On the Capacity of Wireless 1-Hop Intersession Network Coding — A Broadcast Packet Erasure Channel Approach

Chih-Chun Wang, Member, IEEE

Abstract—Motivated by practical wireless network protocols, this paper focuses on wireless intersession network coding (INC) over a 1-hop neighborhood, of which the exact capacity region remains an open problem. Towards better understanding of the capacity, this work first models the wireless overhearing events by broadcast packet erasure channels that are memoryless and stationary. Since most INC gain is resulted from destinations overhearing packets transmitted by other sources, this work then focuses exclusively on the 2-staged INC schemes, which fully capture the throughput benefits of overheard message side information (MSI) through the use of one-time feedback but refrain from exploiting the broadcast spatial diversity gain of channel output feedback. Under this setting, a capacity outer bound is provided for any number of $M$ coexisting unicast sessions. For the special cases of $M \leq 3$, it is shown that the outer bound can be achieved and is indeed the capacity. To quantify the tightness of the outer bound for $M \geq 4$, a capacity inner bound for general $M$ is provided. Both the outer and inner bounds can be evaluated by any linear programming solver. Numeric results show that for $4 \leq M \leq 5$ with randomly chosen channel parameters, the difference between the outer and inner bounds is within 1% for 96.7% of the times. Focusing exclusively on the benefits of MSI, the results in this paper can also be viewed as the generalization of index-coding capacity from wireline broadcast with binary alphabets to wireless broadcast with high-order alphabets.

Index Terms—Broadcast channel capacity, message side information, network code alignment, packet erasure channels, wireless 1-hop network coding.

I. INTRODUCTION

The throughput benefits of network coding (NC) have been demonstrated and elegantly characterized in [28] for the setting of coding over a single multicast session, termed intrasession NC. However, since most traffic is unicast for which intrasession NC provides no capacity gains (cf. [21], [28] for wireline networks and [11], [19], [36] for wireless erasure networks with acknowledgment), the throughput benefits of NC can be fully realized only when considering both coding within a single multicast session [28] and coding over multiple unicast sessions, the so-called intersession network coding (INC) [41]. Determining the capacity of INC is a notoriously challenging problem even for wireline networks [14]–[16]. For example, when considering the integral setting with an unbounded number of coexisting sessions, the complexity of determining the INC capacity is NP-complete [27], [41]. To mitigate its complexity, recent practical wireless INC schemes [20] focus on INC over a 1-hop neighborhood. Despite its simplified setting and attractive complexity savings, the exact capacity region of 1-hop INC remains an open problem and several attempts have since been made to quantify some suboptimal achievable rate regions [6], [10], [26], [33], [35], [39], [47]. Motivated by practical wireless protocols, this paper provides a new model and the corresponding analysis that make progress towards answering the following open questions: Exactly (or at most) how much throughput improvement one can expect from INC in a wireless 1-hop neighborhood over non-coding solutions; and how to achieve (or approach) the corresponding INC capacity.

Most 1-hop INC schemes [20], [44] take advantage of the broadcast nature of wireless media. Take Fig. 1(a) for example. Two unicast sessions $(s_1, d_1)$ and $(s_2, d_2)$ would like to send packets $X_1$ and $X_2$, respectively, through a common relay node $r$. Assuming half-duplex and full interference, each node can either transmit or receive, and no two nodes can transmit simultaneously. Also assuming that there is no noise in the wireless links and each transmitted packet always arrives at the target receiver successfully, without network coding four time slots are necessary to complete the two sessions: $s_1 \xrightarrow{X_1} r$, $s_2 \xrightarrow{X_2} r$, $r \xrightarrow{d_1}$, and $r \xrightarrow{X_2} d_2$. [20] observes that if we further assume that each transmitted packet can always be overheard by more than one node, then one can complete the transmission in three time slots by sending packets: $s_1 \xrightarrow{X_1} \{r, d_2\}$, $s_2 \xrightarrow{X_2} \{r, d_1\}$, and $r \xrightarrow{X_1+X_2} \{d_1, d_2\}$. See Fig. 1(a) for illustration. Destinations $d_1$ and $d_2$ can then decode the desired packets by subtracting the overhead side information $X_2$ and $X_1$, respectively, from the coded message $[X_1+X_2]$. This simple illustrative example quickly becomes a challenging problem when the wireless channels are not perfect. What if $d_i$ cannot overhear all the transmissions from $s_j$ for distinct $i, j \in \{1, 2\}$? What if a coded packet broadcast by $r$ cannot reach both $d_1$ and $d_2$ simultaneously? How much throughput improvement can we still expect from INC? The setting becomes even more complicated when considering the random overhearing events\(^1\)

\(^{1}\)The number of types of different overhearing events is generally proportional to $M^2$, as will be clear in Section II.

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

This work was supported in part by the NSF Awards CCF-0845968 and CNS-0905331.

Part of the results was published in ISIT 2010. Manuscript received ??, 2010; revised ??, 2010.
among $M$ coexisting sessions $(s_i, d_i), i \in [M] \triangleq \{1, \cdots, M\}$ for general $M > 2$ values.

This capacity problem is related to the Gaussian broadcast channel (GBC) problem with message side information (MSI) [22], [25], [32], [37], [42], [46]. Namely, wireless 1-hop INC can be modeled as the following two-staged scheme: In the first stage, all sources $s_j, j \in [M]$ transmit. The destinations $d_i$ can store the overheaded transmission of Stage-1 and use it as MSI in the subsequent stage. In the end of Stage 1, a one-time feedback is sent from all $d_i$ back to $r$ so that the relay $r$ knows which MSI is available at which destination. In the second stage, the relay node $r$ is facing a broadcast channel problem such that $r$ would like to broadcast $M$ data streams (one for each session) to $M$ destinations, respectively, while taking advantage of the MSI available at $d_i, i \in [M]$. Although the GBC capacity without MSI is well known for general $M$ values, the corresponding capacity with general MSI is currently known only for the simplest case of $M \leq 2$ [42] and remains an open problem for $M \geq 3$.

Another related setting that is developed independently in the computer science community is the index coding problem [3], [7]. Index coding considers a similar broadcast with MSI problem (i.e. the Stage-2 transmission from $r$ to $\{d_i\}$) but focuses on noise-free binary channels and each session having 1 bit of information to transmit, which is different from the GBC settings of additive Gaussian noise with each session 1 bit of information to transmit, which is different from the GBC settings of additive Gaussian noise with each session having $w$ bit of information to transmit, which is different from the GBC settings of additive Gaussian noise with each session $w$ bit of information to transmit, which is different from the GBC settings of additive Gaussian noise with each session.

