cannot belong to $T_1^{(k_1)}$ contradicting our earlier assumption. Therefore $k_1 = k_2$.

Proof of Property (iii): Let $K^{(i)}$ represent the set of all edges with a lower topological order with respect to $\left(T_1^{(i)} \cup T_2^{(i)}\right) \setminus \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$. From Lemma 11 we infer that

$$\bigcup_{i+1 \leq k \leq N} T_1^{(k)} \cup T_2^{(k)} \subseteq K^{(i)}.$$

To complete the proof, we need to show the reverse, that is, we need to show that

$$K^{(i)} \subseteq \bigcup_{i+1 \leq k \leq N} T_1^{(k)} \cup T_2^{(k)}.$$

Consider any edge e in $K^{(i)}$. Since e communicates with one of the two destinations, Property (ii) implies that e lies in $T_1^{(m)} \cup T_2^{(m)}$ for some integer m. Let $\overline{m}$ denote the largest integer such that e lies in $T_1^{(\overline{m})} \cup T_2^{(\overline{m})}$. To complete the proof, it suffices to show that $\overline{m} \geq i + 1$. We show this by contradiction. Suppose that $\overline{m} < i$. Because e lies in $\left(T_1^{(\overline{m})} \cup T_2^{(\overline{m})}\right) \setminus \left(T_1^{(\overline{m}+1)} \cup T_2^{(\overline{m}+1)}\right)$, we infer from Lemma 11 that e has a higher topological order with respect to every edge in $T_1^{(i)} \cup T_2^{(i)}$. This contradicts the assumption that e lies in $K^{(i)}$. Therefore, we have $\overline{m} \geq i$. If $\overline{m} = i$, then e lies in $\left(T_1^{(i)} \cup T_2^{(i)}\right) \setminus \left(T_1^{(i+1)} \cup T_2^{(i+1)}\right)$. This also contradicts the assumption that e lies in $K^{(i)}$. Therefore we have $m > i$. This completes the proof.

Proof of Property (v): We show that each set $T_1^{(i)}$ is a cut set between s_1, s_2 and destination edge set T_1. The same argument applies to the case of $T_2^{(i)}$. Note that it suffices to show that any path between s_1 or s_2 and some edge in T_1 passes through at least one edge in each set $T_1^{(i)}$, $i = 0, 1, \ldots, N$. Specifically, consider an arbitrary path P with length n, denoted by a sequence of edges, i.e. $P = \{e_1, e_2, \ldots, e_n\}$, where $\text{Head}(e_i) = \text{Tail}(e_{i+1})$ for $1 \leq 1 \leq n$. We show that if $\text{Tail}(e_1) = s_1$ or $\text{Tail}(e_1) = s_2$ and $e_n \in T_1$, then $T_1^{(i)} \cap P \neq \varnothing$ for all $i = 1, 2, \ldots, N$.

First consider the case $i = 0$, since $e_n \in T_1$ and $T_1^{(0)} = T_1$, clearly $T_1^{(0)} \cap P \neq \varnothing$. To complete the proof, we assume that for some k, $0 \leq k \leq N$, $T_1(k) \cap P \neq \varnothing$ holds and show that $T_1^{(k+1)} \cap P \neq \varnothing$. Let e_j be some edge belonging to both P and $T_1^{(k)}$. If $e_j \in T_1^{(k+1)}$, then clearly, $T_1^{(k+1)} \cap P$ is non-empty. Suppose otherwise, i.e. $e_j \notin T_1^{(k+1)}$, then $e_j \in T_1^{(k)} \backslash T_1^{(k+1)}$. By the construction of the destination reduction algorithm, we know that e_j is not a source edge and $\text{In}(\text{Tail}(e_j)) \subseteq T_1^{(k+1)}$. As a result, $e_j \neq e_1$ and $e_{j-1} \in \text{In}(\text{Tail}(e_j))$. We have $e_{j-1} \in P$ and $e_{j-1} \in T_1^{(k+1)}$, that is, we have shown that $T_1^{(k+1)} \cap P \neq \varnothing$. This completes the proof.

B Proof of Lemma 7

For convenience of notation, let $c_{\mathcal{G}_n}(s, T_n) = M$. Since,

$$\text{rank}(\mathbf{H}_{n-1}) = \text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}\right)^{-1}\mathbf{B}_{n-1}^T\right) = M - 1 \tag{53}$$

$$\text{rank}(\mathbf{H}_{n-1}^*) = \text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - \mathbf{F}_{n-1}^*\right)^{-1}\mathbf{B}_{n-1}^{*T}\right) \geq M\ , \tag{54}$$

there exists an $(M-1)\times(M-1)$ submatrix S_1 in $\mathbf{H}_{n-1}$ whose determinant f_1 is not zero. Similarly in matrix $\mathbf{H}_{n-1}^*$, there exists some $M \times M$ submatrix S_2, whose determinant f_2 is not zero.

Next consider the submatrix S_1 in $\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\mathbf{B}_{n-1}^{*T}$ as well as the submatrix S_2 in $\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\mathbf{B}_{n-1}^T$. Let their determinants be $f_1(p,q)$ and $f_2(p,q)$. Clearly $f_1(p,q) \neq 0$ and $f_2(p,q) \neq 0$ since $f_1(1,0) \neq 0$ and $f_2(0,1) \neq 0$. Thus, $f(p,q) = f_1(p,q)f_2(p,q)$ is a non-zero polynomial. Applying Lemma 4 in [8] to $f(p,q)$, we conclude that if the underlying field $\mathbb{F}$ is large enough, choosing p,q uniformly at random from $\mathbb{F}$ will yield $f(p,q) \neq 0$ with a probability that tends to 1 as the field size increases. Equivalently, with the random choices of p and q,

$$\text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\mathbf{B}_{n-1}^T\right) = M - 1 \tag{55}$$

$$\text{rank}\left(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\mathbf{B}_{n-1}^{*T}\right) \geq M\ . \tag{56}$$

Furthermore, since $\mathbf{B}_{n-1}^{*T} = \left[\begin{array}{c|c|c|c} \mathbf{B}_{n-1}^T & \mathbf{u}_{i_1} & \ldots & \mathbf{u}_{i_h} \end{array}\right]$, we conclude that the matrix

$$\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\left[\begin{array}{c|c|c} \mathbf{u}_{i_1} & \ldots & \mathbf{u}_{i_h} \end{array}\right]\ ,$$

which are the symbols carried by the edges e_{i_1} to e_{i_h}, contains at least 1 columns that are linearly independent from the columns of $\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\mathbf{B}_{n-1}^T$. Because $e_{i_1}, \ldots, e_{i_h}$ are the parent edges of $e_{n-1+1}, \ldots, e_n$, by choosing the local coding vectors for the last edge uniformly at random from $\mathbb{F}$, we are setting the columns of

$$\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\left[\mathbf{e}_{n-1+1} \quad \ldots \quad \mathbf{e}_n\right]$$

to be a random linear combinations of the columns of

$$\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\left[\begin{array}{c|c|c} \mathbf{u}_{i_1} & \ldots & \mathbf{u}_{i_h} \end{array}\right]\ .$$

Therefore,

$$\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1} - p\mathbf{F}_{n-1} - q\mathbf{F}_{n-1}^*\right)^{-1}\left[\mathbf{e}_n\right]$$

contains exactly one linearly independent column from $\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1}-p\mathbf{F}_{n-1}-q\mathbf{F}^{*}_{n-1}\right)^{-1}\mathbf{B}^{T}_{n-1}$. Hence, we conclude that,

$$\operatorname{rank}\Big(\mathbf{A}_{n-1}\left(\mathbf{I}_{n-1}-p\mathbf{F}_{n-1}-q\mathbf{F}^{*}_{n-1}\right)^{-1}\left[\begin{array}{c|c}\mathbf{B}^{T}_{n-1} & \mathbf{e}_n\end{array}\right]\Big)=M \tag{57}$$

C Proof of Lemma 2

Since $\operatorname{Grank}\left([\mathbf{H}_1\ \ \mathbf{A}],[\mathbf{H}_2\ \ \mathbf{B}],\mathbf{G}_2\right) > \operatorname{Grank}\left(\mathbf{H}_1,\mathbf{H}_2,\mathbf{G}_2\right)$, it is clear that, $\text{colspan}\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\right) \not\subset \text{colspan}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)$.