Similar to the existing works, this paper models the wireless 1-hop INC problem by the same 2-stage setting as in GBCs with MSI and in index coding. On the other hand, to capture the packet-by-packet behavior in wireless networks, we consider broadcast packet erasure channels (PECs) [11], [12], [17], [24], [34], [40], [45] instead of the GBC or the noise-free channel in index coding. Each transmission symbol (a packet) is chosen from a finite field $\mathbb{GF}(q)$ for some sufficiently large $q$ and our objective is to maximize the achievable real-valued rate $R_i$, which is different from the symbol power constraints in GBCs and from the binary-alphabet, integral setting of index coding. Unlike the existing GBC and index coding results that focus mainly on the broadcast problem, we further take into account how $s_i$ should transmit in the first stage in order to optimize the overheaded MSI that best benefits the subsequent broadcast stage.$^2$

Focusing on the 2-staged setting, a capacity outer bound is provided for any number $M$ of coexisting unicast sessions and any overhearing events modeled by PECs that are memoryless and stationary. For the cases of $M \leq 3$, it is shown that the outer bound meets the capacity, which is a step forward from the $M = 2$ capacity characterization in the GBC with MSI. To quantify the tightness of the outer bound for $M \geq 4$, a capacity inner bound for general $M$ values is provided, which is based on the concept of code alignment in the finite field (also see [13], [38], [40], [43]) that is in parallel with the interference alignment technique [4] in the Euclidean space. Both bounds can be computed by linear programming (LP) solvers for any $M$ value. Numeric results show that for $4 \leq M \leq 5$ with randomly chosen PEC parameters, the gap between the two bounds is within 1% for 96.7% of the times, which demonstrates the effectiveness of our capacity bounds from a practical perspective.

The rest of this paper is organized as follows. Section II provides the formal problem formulation. Section III states the main results, which include the capacity region of 2-staged wireless 1-hop INC for $M \leq 3$ sessions and a pair of outer and inner bounds for general $M$ sessions. Section IV presents the equivalent LP-based form of the outer bound. Section V proves the capacity characterization for $M \leq 3$, while Section VI proves the achievability results for general $M$ values. The results of numerical experiments and some observation remarks are included in Section VII. Section VIII concludes the paper.

II. PROBLEM FORMULATION

A. Packet Erasure Channels (PECs)

For any positive integer $K$, we use $[K] \triangleq \{1, 2, \cdots, K\}$ to denote the set of integers from 1 to $K$, and use $2^{[K]}$ to denote the collection of all subsets of $[K]$.

A 1-to-$K$ PEC takes an input $W \in \mathbb{GF}(q)$ and outputs a $K$-dimensional vector in $Z \triangleq (Z_1, \cdots, Z_K) \in \{(W) \cup \{\cdot\})^K$, where the $k$-th coordinate $Z_k$ being “*” denotes that the transmitted symbol $W$ does not reach the $k$-th receiver (thus being erased). We also assume that there is no other type of noise, i.e., the individual output $Z_k$ is either equal to the input $W$ or an erasure “*.” The success probabilities of a 1-to-$K$ PEC from a single source $s$ to $K$ destinations are described by $2^K$ non-negative parameters: $p_{s \rightarrow T} = T|W = w) = p_{s \rightarrow T}^{W = w}$.

That is, $p_{s \rightarrow T}^{W = w}$ denotes the probability that the transmitted symbol $W$ is received by and only by the receivers in $T$. We

$^2$It is worth mentioning that the 2-staged setting in this work does incur some suboptimality as we refrain the relay $r$ from exploiting the spatial diversity gain of broadcast channels with feedback [31]. On the other hand, the 2-staged setting fully captures the benefits that destinations could overhear packets transmitted from other sources, which is a major contribution of the capacity gain of wireless INC. See Section II-C for further discussion.
can further generalize this notation by defining

\[ p_{s \rightarrow T_1 T_2} = \sum_{T \in 2^{[K]} : T_1 \subseteq T, T_2 \subseteq ([K] \setminus T)} p_{s \rightarrow T | [K] \setminus T}. \]

for any pair of disjoint sets \( T_1, T_2 \in 2^{[K]} \). Namely, \( p_{s \rightarrow T_1 T_2} \) is the probability that the transmitted symbol \( W \) is received by all the receivers in \( T_1 \) and not by any receivers in \( T_2 \). This notation \( p_{s \rightarrow T_1 T_2} \) can be viewed as a marginal success probability that does not care whether the receivers outside \((T_1 \cup T_2)\) receive the transmitted symbol or not.

The following notation will also be used frequently in this work. For all \( T \in 2^{[K]} \), we define

\[ p_{s \upharpoonright T} = \sum_{T' \in 2^{[K]} : T' \cap T \neq \emptyset} p_{s \rightarrow T' | [K] \setminus T'}. \]

That is, \( p_{s \upharpoonright T} \) is the probability that at least one of the receivers in \( T \) successfully receives the transmitted symbol \( W \). By convention, we define \( p_{s \upharpoonright \emptyset} = 0 \). For example, when \( K = 2 \),

\[ p_{s \upharpoonright \{1,2\}} = p_{s \rightarrow \{1\}} + p_{s \rightarrow \{2\}} + p_{s \rightarrow \{1,2\}} \]

is the probability that at least one of the two receivers receives the transmitted symbol \( W \). It can be verified by definition that both the following two notations

\[ \forall k \in [K], \ p_{s \rightarrow \{k\} \emptyset} = p_{s \rightarrow \{k\}} \tag{1} \]

describe the marginal probability that the \( k \)-th receiver receives \( W \) successfully. We sometimes use \( p_{s \upharpoonright k} \) as shorthand for \( (1) \).

We can repeatedly use the channel for \( n \) time slots and let \( W(t) \) and \( Z(t) \) denote the input and output for the \( t \)-th time slot. We assume that the 1-to-\( K \) broadcast PEC is memoryless and time-invariant, i.e., for any given functions \( w(\cdot) : [n] \rightarrow \text{GF}(q) \), and \( T(\cdot) : [n] \rightarrow 2^{[K]} \),

\[ \text{Prob}(\forall t \in [n], \{ k : Z_k(t) = w(t) \} = T(t)) \]

\[ \forall t \in [n], W(t) = w(t) = \prod_{t=1}^{n} p_{s \rightarrow T(t) | [K] \setminus T(t)}. \]

Note that the above setting allows the success events among different receivers to be dependent, also defined as spatial dependence. For example, when two logical receivers are situated in the same physical node, we simply set the \( p_{s \rightarrow T | [K] \setminus T} \) parameters to allow perfect correlation between the success events of the two receivers.

B. The Canonical 1-Hop Relay Network With No Direct Path

We consider the following canonical 1-hop relay network. Such a network consists of \( M \) source/destination pairs \((s_1, d_1), \forall i \in [M] \) and a common relay node \( r \). Source \( s_i \) would like to send \( nR_i \) information packets \( X_{i,1} \) to \( X_{i,nR_i} \) to destination \( d_i \) through the common relay \( r \) for some positive integer \( n \) and fractional rate \( R_i \). For notational simplicity, we define the information vector \( X_i \triangleq (X_{i,1}, \cdots, X_{i,nR_i}). \)

The sources, destinations, and the relay are connected by broadcast PECs. More specifically, each \( s_i \) is associated with a 1-to-\( M \) PEC that sends a symbol \( W_i \in \text{GF}(q) \) from \( s_i \) to \( \{ d_j : \forall j \in [M] \setminus \{ i \} \cup \{ r \} \} \). For each \( s_i \), the outputs of its

1-to-M PEC is denoted by \( Z_{i \rightarrow j}, j \in [M] \setminus i \) and by \( Z_{i \rightarrow r} \), which refer to the outputs arriving at destination \( d_i \) and the relay \( r \), respectively. The common relay \( r \) is also associated with a 1-to-M PEC that sends a symbol \( W_r \in \text{GF}(q) \) from \( r \) to \( \{ d_j : \forall j \in [M] \} \). The corresponding output symbol that arrives at destination \( d_i \) is denoted by \( Z_{r \rightarrow i} \), for all \( j \in [M] \).