We first consider the case when at least one of conditions (i) and (ii) does not hold. In this case, we pick entries of $\mathbf{f}$ uniformly at random from a large enough field.

1. If $\text{colspan}\left(\mathbf{B}\right)\in\text{colspan}\left(\mathbf{H}_2\right)$, then $\mathbf{Bf}\in\text{colspan}\left(\mathbf{H}_2\right)$, but $\mathbf{Af}\notin\text{colspan}\left(\mathbf{H}_1\right)$. Consequently,

$$\begin{aligned}\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \mathbf{Af}\\ \mathbf{H}_2 & \mathbf{Bf}\end{bmatrix}\right)&=\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)+1\ ,\\ \operatorname{rank}\left([\mathbf{H}_2\ \ \mathbf{Bf}\ \ \mathbf{G}_2]\right)&=\operatorname{rank}\left([\mathbf{H}_2\ \ \mathbf{G}_2]\right)\ ,\\ \operatorname{rank}\left([\mathbf{H}_2\ \ \mathbf{Bf}]\right)&=\operatorname{rank}\left(\mathbf{H}_2\right)\ .\end{aligned}$$

2. If $\text{colspan}\left(\mathbf{B}\right)\notin\text{colspan}\left([\mathbf{H}_2\ \ \mathbf{G}_2]\right)$, then $\mathbf{Bf}\notin\text{colspan}\left([\mathbf{H}_2\ \ \mathbf{G}_2]\right)$, and $\mathbf{Af}\notin\text{colspan}\left(\mathbf{H}_1\right)$. Hence,

$$\begin{aligned}\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \mathbf{Af}\\ \mathbf{H}_2 & \mathbf{Bf}\end{bmatrix}\right)&=\operatorname{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)+1\ ,\\ \operatorname{rank}\left([\mathbf{H}_2\ \ \mathbf{Bf}\ \ \mathbf{G}_2]\right)&=\operatorname{rank}\left([\mathbf{H}_2\ \ \mathbf{G}_2]\right)+1\ ,\\ \operatorname{rank}\left([\mathbf{H}_2\ \ \mathbf{Bf}]\right)&=\operatorname{rank}\left(\mathbf{H}_2\right)+1\ .\end{aligned}$$

Therefore, in both cases, we $\operatorname{Grank}\left([\mathbf{H}_1\ \ \mathbf{Af}],[\mathbf{H}_2\ \ \mathbf{Bf}],\mathbf{G}_2\right)=\operatorname{Grank}\left(\mathbf{H}_1,\mathbf{H}_2,\mathbf{G}_2\right)+1$.

It remains to find the vector $\mathbf{f}$ that satisfies (12) when conditions (i) and (ii) both hold. In particular, it suffices to find a vector $\mathbf{f}$ such that

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\mathbf{f}\notin\text{colspan}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right)\ , \tag{58}$$

$$\mathbf{Bf}\in\text{colspan}\left(\mathbf{H}_2\right)\ . \tag{59}$$

Suppose the columns of the matrix $\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}$ are indexed by the index set $[Q_1]=\{1,2,\ldots,Q_1\}$, while the columns are from $\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}$ are indexed by the index set $[M]=\{1,2,\ldots,M\}$. Let Γ be the subset of $[Q_1]$ such that, for the matrix $\mathbf{H}_2$, its submatrix consists columns indexed by Γ forms basis of the matrix. Denote this submatrix as $\mathbf{H}_2^{\Gamma}$. Thus, $\operatorname{rank}\left(\mathbf{H}_2\right)=|\Gamma|$. By the basis extension theorem, there exists a subset of columns of $\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}$, indexed by $\alpha_B\subset[M]$, such that the columns of matrix $[\mathbf{H}_2^{\Gamma}\ \ \mathbf{B}^{\alpha_B}]$ form a basis of $[\mathbf{H}_2\ \ \mathbf{B}]$. Since columns of $\mathbf{H}_2^{\Gamma}$ are linearly independent, columns of