We consider a block network code setting. Namely, each of the sources \( s_j \to s_M \) and the relay \( r \) can use their associated PECs for exactly \( n \) times. We use the following vector notation to denote the \((M+1)\) transmitted symbols and their corresponding outputs. For all \( i \in [M] \), we define

\[ W_i \triangleq (W_i(1), W_i(2), \cdots, W_i(n)). \]

\[ Z_{i \rightarrow r} \triangleq (Z_{i \rightarrow r}(1), Z_{i \rightarrow r}(2), \cdots, Z_{i \rightarrow r}(n)) \]

\[ W_r \triangleq (W_r(1), W_r(2), \cdots, W_r(n)). \]

\[ Z_{r \rightarrow i} \triangleq (Z_{r \rightarrow i}(1), Z_{r \rightarrow i}(2), \cdots, Z_{r \rightarrow i}(n)), \]

and for all \( i, j \in [M] \) and \( i \neq j \), we define

\[ Z_{i \rightarrow j} \triangleq (Z_{i \rightarrow j}(1), Z_{i \rightarrow j}(2), \cdots, Z_{i \rightarrow j}(n)), \]

for which we use the input argument “(t)”, \( t = 1, \cdots, n \), to distinguish the \( n \) channel usages. An illustration of a canonical 1-hop relay network with \( M = 2 \) is provided in Fig. 2. In this work, we also assume that all the PECs are memoryless and stationary. The \((M+1)\) PEC channels can thus be described by the joint success probabilities \( p_{r \rightarrow \cdot}, \) and \( p_{s_i \rightarrow \cdot} \) for all \( i \in [M] \) as described in Section II-A. For example, if \( M = 4 \), \( p_{r \rightarrow (1,4) | (2,3)} \) denotes the probability that a packet sent by \( r \) is received by \( d_1 \) and \( d_4 \) but not by \( d_2 \) and \( d_3 \). Similarly, \( p_{s_2 \rightarrow (3,4) | (1,2)} \) denotes the probability that a packet sent by \( s_2 \) is received by \( d_1 \) and \( d_4 \) but not by \( d_1 \) and \( r \). The channel statistics \( p_{r \rightarrow \cdot} \) and \( p_{s_i \rightarrow \cdot} \) are assumed to be known to all network nodes, which could be achieved through a training period.

\[ \text{Fig. 2. Illustration of the canonical 1-hop relay network with no direct communication between the source and destination } (s_i, d_i) \text{ for the case of } M = 2. \]

\[ \text{Most existing wireless protocols, including the practical INC protocol in (20), assume a single route and focus on transmit all packets to its immediate downstream neighbor. Motivated by this observation, we preclude any 2-hop transmission directly from } s_i \text{ to } d_i \text{ in the canonical setting. The recent concept of opportunistic routing, which allows 2-hop overhearing directly, will be discussed in Section II-D.} \]

\[ \text{Most existing wireless protocols, including the practical INC protocol in (20), assume a single route and focus on transmit all packets to its immediate downstream neighbor. Motivated by this observation, we preclude any 2-hop transmission directly from } s_i \text{ to } d_i \text{ in the canonical setting. The recent concept of opportunistic routing, which allows 2-hop overhearing directly, will be discussed in Section II-D.} \]
Before defining the 2-staged network coding capacity, we define the channel status vector of the PEC outputs $Z_{i,nR_i}$ by

$$1(Z_{i,nR_i}(\cdot) \neq 0) \triangleq 1(Z_{i,nR_i}(1) \neq 0), \cdots, 1(Z_{i,nR_i}(n) \neq 0),$$

where $1(\cdot)$ is the indicator function. Namely, $1(Z_{i,nR_i}(\cdot) \neq 0)$ is an $n$-dimensional binary vector such that its $t$-th coordinate is zero if and only if in the $t$-th channel usage of source $s_i$, destination $d_j$ receives an erasure.

A 2-staged 1-hop intersession network code is then defined by $(M + 1)$ encoding functions $g_s(\cdot)$ and $g_r(\cdot)$:

$$\forall i \in [M], \ W_i = g_s(X_i)$$

and $M$ decoding functions $g_d(\cdot)$:

$$\forall j \in [M], \ \hat{X}_j = g_d(\{Z_{i,nR_i} : \forall i \in [M]\})$$

Namely, each source $s_i$ generates the coded packet $W_i$ based on its information packets $X_i$, see (2). Then each destination $d_j$ is allowed to feedback the channel status vectors $1(Z_{i,nR_i}(\cdot) \neq 0) : \forall i \in [M]\}$ back to the relay $r$. The relay $r$ will make its coding decision based on what it receives from the sources $s_i$ and on the overhearing patterns reported by the destinations $d_j$ through the channel status vectors, see (3). In the end, each destination $d_j$ decodes the original packets $\hat{X}_j$ on the packets received from the sources $\{s_i : \forall i \in [M]\}$ and the relay $r$, see (4). The encoding and decoding functions can be linear or nonlinear.

Assume that the information symbols $X_i = (X_{i,1}, \cdots, X_{i,nR_i})$ are independently and uniformly distributed over $GF(q)$ for all $i \in [M]$. We can then define the achievable rate of any 2-staged 1-hop intersession network code as follows.

**Definition 1:** A rate vector $(R_1, \cdots, R_M)$ is achievable if for any $\epsilon > 0$, there exists a 2-staged 1-hop intersession network code satisfying

$$\forall i \in [M], \ P(\hat{X}_i \neq X_i) < \epsilon,$$

for sufficiently large $n$ and sufficiently large underlying finite field $GF(q)$.

**Definition 2:** The 2-staged INC capacity region of a canonical 1-hop relay network of $M$ sessions is defined as the closure of all achievable rate vectors $(R_1, \cdots, R_M)$.

C. Comparison to The Practical COPE Protocol

The definition of a 1-hop intersession network code in (2) to (4) models closely the following 2-stage wireless 1-hop networking protocol. That is, all sources $s_i$ and the relay $r$ have full knowledge of the PEC parameters before transmission. Each $(s_i, d_i)$ session has $nR_i$ information packets $X_{i,1}, \cdots, X_{i,nR_i}$ to transmit. In the first stage, the sources $s_i, i \in [M]$, take turns and each $s_i$ uses the 1-to-$M$ PEC for $n$ times and sends out $n$ coded symbols generated from the information symbols $X_{i,1}, \cdots, X_{i,nR_i}$, possibly in a non-linear fashion. Due to the randomness of wireless channels, some packets will arrive at $r$ and/or be overhead by $d_j$ and some will not. In the beginning of the second stage, all destinations $\{d_i : i \in [M]\}$ report to the relay $r$ the reception status of the packets sent in the first stage so that $r$ is fully aware of the MSI available at individual $d_i$. The relay $r$ then uses its 1-to-$M$ PEC for $n$ times and sends out coded packets generated from the packets received by $r$ during the first stage, possibly in a non-linear fashion. No feedback is allowed in the middle of the second stage. In practice, feedback is costly and the “reception reports” may only be sent infrequently [20]. Our setting thus also models the extreme case that a one-time exchange of reception/channel status bit maps is allowed only in the end of Stage 1. After the second stage, each $d_i$ computes the decoded symbols $\hat{X}_i \triangleq (\hat{X}_{i,1}, \cdots, \hat{X}_{i,nR_i})$ based on all the symbols received by $d_i$.

This 2-stage scheme can be viewed as a fully coordinated, delay elastic approximation of the practical COPE protocol [20]. COPE protocol contains three major components: (i) Opportunistic listening: Each destination is in a promiscuous monitoring mode and stores all the overhead packets, (ii) Opportunistic coding: The relay node decides which packets to be coded together opportunistically, based on the overhearing patterns of its neighbors, and (iii) Learning the states of the neighbors: Although reception reports are periodically sent to advertise the overhearing patterns of the next-hop neighbors of the relay, the relay node still needs to extrapolate the overhearing status of its neighbors since there is always a time lag due to the infrequent periodic feedback.

Our 2-stage scheme closely captures the opportunistic listening component of COPE by modeling the wireless packet transmission as a random broadcast packet erasure channel. In (3), the channel status vector is used to make the coding decision, which captures the opportunistic coding component of COPE. By allowing sufficiently large block length $n$, we also circumvent the problem of learning the states of the neighbors since we simply wait for a sufficiently large $M \cdot n$ time slots and then use the channel status vectors to convey the exact overhearing pattern to the relay $r$. There is thus no need for extrapolating the overhearing status.

In COPE, the reception reports are broadcast periodically. In contrast, in our 2-staged setting the relay is only allowed to use the channel status vectors of the source-to-destination output $Z_{i,nR_i}$ (see (3)) but not the status vectors of the relay-to-destination output $Z_{r,nR_i}$. Since channel output feedback could potentially increase the broadcast capacity [17], [31], [45], the periodic broadcast of reception reports of COPE could lead to a throughput advantage over a 2-staged scheme. On the other hand, empirically, most throughput gain of the COPE protocol follows from the MSI instead of the channel out-

Since we focus only on 1-hop INC at relay $r$, we required that each $s_i$ can only transmit coded symbols generated from its own information packets. That is, no INC is allowed at $s_i$ even if $s_i$ may overhear some packets from $s_j$ or when $s_j$ and $s_i$ are situated at the same physical node. On the other hand, we allow two destinations $d_i$ and $d_j$ to be situated at the same physical node, which is modelled by choosing the $p_{d_i} \rightarrow d_j$ and $p_{d_j} \rightarrow d_i$ parameters so that the success events for $d_i$ and $d_j$ are perfectly correlated with each other.
put feedback. Analytically, our preliminary feedback-oriented study [?] shows that in average the difference of the capacity region with $1_{\left(Z_{i-1}\right)\neq+}$ feedback and that of our 2-staged setting is less than 3% for $M=2$. In sum, the contribution of the 2-staged setting is two-fold: Firstly, it fully characterizes the benefits of MSI by decoupling it from the convoluted relationship with channel output feedback. Secondly, it serves as a good approximation of the existing COPE protocol. A unified treatment of MSI and feedback is likely to require a different set of analytical approaches (cf. [40] for the broadcast PEC capacity with feedback), which is beyond the scope of this work.

D. Non-Canonical Settings of 1-Hop Wireless Relay Networks

In a wireless network, it is possible that a packet transmitted wirelessly by $s_i$ may jump two hops and be heard directly by $d_j$ although this two-hop setting generally happens with a smaller probability than a 1-hop jump. See Fig. 1(c) for illustration. Recently, new opportunistic routing (OR) protocols, such as the MORE protocol [5], [23], take advantage of this spatial diversity and show that throughput can be drastically improved even without INC [5], [30].

The second commonly used technique to enhance the throughput of wireless networks is to perform cross-layer resource allocation [29] that allots different numbers of time slots $n_1$ and $n_2$, for the sources $s_i$ and the relay $r$, respectively, in order to maximize the achievable throughput (or sometimes maximize the utility instead).

Although our main results in Section III are based on the canonical setting of Section II-B, they can also be generalized to take into account both OR and cross-layer optimization. It turns out that introducing cross-layer optimization does not affect the tightness of the proposed outer and inner bounds (they still match for $M \leq 3$). However, incorporating OR does widen the gap, 1–2% numerically [?], between the outer and inner bounds. Detailed discussion of the non-canonical setting is relegated to Appendix H.

III. MAIN RESULTS

A. The Capacity Outer Bound For General $M$ Values

We first introduce a critical definition and a central proposition for deriving the capacity outer bound.

Definition 3: We use the convention that any function that is not decreasing is called an increasing function. With this convention, a function $f : [0, 1] \rightarrow \mathbb{R}^+$ is a ZPPLCIF if $f(\cdot)$ is a Zero-Passing (i.e., $f(0) = 0$), Piecewise Linear, Concave, Increasing Function. Note that for the following, whenever we say a function is a ZPPLCIF, we automatically imply that its domain is $[0, 1]$ and its range is non-negative reals $\mathbb{R}^+$. Consider a 1-to-3 PEC from a source $s$ with the input symbol denoted by $W \in \text{GF}(q)$ and the output symbols denoted by $Z_1$, $Z_2$, and $Z_3$, respectively. Recall that $p_{s;i} = \text{Prob}(Z_i = W)$, $i = 1, 2, 3$, is the marginal success probability and we use $p_1$, $p_2$, and $p_3$ as shorthand. Without loss of generality, assume that $p_1 > p_2 > p_3$, which can be achieved by relabeling the receivers.

Proposition 1 (Concavity of Mutual Information): For any $n$, suppose we use the memoryless and stationary PEC $n$ times and let $W$, $Z_1$, $Z_2$, and $Z_3$ denote the $n$-dimensional vectors of the input and output symbols, respectively. Let $U$ denote an auxiliary random variable such that we transmit $W = g(U)$ for some encoding function $g(\cdot)$ and $U \rightarrow W \rightarrow (Z_1, Z_2, Z_3)$ form a Markov chain. For any distribution $P_U$ and any encoding function $g(\cdot)$, we have

$$I(U; Z_1) \geq I(U; Z_2) \geq I(U; Z_3)$$

By repeatedly using the above proposition, one can also derive the following corollary:

Corollary 1: Consider a 1-to-$M$ PEC with $M$ different receivers $Z_1$ to $Z_M$. Without loss of generality, we add two auxiliary receivers $Z_0$ and $Z_{M+1}$ and assume that their marginals satisfy $p_0 = 1 > p_1 > \cdots > p_M > p_{M+1} = 0$. Let $U$ denote an auxiliary random variable such that we transmit $W = g(U)$ for some encoding function $g(\cdot)$ and $U \rightarrow W \rightarrow (Z_0, \cdots, Z_{M+1})$ form a Markov chain. For any distribution $P_U$ and any encoding function $g(\cdot)$, if we define $f : [0, 1] \rightarrow \mathbb{R}^+$ by $f(p_i) = I(U; Z_i)$, $\forall i \in \{0, \cdots, M+1\}$ and linearly interpolating the $f(p)$ value for those $p \in (p_{i-1}, p_i)$, $\forall i \in [M+1]$, then $f(\cdot)$ is a ZPPLCIF.

The proof of Proposition 1 is relegated to Appendix A.

Recall that $p_{r;i}$ denotes the marginal success probability from $r$ to $d_i$. Also recall that $p_{s;i,r}$ (resp. $p_{s;i,T}$) is the marginal probability that $d_j$ (resp. $r$) receives a packet sent by $s_i$, and $p_{s;,T}$ is the probability that a packet sent by $s_i$ is received by at least one node in $T$. The capacity outer bound can then be stated based on the concept of ZPPLCIFs.