$\mathbf{H}^{\Gamma} = \begin{bmatrix}\mathbf{H}_1^{\Gamma}\\ \mathbf{H}_2^{\Gamma}\end{bmatrix}$ are also linearly independent. Hence, there exists a subset $\alpha_H \subset [Q_1]$, such that $\alpha_H \cap \Gamma = \varnothing$, such that the columns of $\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H}\end{bmatrix}$ form a basis of $\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}$. Consider the matrix $\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H} & \mathbf{A}^{\alpha_B}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H} & \mathbf{B}^{\alpha_B}\end{bmatrix}$. We show that its columns are linearly independent. Consider a length $|\Gamma|$ vector $\mathbf{c}^{\Gamma}$, a length $|\alpha_H|$ vector $\mathbf{c}^{\alpha_H}$ and a length $|\alpha_B|$ vector $\mathbf{c}^{\alpha_B}$, such that

$$\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H} & \mathbf{A}^{\alpha_B}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H} & \mathbf{B}^{\alpha_B}\end{bmatrix}\begin{bmatrix}\mathbf{c}^{\Gamma}\\ \mathbf{c}^{\alpha_H}\\ \mathbf{c}^{\alpha_B}\end{bmatrix} = \mathbf{0}\ . \tag{60}$$

We aim to show that $\begin{bmatrix}\mathbf{c}^{\Gamma}\\ \mathbf{c}^{\alpha_H}\\ \mathbf{c}^{\alpha_B}\end{bmatrix} = \mathbf{0}$. Since the columns of $\mathbf{H}_2^{\Gamma}$ is a basis for $\mathbf{H}_2$, there exists a matrix $\mathbf{D}$ such that $\mathbf{H}_2^{\alpha_H} = \mathbf{H}_2^{\Gamma}$. Thus,

$$\begin{bmatrix}\mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H} & \mathbf{B}^{\alpha_B}\end{bmatrix}\begin{bmatrix}\mathbf{c}^{\Gamma}\\ \mathbf{c}^{\alpha_H}\\ \mathbf{c}^{\alpha_B}\end{bmatrix} = \mathbf{H}_2^{\Gamma}(\mathbf{c}^{\Gamma} + \mathbf{D}\mathbf{c}^{\alpha_H}) + \mathbf{B}^{\alpha_B}\mathbf{c}^{\alpha_B} = \mathbf{0}. \tag{61}$$

But $\begin{bmatrix}\mathbf{H}_2^{\Gamma} & \mathbf{B}^{\alpha_B}\end{bmatrix}$ forms a basis. As a result, $\mathbf{c}^{\alpha_B} = \mathbf{0}$ and $\mathbf{c}^{\Gamma} + \mathbf{D}\mathbf{c}^{\alpha_H} = \mathbf{0}$. Substituting this in (60), we get

$$\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H}\end{bmatrix}\begin{bmatrix}\mathbf{c}^{\Gamma}\\ \mathbf{c}^{\alpha_H}\end{bmatrix} = \mathbf{0}\ . \tag{62}$$

Since columns of $\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H}\end{bmatrix}$ also form a basis, we must have $\mathbf{c}^{\Gamma} = \mathbf{0}$ and $\mathbf{c}^{\alpha_H} = \mathbf{0}$. Hence, columns of $\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H} & \mathbf{A}^{\alpha_B}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H} & \mathbf{B}^{\alpha_B}\end{bmatrix}$ are linearly independent.

By the basis extension theorem, there exits a set $\beta_B \subset [M]\backslash\alpha_B$, such that the columns of $\begin{bmatrix}\mathbf{H}_1^{\Gamma} & \mathbf{H}_1^{\alpha_H} & \mathbf{A}^{\alpha_B} & \mathbf{A}^{\beta_B}\\ \mathbf{H}_2^{\Gamma} & \mathbf{H}_2^{\alpha_H} & \mathbf{B}^{\alpha_B} & \mathbf{B}^{\beta_B}\end{bmatrix}$ form a basis for the matrix $\begin{bmatrix}\mathbf{H}_1 & \mathbf{A}\\ \mathbf{H}_2 & \mathbf{B}\end{bmatrix}$. We next show that $|\beta_B| > 0$.

We know that $\mathrm{Grank}\,([\mathbf{H}_1\ \ \mathbf{A}], [\mathbf{H}_2\ \ \mathbf{B}], \mathbf{G}_2) > \mathrm{Grank}\,(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2)$. Therefore,

$$\begin{aligned}
&\mathrm{Grank}\,([\mathbf{H}_1\ \ \mathbf{A}], [\mathbf{H}_2\ \ \mathbf{B}], \mathbf{G}_2) - \mathrm{Grank}\,(\mathbf{H}_1, \mathbf{H}_2, \mathbf{G}_2)\\
&= \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \mathbf{A}\\ \mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right) - \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right) + \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B} & \mathbf{G}_2\end{bmatrix}\right) - \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right)\\
&\qquad - \mathrm{rank}\,(\mathbf{H}_2) + \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right)\\
&\overset{(a)}{=} \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1 & \mathbf{A}\\ \mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right) - \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}\right) - \left(\mathrm{rank}\,(\mathbf{H}_2) - \mathrm{rank}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right)\right)\\
&= (|\Gamma| + |\alpha_H| + |\alpha_B| + |\beta_B| - |\Gamma| - |\alpha_H|) - (|\Gamma| - |\Gamma| + |\alpha_B|)\\
&= |\beta_B| > 0\ ,
\end{aligned}$$

where (a) follows from condition (ii), i.e. $\mathsf{colspan}\,(\mathbf{B}) \subset \mathsf{colspan}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{G}_2\end{bmatrix}\right)$.