Proposition 2: [Capacity Outer Bound] Without loss of generality, assume that $p_{r;i} > p_{r;i+1} > \cdots > p_{r;M} > 0$. A rate vector $(R_1, \cdots, R_M)$ is feasible only if there exist $2^M$ ZPPLCIFs $\{f_S(\cdot) : S \in 2^{[M]}\}$ indexed by $S \in 2^{[M]}$ such that the following $(M + M^22^{M-1} + 1)$ inequalities are satisfied:

$$\forall i \in [M], \ R_i \leq p_{s;i,r}$$

$$\forall i, k \in [M], S, T \in 2^{[M]}$$

$$\left\{f_S(\cdot) : S \in 2^{[M]}\right\}$$

$$\text{satifying } k \neq S \text{ and } T = S \cup \{k\},$$

$$\int_{f_S(p_{r;i})} \geq f_T(p_{r;i}) + 1_{\{i \leq k\}} \left(R_k - p_{s;i,T}^S\right)$$

and

$$f_S(p_{r;M}) \leq p_{r;M},$$

where $1_{\{\cdot\}}$ is the indicator function and $(\cdot)^+ \triangleq \max(\cdot, 0)$ is the projection to the non-negative reals.

We first provide a sketch of the proof, which sheds some insights on the above outer bound. A detailed proof of Proposition 2 is relegated to Appendix B.

Sketch of the proof: Following the 2-stage interpretation in Section II-C, let us focus on the 1-to-$M$ broadcast PEC used by $r$ in the second stage with MSI already created in the first stage. The key step is that for any given wireless 1-hop INC scheme, we consider the following normalized mutual information function with the the base of the logarithm being
q. That is, we choose the function \( f_S(\cdot) \) by setting
\[
 f_S(p_{r; i}) = \frac{I(U; Z_{r-i})}{n} \left( \{Z_{i-j}; \forall j \in S, \forall l \in [M] \setminus j \} \right),
\]
for all \( p_{r; i} \) and interpolating the \( f_S(p) \) value for all \( p \neq p_{r; i} \). The random variable \( U \) in (10) is defined by
\[
 U = \{X_i, Z_{i-j}, 1_{(Z_{j-i}(\cdot)) \neq \emptyset : \forall i, j \in [M], i \neq j} \}.
\]
Namely, \( U \) contains all information packets \( X_i \), all the packets that arrive at the relay \( r \), and all the channel status indicators that are fed back to \( r \). The \( Z_{r-i} \) in (10) denotes the \( n \)-dimensional \((r, d_i)\) PEC output vector, \( \{Z_{i-j}; \forall j \in S, \forall l \in [M] \setminus j \} \) is all the side information received by \( \{d_j; \forall j \in S\} \) in the first stage, and \( \{X_j; \forall j \in S\} \) is the information messages for session \( d_j, j \in S \). By Proposition 1, \( f_S(\cdot) \) must be a ZPPLCIF.

With the above construction of \( f_S(\cdot) \), for any \( k \notin S \) and \( T = S \cup \{k\} \), the following simplified version of (8)
\[
 f_S(p_{r;k}) \geq f_T(p_{r;k}) + (R_k - p_{s_k}; S)^+ \tag{11}
\]
can be interpreted as follows.

Let us temporarily allow a genie to grant \( d_k \) some additional information: \( \{Z_{l-j}; X_j : \forall j \in S, \forall l \in [M] \setminus j\} \). The message information desired by \( d_k \): The term \((R_k - p_{s_k}; S)^+\) quantifies the (minimum) amount of information symbols \( X_{k,l} \), \( l \in [nR_k] \) that are not received by any \( d_j \), \( j \in S \). Since \( k \notin S \), none of the above information symbols are included in the additional information \( \{Z_{l-j}; X_j : \forall j \in S, \forall l \in [M] \setminus j\} \) given to \( d_k \). As a result, \( d_k \) still desires this amount of information.

The total information observable by \( d_k \): By (10), the mutual information expression \( f_S(p_{r;k}) \) corresponds to the total amount of additional information that can still reach \( d_k \) even after \( d_k \) receiving the additional information \( \{Z_{l-j}; X_j : \forall j \in S, \forall l \in [M] \setminus j\} \). The undesired interference from the perspective of \( d_k \):

We also observe that
\[
 f_T(p_{r;k}) = \frac{I(U; Z_{r-k})}{n} \left( \{Z_{i-j}; X_j : \forall j \in S \cup \{k\}, \forall l \in [M] \setminus j \} \right)
\]
quantifies the amount of information that is not intended for \( d_k \) (since we condition on \( X_k \)) and cannot be canceled by the side information that is available at \( d_k \), which includes the overhead \( \{Z_{l-k}; \forall l \in [M] \setminus k\} \) and the information given by the genie \( \{Z_{l-j}; X_j : \forall j \in S, \forall l \in [M] \setminus j\} \). Therefore, \( f_T(p_{r;k}) \) can be viewed as the amount of interference that cannot be resolved by the side information available to \( d_k \).

Combining the above three observations: (11) is thus a restatement that the total amount of available mutual information must be no smaller than the amount of interference plus the amount of information that needs to be decoded.

Since by definition (10) \( f_\emptyset(p_{r; M}) = \frac{I(U; Z_{r-M})}{n} \), equation (9) is simply a restatement that the total mutual information at \( d_M \) cannot exceed the total number of successfully received time slots. Eq. (7) quantifies the maximal rate that \( s_i \) can convey to the relay node \( r \). A detailed proof can be completed following the above reasonings for (7) to (9).

Remark: The central idea of Proposition 2 is the interpretation of (11), a similar version of which was also used in the index-coding outer bound [Theorem 6 of [3]]. Proposition 2 can thus be viewed as a generalization of the index-coding outer bound for the wireless setting.

B. The Capacity Region For \( M \leq 3 \)

The tightness result of the outer bound in Proposition 2 is provided as follows.

Proposition 3: The capacity outer bound in Proposition 2 is indeed the capacity region for the cases of \( M \leq 3 \).

A detailed proof of Proposition 3 is provided in Sections V-A and V-B for \( M = 2 \) and \( 3 \), respectively.

C. The Capacity Inner Bound For General \( M \) Values

To quantify the tightness of the proposed outer bound for general \( M \geq 4 \), we provide the following capacity inner bound, which needs some new notation.

For any arbitrary permutation \( \pi \) on \([M]\), \( \pi : [M] \mapsto [M] \), we define a monotonically decreasing sequence of \((M + 1)\) sets \( S_\pi^j \) for \( j \in [M + 1] \) such that for each \( j \), we have \( S_\pi^j = \{j', j = j', \ldots, M\} \). By definition, \( S_\pi^0 = \emptyset \) and \( S_\pi^M = [M] \), respectively. Note that there are \( M! \) different distinct permutations \( \pi \) on \([M]\). For the capacity inner bound, we consider \((M + 1)!\) ZPPLCIFs \( \{f_\pi, S_\pi^j : \forall \pi, \forall j \in [M + 1]\} \), which are indexed by \( \pi \) and \( S_\pi^j \) jointly.

For example, if \( M = 2 \), we have two permutations \( \pi_1 = (1, 2) \) and \( \pi_2 = (2, 1) \). Thus \( S_\pi^1 = \{1, 2\} \), \( S_\pi^2 = \{\emptyset, S_\pi^0 = \emptyset \), \( S_\pi^3 = \emptyset \), \( S_\pi^2 = \{1\} \), \( S_\pi^1 = \emptyset \). We thus consider 6 ZPPLCIFs \( f_\pi_1, (1, 2), f_\pi_1, (2), f_\pi_2, \emptyset, f_\pi_2, (1, 2), f_\pi_3, (1) \), and \( f_\pi_3, \emptyset \).

We then describe an achievable rate region as follows.

Proposition 4 (Capacity Inner Bound): Following the same settings and definitions of Proposition 2, a rate vector \((R_1, \ldots, R_M)\) can be achieved by a linear network code if there exist \( M!(2^{M + 1} - M - 2) \) non-negative variables:

\[
 \left\{ y_{k,T,(\pi,S_\pi^j)} \geq 0 : \forall k \in [M], \forall T \in 2^{[M]}, \forall \pi, \forall j \in [M + 1], \right. \\
 \left. \text{satisfying } k \notin S_\pi^j \text{ and } T \subseteq S_\pi^j \right\} 
\]

and \( M! 2^{M-1} \) non-negative variables:

\[
 \left\{ z_{k,T} \geq 0 : \forall T \in 2^{[M]}, \forall k \in [M] \setminus T \right\} ,
\]

and \((M + 1)!\) ZPPLCIFs \( \{f_\pi, S_\pi^j : \forall \pi, \forall j \in [M + 1]\} \), such that jointly the following \((M + 1)!M^3 + M2^{M-1} + M2^{M-1} + M + 1)\) inequalities are satisfied:

\[
 \forall i \in [M], \quad R_i < p_{s_i} \\
 \forall \pi, \forall i, j \in [M], \quad f_{\pi, S_\pi^j}(p_{r;i}) \geq f_{\pi, S_\pi^j}(p_{r;i}) \\
 + 1 \{i \leq k, k \notin S_\pi^j \} \sum_{\forall T \in 2^{[M] \setminus T} \subseteq S_\pi^j} y_{k,T,(\pi,S_\pi^{j+1})} \tag{13}
\]
versus this case, we have four functions

\[ \forall k \in [M], \forall T \in 2^{[M]}, k \notin T, \sum_{\forall \pi, \forall j \in [M+1]; S^\pi_j \supseteq T \triangleleft \emptyset} y_{k,T,(\pi,S^\pi_j)} \geq z_{k,T} \]  

(14)

\[ \forall S \in 2^{[M]}, \forall k \in [M \setminus S], z_{k,S} \leq p_{s_k \rightarrow ((M \setminus S) \cup \{k\})} \]  

(15)

\[ \forall k \in [M], \sum_{\forall S \in 2^{[M]}; k \notin S} z_{k,S} \geq R_k \]  

(16)

\[ \sum_{\forall \pi} f_{\pi,0}(p_{\pi};M) < p_{r;M}. \]  

(17)

A detailed proof of Proposition 4 is provided in Section VI.

It is worth noting the similarity between Propositions 2 and 4. The main differences are: (i) In the inner bound, the newly introduced permutation \( \pi \) in (17) is used to perform time multiplexing when compared to (9); (ii) In the inner bound, the quantities \((R_k - p_{s_{1:1};S})^+\) are replaced by the redistribution

\[ \sum_{\forall k \in [M], \forall \pi} f_{\pi,0}(p_{\pi};M) < p_{r;M}. \]

inequalities (14) to (16), which are based on the \( y \) and \( z \) variables and on the probability mass function \( p_{s_k \rightarrow T}, \pi, r, T \); (iii) In the inner bound, when considering a fixed pair of \( S \) versus \( S_j \), multiple \( k \) values are considered in (13) for all \( k \notin S_j \). For comparison, for a fixed pair of \( S \) versus \( T \), only a single value \( k \) is considered. This is due to that for the achievability result, one needs to take care of decoding the information for all destinations \( d_k \). But for the outer bound result, one only needs to consider a single receiver for which decoding may fail.

IV. ALTERNATIVE FORMS OF THE OUTER BOUND

In this section, we present two equivalent forms of the outer bound of Proposition 2, the former of which is based on the rate vectors \((R_1, \cdots , R_M)\) only, while the latter of which is based on the feasibility of a linear programming problem.

A. A Rate-Based Outer Bound For \( M \leq 3 \)

We first consider the simplest case for which \( M = 2 \). In this case, we have four functions \( f_{\pi,0}(\cdot) \) to consider: \( f_{\{1,2\}}(\cdot), f_{\{1\}}(\cdot), f_{\{2\}}(\cdot), \) and \( f_{\emptyset}(\cdot) \). We discuss each function separately as follows.

Case 1: \( S = \{1, 2\} \). Since \( f_{\{1,2\}}(\cdot) \) is always on the smaller side of the inequalities in (8) and (9), we can simply set

\[ f_{\{1,2\}}(p) \overset{\Delta}{=} 0 \] for all \( p \in [0, 1] \).

Case 2: \( S = \{1\} \). (8) implies that

\[ f_{\{1\}}(p_{r;1}) \geq f_{\{1,2\}}(p_{r;1}) + (R_2 - p_{s_{2:1}})^+ = (R_2 - p_{s_{2:1}})^+. \]

Case 3: \( S = \{2\} \). (8) implies that

\[ f_{\{2\}}(p_{r;2}) \geq f_{\{1,2\}}(p_{r;2}) + (R_2 - p_{s_{2:1}})^+ = (R_2 - p_{s_{2:1}})^+. \]

Case 4: \( S = \emptyset \). Using \( S = \emptyset \) and \( T = \{1\} \), the discussion in Case 2 and (8) jointly imply that

\[ f_{\emptyset}(p_{r;1}) \geq f_{\{1\}}(p_{r;1}) + R_2 \geq R_1 + (R_2 - p_{s_{2:1}})^+ \]

(19)

Using \( S = \emptyset \) and \( T = \{2\} \), the discussion in Case 3 and (8) jointly imply that

\[ f_{\emptyset}(p_{r;2}) \geq f_{\{2\}}(p_{r;2}) + R_2 \geq R_1 + (R_1 - p_{s_{1:2}})^+ \]

(20)

By further taking into account of (7), we have

Corollary 2: For the case of \( M = 2 \), the outer bound of Proposition 2 has the following equivalent form:

\[ R_1 \leq \min \left( p_{s_{1:2}}, p_{r;1} - (R_2 - p_{s_{2:1}})^+ \right) \]

\[ R_2 \leq \min \left( p_{s_{2:2}}, p_{r;2} - \frac{p_{r;2}}{p_{r;1}} (R_1 - p_{s_{1:2}})^+ \right). \]

B. A Linear-Programming-Based Outer Bound For Arbitrary \( M \) Values

Although we can repeat the steps of the previous subsection and convert the outer bound in Proposition 2 to its rate-based counterpart for arbitrary \( M \) values, the description of the outer bound grows exponentially. On the other hand, a unique feature of the proposed outer bound is that it can be checked by any linear programming (LP) solver. To that end, we first describe the following lemma.

Lemma 1: For any ZPPLCIF \( f(\cdot) \) that changes its slope only at \( M \) points: \((p_i, f(p_i))\) for \( i \in [M] \) and \( p_1 > \cdots > \)
by explicitly constructing capacity-achieving linear network codes. Note that since our constructions focus on a sufficiently large block length \( n \) and a sufficient large finite field size \( \text{GF}(q) \), we can, without loss of generality, resort to the first-order analysis (the law of large numbers) and assume that random linear network coding (RLNC) [18] generates Maximum Distance Separable (MDS) codes with arbitrarily close-to-one probability.