Since $\beta_B \neq \varnothing$, note that the columns of $\begin{bmatrix}\mathbf{H}_2^\Gamma & \mathbf{B}^{\alpha_B} & \mathbf{B}^{\beta_B}\end{bmatrix}$ is a linearly dependent set of vectors. This is because, this set of vectors spans $[\mathbf{H}_2 \quad \mathbf{B}]$, whose basis is $\begin{bmatrix}\mathbf{H}_2^\Gamma & \mathbf{B}^{\alpha_B}\end{bmatrix}$. The nullspace of $\begin{bmatrix}\mathbf{H}_2^\Gamma & \mathbf{B}^{\alpha_B} & \mathbf{B}^{\beta_B}\end{bmatrix}$ therefore contains at least one non-zero vector. Pick an arbitrary non-zero vector from this nullspace. Denote the vector as be $\begin{bmatrix}\mathbf{c}^\Gamma \\ \mathbf{c}^{\alpha_B} \\ \mathbf{c}^{\beta_B}\end{bmatrix}$, where $\mathbf{c}^\Gamma$ is a length $|\Gamma|$ subvector, $\mathbf{c}^{\alpha_B}$ is a length $|\alpha_B|$ subvector and $\mathbf{c}^{\beta_B}$ is a length $|\beta_B|$ subvector. We have, $\mathbf{c}^{\beta_B} \neq \mathbf{0}$ and $\mathbf{H}_2^\Gamma \mathbf{c}^\Gamma + \mathbf{B}^{\alpha_B}\mathbf{c}^{\alpha_B} + \mathbf{B}^{\beta_B}\mathbf{c}^{\beta_B} = \mathbf{0}$, hence, $\mathbf{B}^{\alpha_B}\mathbf{c}^{\alpha_B} + \mathbf{B}^{\beta_B}\mathbf{c}^{\beta_B} \in \mathsf{colspan}\left(\mathbf{H}_2^\Gamma\right) = \mathsf{colspan}\left(\mathbf{H}_2\right)$. On the other hand, since columns of $\begin{bmatrix}\mathbf{H}_1^\Gamma & \mathbf{H}_1^{\alpha_H} & \mathbf{A}^{\alpha_B} & \mathbf{A}^{\beta_B} \\ \mathbf{H}_2^\Gamma & \mathbf{H}_2^{\alpha_H} & \mathbf{B}^{\alpha_B} & \mathbf{B}^{\beta_B}\end{bmatrix}$ is a basis, the columns of $\begin{bmatrix}\mathbf{A}^{\alpha_B} & \mathbf{A}^{\beta_B} \\ \mathbf{B}^{\alpha_B} & \mathbf{B}^{\beta_B}\end{bmatrix}$ is linearly independent of the columns of $\begin{bmatrix}\mathbf{H}_1^\Gamma & \mathbf{H}_1^{\alpha_H} \\ \mathbf{H}_2^\Gamma & \mathbf{H}_2^{\alpha_H}\end{bmatrix}$, which is a basis for $\begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\end{bmatrix}$. Therefore, $\begin{bmatrix}\mathbf{A}^{\alpha_B} \\ \mathbf{B}^{\alpha_B}\end{bmatrix}\mathbf{c}^{\alpha_B} + \begin{bmatrix}\mathbf{A}^{\alpha_B} \\ \mathbf{B}^{\alpha_B}\end{bmatrix}\mathbf{c}^{\beta_B} \notin \mathsf{colspan}\left(\begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\end{bmatrix}\right)$.

We are now ready to specify $\mathbf{f}$ satisfying (58) and (59). For any set $I \subseteq [M]$, let $\mathbf{f}^I$ denote the entries of vector $\mathbf{f}^I$ corresponding to set I. We specify the $M \times 1$ vector $\mathbf{f}$ as follows: $\mathbf{f}^{\alpha_B} = \mathbf{c}^{\alpha_B}$, $\mathbf{f}^{\beta_B} = \mathbf{c}^{\beta_B}$, and $\mathbf{f}^I = \mathbf{0}$ for any set I which is disjoint with $\alpha_B \cup \beta_B$. We note that with this choice of $\mathbf{f}$, we have

$$\mathbf{B}\mathbf{f} = \mathbf{B}^{\alpha_B}\mathbf{c}^{\alpha_B} + \mathbf{B}^{\beta_B}\mathbf{c}^{\beta_B}$$

$$\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}\mathbf{f} = \begin{bmatrix}\mathbf{A}^{\alpha_B} \\ \mathbf{B}^{\alpha_B}\end{bmatrix}\mathbf{c}^{\alpha_B} + \begin{bmatrix}\mathbf{A}^{\alpha_B} \\ \mathbf{B}^{\alpha_B}\end{bmatrix}\mathbf{c}^{\beta_B}$$

It can be readily verified that (58) and (59) are satisfied.

So far, we have showed that there is at least one vector $\mathbf{f}$ which satisfies (58) and (59) and subsequently satisfies (12). Next, we show that if conditions (i) and (ii) hold, then picking a random vector $\mathbf{v}$ from the nullspace of $\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}$ and setting $\mathbf{f}$ to be the last M entries of the random vector $\mathbf{v}$ satisfies (12) with high probability,. In particular, we show that such a vector $\mathbf{f}$ satisfies (58) and (59) with high probability.