A. The Capacity of \( M = 2 \)

For any \((R_1, R_2)\) in the interior of the outer bound in Corollary 2, we use the following linear INC scheme that has three major steps.

Step 1: For each time slot \( t \in [n] \), each source \( s_i, i \in \{1, 2\} \), sends out the coded symbol \( W_i(t) \) by RLNC, i.e.,

\[
W_i(t) = \sum_{k=1}^{nR_i} c_{i,k}(t)X_{i,k},
\]

for which we choose the mixing coefficients \( c_{i,k}(t) \) independently and uniformly randomly from \( \text{GF}(q) \). Oftentimes, we use \( c_i(t) = (c_{i,1}(t), \ldots, c_{i,nR_i}(t)) \) to denote the \((nR_i)\)-dimensional coding vector used in time \( t \). (29) can thus be simplified as \( W_i(t) = c_i(t)X_i^T \) where \( X_i^T \) is the transpose of the row vector \( X_i \). We also use \( \Omega_i = (\text{GF}(q))^{nR_i} \) to denote the full \((nR_i)\)-dimensional message space for the \( i \)-th user.

After Step 1, relay \( r \) receives \( nR_i \), coded packets, which is larger than \( nR_i \), the number of information packets. By the MDS property of RLNC, \( r \) can successfully decode \( X_{i,1} \) to \( X_{i,nR_i} \) for all \( i \in \{1, 2\} \).

Step 2: Each destination \( d_j \) sends to \( r \) its own overhearing channel status vector \( 1_{(Z_{i,j} - \star) \neq \star} \) for all distinct \( i, j \in \{1, 2\} \). Assuming that the seed used by \( s_i \) to generate the coded packets is also known to the relay \( r \), \( r \) thus knows the \( c_i(t) \) values for all \( i \in \{1, 2\} \) and \( t \in [n] \). Based on the knowledge of the channel status vectors \((1_{(Z_{i,j} - \star) \neq \star}) \), the relay \( r \) also derives

\[
\mathcal{V}_{1 \rightarrow 2} \triangleq \text{span} \left( \{c_1(t) : \forall t \text{ s.t. } Z_{1 \rightarrow 2}(t) \neq \star\} \right)
\]

That is, \( \mathcal{V}_{1 \rightarrow 2} \) is the linear subspace spanned by all coding vectors of the user-1 packets that are overheard by \( d_2 \). Again

\[
R_i \leq p_{s_i,r}, \forall i \in \{1, 2, 3\}
\]

\[
R_1 \leq p_{r_1} - \max \left( (R_2 - p_{s_2;1})^+ + (R_3 - p_{s_3;1,2})^+, (R_2 - p_{s_2;1,3})^+ + (R_3 - p_{s_3;1})^+ \right)
\]

\[
R_2 \leq p_{r_2} - \max \left( p_{r_2} p_{r_1}^{-1} \left( R_1 - p_{s_1;2,3} \right)^+ + (R_3 - p_{s_3;2})^+, \frac{p_{r_2}}{p_{r_1}} \left( R_1 - p_{s_1;2,3} \right)^+ + (R_3 - p_{s_3;1,2})^+ \right), \frac{p_{r_1} - p_{r_2}}{p_{r_1} - p_{r_3}} \left( R_1 - p_{s_1;2,3} \right)^+ + (R_3 - p_{s_3;2})^+ \right) + \frac{p_{r_2} - p_{r_3}}{p_{r_1} - p_{r_3}} \left( R_1 - p_{s_1;2,3} \right)^+ + (R_3 - p_{s_3;1,2})^+ \right)
\]

\[
R_3 \leq p_{r_3} - \max \left( p_{r_3} p_{r_1}^{-1} \left( R_1 - p_{s_1;3} \right)^+ + (R_2 - p_{s_2;1,3})^+, \frac{p_{r_3}}{p_{r_1}} \left( R_1 - p_{s_1;2,3} \right)^+ + \frac{p_{r_3}}{p_{r_2}} (R_2 - p_{s_2;3})^+ \right). \tag{27}
\]

Fig. 3. The capacity region for the case of \( M = 3 \).
by the MDS property of RLNC, the corresponding normalized rank of $V_{1\rightarrow 2}$ is

$$\text{Rk}_{1\rightarrow 2} \triangleq \frac{1}{n} \text{Rank}(V_{1\rightarrow 2}) = \min(R_1, p{s}_1; 2). \quad (30)$$

Before proceeding, we define the complement space operator $^\ominus$.

**Definition 4:** For any three linear subspaces $U$, $V$, and $W$, we say $W = U \ominus V$ is a complement space of $V$ with respect to $U$ if

$$\text{span}(V, W) = U$$

and $\text{Rank}(V) + \text{Rank}(W) = \text{Rank}(U)$.

The relay $r$ then constructs a complement space $V_{1\rightarrow 2} = \Omega_1 \ominus V_{1\rightarrow 2}$, i.e., $V_{1\rightarrow 2}$ is the linear subspace that is not overhead by $d_2$. By the property of complement spaces, the normalized rank of $V_{1\rightarrow 2}$ becomes

$$\text{Rk}_{1\rightarrow 2} \triangleq \frac{\text{Rank}(V_{1\rightarrow 2})}{n} = \frac{R_1}{n} - \frac{\text{Rank}(V_{1\rightarrow 2})}{n} = (R_1 - p{s}_1; 2)^{+}, \quad (31)$$

Let $v_k^{[1\rightarrow 2]}$, $k = 1, \ldots, n\text{Rk}_{1\rightarrow 2}$ and $v_l^{[1\rightarrow 2]}$, $l = 1, \ldots, n\text{Rk}_{1\rightarrow 2}$ denote the basis vectors of $V_{1\rightarrow 2}$ and $V_{1\rightarrow 2}$, respectively. We also define $V_{2\rightarrow 1}$, $V_{2\rightarrow 2}$, $V_{2\rightarrow 3}$, and $V_{2\rightarrow 4}$ by symmetry.

**Step 3:** The goal of this step is for the relay $r$ to construct the corresponding inter-session-coded packet $W_r(t)$ for $t \in [n]$. More explicitly, the inter-session-coded packet is constructed by

$$W_r(t) = \left( \sum_{k=1}^{\text{Rk}_{1\rightarrow 2}} c_{k_1}^{[1\rightarrow 2]}(t) v_{k_1}^{[1\rightarrow 2]} + \sum_{k=1}^{\text{Rk}_{1\rightarrow 2}} c_{k_2}^{[1\rightarrow 2]}(t) v_{k_2}^{[1\rightarrow 2]} \right) x_1^T + \left( \sum_{k=1}^{\text{Rk}_{1\rightarrow 2}} c_{k_3}^{[2\rightarrow 1]}(t) v_{k_3}^{[2\rightarrow 1]} + \sum_{k=1}^{\text{Rk}_{1\rightarrow 2}} c_{k_4}^{[2\rightarrow 1]}(t) v_{k_4}^{[2\rightarrow 1]} \right) x_2^T,$$

and our task is to design the coefficients $c_{k_1}^{[1\rightarrow 2]}(t)$, $c_{k_2}^{[1\rightarrow 2]}(t)$, $c_{k_3}^{[2\rightarrow 1]}(t)$, and $c_{k_4}^{[2\rightarrow 1]}(t)$. Namely, instead of working on the elementary basis, we now work on the new basis $v_{k_1}^{[1\rightarrow 2]}$, $v_{k_2}^{[1\rightarrow 2]}$, $v_{k_3}^{[2\rightarrow 1]}$, and $v_{k_4}^{[2\rightarrow 1]}$.