Let $\mathbf{v} = \begin{bmatrix}\mathbf{v}_1 \\ \mathbf{v}_2\end{bmatrix}$, $\mathbf{v}_1 \in \mathbb{F}^{Q_1}, \mathbf{v}_2 \in \mathbb{F}^M$, be a random vector from the nullspace of $\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}$, i.e. $\mathbf{H}_2\mathbf{v}_1 + \mathbf{B}\mathbf{v}_2 = \mathbf{0}$. Clearly, $\mathbf{f} = \mathbf{v}_2$ satisfies (59). We need to show that $\mathbf{f}$ chosen this way satisfies (58) with high probability. Note that the vector $\mathbf{v}_2$ fails to satisfy (58) if and only if there exists some $\mathbf{v}_1' \in \mathbb{F}^{Q_1}$, such that,

$$\begin{bmatrix}\mathbf{H}_1 & \mathbf{A} \\ \mathbf{H}_2 & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{v}_1' \\ \mathbf{v}_2\end{bmatrix} = \mathbf{0}\ . \tag{63}$$

Let $\mathcal{R}$ be the set of all such vectors in the nullspace of $\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}$, i.e.

$$\mathcal{R} = \left\{\begin{bmatrix}\mathbf{v}_1 \\ \mathbf{v}_2\end{bmatrix} \in \mathrm{Ker}\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right) : \exists\ \mathbf{v}_1' \in \mathbb{F}^{Q_1}, s.t. \begin{bmatrix}\mathbf{H}_1 & \mathbf{A} \\ \mathbf{H}_2 & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{v}_1' \\ \mathbf{v}_2\end{bmatrix} = \mathbf{0}\right\}\ . \tag{64}$$

Clearly, if $\begin{bmatrix}\mathbf{v}_1 \\ \mathbf{v}_2\end{bmatrix}$, $\begin{bmatrix}\mathbf{u}_1 \\ \mathbf{u}_2\end{bmatrix} \in \mathcal{R}$, then there exists $\mathbf{v}_1', \mathbf{u}_1' \in \mathbb{F}^{Q_1}$, s.t.

$$\begin{bmatrix}\mathbf{H}_1 & \mathbf{A} \\ \mathbf{H}_2 & \mathbf{B}\end{bmatrix}\begin{bmatrix}a\mathbf{v}_1' + b\mathbf{u}_1' \\ a\mathbf{v}_2 + b\mathbf{u}_2\end{bmatrix} = \mathbf{0} \tag{65}$$

for any $a, b \in \mathbb{F}$, i.e. $\begin{bmatrix} a\mathbf{v}'_1 + b\mathbf{u}'_1 \\ a\mathbf{v}_2 + b\mathbf{u}_2 \end{bmatrix} \in \mathcal{R}$. Hence, $\mathcal{R}$ is a subspace of Ker $\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right)$.

Furthermore, note that $\mathcal{R}$ is a proper subspace. This is because we have already shown that there exists a vector in the null space of $\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}$ that satisfy (58), that is, we have already shown the existence of a vector in the nullspace $\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}$ that does not lie in $\mathcal{R}$. Therefore, by picking $\mathbf{v}$ uniformly at random from Ker $\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right)$, the probability that both condition (58) and (59) are satisfied is

$$1 - P(\mathbf{v} \in \mathcal{R}) = 1 - \frac{1}{|\mathbb{F}|^t}\ , t \in \mathbb{Z}^+ \tag{66}$$

where $|\mathbb{F}|$ is the size of the underlying field and t is the difference between the dimension of subspace $\mathcal{R}$ and the dimension of Ker $\left(\begin{bmatrix}\mathbf{H}_2 & \mathbf{B}\end{bmatrix}\right)$. Hence, as the field size increases arbitrarily, the probability of both (58) and (59) hold approaches to 1. This completes the proof.

D Proof of Lemma 9

Clearly, at the initial stage, by the construction of the algorithm, every column of the matrix corresponds to a source edge and is assigned a unit vector. As a result, none of the columns in the matrix $\begin{bmatrix} \mathbf{H}_1^{T_1^{(N)}} \\ \mathbf{H}_2^{T_1^{(N)}} \end{bmatrix}$ is all-zero.

Suppose that there is no all-zero column in the matrix $\begin{bmatrix} \mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}} \end{bmatrix}$. Clearly, in $\begin{bmatrix} \mathbf{H}_1^{T_1^{(i)}} \\ \mathbf{H}_2^{T_1^{(i)}} \end{bmatrix}$, no all-zero columns can be generated from random coding of columns of $\begin{bmatrix} \mathbf{H}_1^{T_1^{(i+1)}} \\ \mathbf{H}_2^{T_1^{(i+1)}} \end{bmatrix}$ with a probability that tends to 1 as the field size increases. It remains to consider the case when alignment happens. But by Lemma 2, the alignment step will generate a column that does not belong to the column span of the matrix $\begin{bmatrix} \mathbf{H}_1^{U_1^{(i)}} \\ \mathbf{H}_2^{U_1^{(i)}} \end{bmatrix}$. Hence, any column generated by alignment cannot be all zero either. This completes the proof.

E Proof of Lemma 10

Suppose that from stage $k+1$ to stage k, the algorithm performs some alignment step. Since $A_2^{(k)} \subset O_1^{(k)}$, we have $Q^{(k)} = T_2^{(k)} \backslash T_1^{(k)} = U_2^{(k)} \cup B_2^{(k)} \backslash U_1^{(k)}$. By the construction of the algorithm, if at some step j in phase 2, an alignment step takes place, then we have,

$$\text{colspan}\left(\mathbf{H}_2^{I_1^{(i)}}\right) \not\subset \text{colspan}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(i)}} & \mathbf{H}_2^{O_{1,j-1}^{(i)}}\end{bmatrix}\right),$$
$$\text{colspan}\left(\mathbf{H}_2^{I_1^{(k)}}\right) \subset \text{colspan}\left(\begin{bmatrix}\mathbf{H}_2^{U_1^{(k)}} & \mathbf{H}_2^{O_{1,j-1}^{(k)}} & \mathbf{G}_2^{U_2^{(k)}} & \mathbf{G}_2^{A_{2,j-1}^{(k)}} & \mathbf{G}_2^{B_2^{(k)}}\end{bmatrix}\right) \ .$$

Since $A^{(k)}_{2,j-1} \subset O^{(k)}_{1,j-1}$, $\mathbf{G}_2^{A^{(k)}_{2,j-1}}$ is a submatrix of $\mathbf{H}_2^{O^{(k)}_{1,j-1}}$. For the conditions to hold, we must have

$$\mathbf{G}_2^{U_2^{(k)} \cup B_2^{(k)} \backslash U_1^{(k)}} = \mathbf{G}_2^{Q^{(k)}} \neq \mathbf{0} \ .$$

Now consider $Q^{(k-1)} = T_2^{(k-1)} \backslash T_1^{(k-1)}$. By the construction of the algorithm, $Q^{(k-1)}$ does not communicate with T_1. Hence, all the columns corresponding to edges in $Q^{(k-1)}$ are either source edge columns or columns generated by random coding in phase 1 at some stage of the algorithm. On the other hand, since all edges in $Q^{(k)}$ communicate with T_2 but not T_1, for each edge $e_i^{(k)} \in Q^{(k)}$ there exists an edge $e_i^{(k-1)} \in Q^{(k-1)}$ such that either $e_i^{(k)} = e_i^{(k-1)}$ or $e_i^{(k)}$ is a parent edge of $e_i^{(k-1)}$. Thus, each column in $\mathbf{G}_2^{Q^{(k)}}$ participates in random coding for at least one edge in $Q^{(k-1)}$. Now since $\mathbf{G}_2^{Q^{(k)}} \neq \mathbf{0}$, we conclude that at least one column of $\mathbf{G}_2^{Q^{(k-1)}}$ is not all-zero and thus $\mathbf{G}_2^{Q^{(k-1)}}$ is not a zero matrix. Subsequently we have, for all $0 \leq i \leq k$, $\mathbf{G}_2^{Q^{(i)}} \neq \mathbf{0}$, which completes the proof.