The design of the coefficients contains two phases. **Phase 1** lasts for $\frac{n\text{Rk}_{1\rightarrow 2}}{p_{r;1}}$ time slots. Namely, for all $t$ satisfying $0 < t \leq \frac{n\text{Rk}_{1\rightarrow 2}}{p_{r;1}}$, we choose the coefficients $c_{k_1}^{[1\rightarrow 2]}(t)$ independently and uniformly randomly from GF($q$) for all $k_1 \in [n\text{Rk}_{1\rightarrow 2}]$. For all other coefficients $c_{k_1}^{[1\rightarrow 2]}(t)$, $c_{k_2}^{[1\rightarrow 2]}(t)$, and $c_{k_4}^{[2\rightarrow 1]}(t)$, we set them to zero.

**Phase 2** lasts for

$$\max\left( \frac{n\text{Rk}_{1\rightarrow 2} + n\text{Rk}_{2\rightarrow T}}{p_{r;1}}, \frac{n\text{Rk}_{2\rightarrow 1} + n\text{Rk}_{2\rightarrow T}}{p_{r;2}} \right) \quad (32)$$

The complement subspace $U \ominus V$ is not unique, i.e., we may have $V_1 = U \ominus V$ and $V_2 = U \ominus V$ for two distinct subspaces $V_1$ and $V_2$. The correctness of our network code construction holds for any choice of the complement subspace.
of freedom \( n (R_{k_2-1} + R_{k_2-3}) \). By the MDS property of RLNC, it is guaranteed that \( d_2 \) can decode both subspaces \( \mathcal{V}_{2-1} \) and \( \mathcal{V}_{2-3} \), and thus decode its information symbols \( X_2 \). The above discussion proves the decodability of the proposed 2-phased scheme.

The remaining task is to show that the 2-phased scheme does not exceed the block length budget \( n \). That is, we need to show that

\[
\begin{align*}
    n & \geq \frac{n R_{k_1-3}}{p_{r,1}} + \\
    & \quad \max \left( \frac{n R_{k_1-2} + n R_{k_2-3}}{p_{r,1}}, \frac{n R_{k_2-1} + n R_{k_2-3}}{p_{r,2}} \right). 
\end{align*}
\]

(33)

By substituting the \( R_k \) variables in (33) with (30) and (31) and by noticing that the resulting (33) is a restatement of the outer bound in Corollary 2, the proof is thus complete.

Remark: In the decoding step, we assume that for any \( t \), the global encoding kernel [21] of \( W_r(t) \), also known as the global coding vector, is available at the destinations. This assumption generally is not true. However, we can use the same idea as in the generation-based network coding protocol [8] to circumvent this difficulty. Namely, we raise the finite field size from \( \text{GF}(q) \) to \( \text{GF}(q^K) \), and treat each \( \text{GF}(q^K) \) symbol as a vector of \( K \) symbols. Then we can use the first \( n (R_{1+1} + R_{2}) \) symbols of each vector to carry the \( (n (R_{1+1} + R_{2})) \)-dimensional global encoding kernel and repeatedly use the same kernel to encode the remaining \( (K-n (R_{1+1} + R_{2})) \) symbols. In this way, the cost of transmitting the kernel can be fully absorbed when we choose a sufficiently large \( K \). For simplicity, we will assume that the global encoding kernel is known to the destinations in our subsequent discussion of achievability.

B. The Capacity of \( M = 3 \)

For any \( (R_1, R_2, R_3) \) in the interior of the outer bound of Corollary 3, perform the following three steps.

Step 1: For each time slot \( t \in [n] \), each source \( s_i \), \( i \in \{1, 2, 3\} \), sends out the coded symbol \( W_i(t) \) by RLNC, i.e.,

\[
    W_i(t) = \sum_{k=1}^{n R_i} c_{i,k}(t) X_{i,k},
\]

for which we choose the mixing coefficients \( c_{i,k}(t) \) independently and uniformly randomly from \( \text{GF}(q) \). \( c_i(t) \triangleq (c_{i,1}(t), \ldots, c_{i,n R_i}(t)) \) denotes the corresponding \((n R_i)\)-dimensional coding vector used in time \( t \).

After Step 1, relay \( r \) receives \( n p_{s_i,r} \) coded packets, which is larger than \( n R_i \), the number of information packets. By the MDS property of RLNC, \( r \) can successfully decode \( X_{i,1} \) to \( X_{i,n R_i} \) for all \( i \in \{1, 2, 3\} \).

Step 2: Each destination \( d_j \), \( j \in \{1, 2, 3\} \), sends to \( r \) its own overhearing channel status vector \( 1_{\{Z_{i,j} \neq *\}} \) for \( i \neq j \). Based on the knowledge about the channel status vectors \( 1_{\{Z_{i,j} \neq *\}} \), the relay \( r \) derives the following linear subspaces \( \mathcal{V}_{i,j} \) for all \( i, j \in \{1, 2, 3\}, i \neq j \):

\[
    \mathcal{V}_{i,j} \triangleq \mathop{\text{span}} \{ (c_i(t) : \forall t \text{ s.t. } Z_{i,j}(t) \neq \ast) \}
\]

That is, \( \mathcal{V}_{i,j} \) is the linear subspace spanned by all the coding vectors of the user-\( i \) packets that are overheard by \( d_j \). For any distinct \( i, j_1, j_2 \in \{1, 2, 3\} \), define

\[
    \begin{align*}
        & \mathcal{V}_{i \rightarrow j, j_2} \triangleq \mathcal{V}_{i,j_1} \cap \mathcal{V}_{i,j_2} \\
        & \mathcal{V}_{i \rightarrow j_1 j_2} \triangleq \mathcal{V}_{i,j_1} \oplus \mathcal{V}_{i,j_2} \\
        & \mathcal{V}_{i \rightarrow j_2} \triangleq \mathcal{V}_{i:j_2} \\
        \end{align*}
\]

which correspond to the linear subspaces of user-\( i \) that are overheard by both \( j_1 \) and \( j_2 \), by \( j_1 \) only, by \( j_2 \) only, and by neither \( j_1 \) nor \( j_2 \), respectively. By the definition of complement spaces and by the MDS property of RLNC, the corresponding normalized ranks \( R_{k,\ldots} \) must satisfy:

\[
    \begin{align*}
        & R_{k \rightarrow j_1 j_2} + R_{k \rightarrow j_2} = \text{Rank}(\mathcal{V}_{i,j}) = \min(R_i, p_{s_i,j}) \\
        & R_{k \rightarrow j_1 j_2} + R_{k \rightarrow j_1} = \text{Rank}(\mathcal{V}_{i,j}) = \min(R_i, p_{s_i,j}) \\
        & R_{k \rightarrow j_1 j_2} + R_{k \rightarrow j_2} + R_{k \rightarrow j_1} = \text{Rank}(\mathcal{V}_{i,j}) = \min(R_i, p_{s_i,j} + \sum_{j \in \{1, 2\}} j) \\
        & R_{k \rightarrow j_1 j_2} + R_{k \rightarrow j_1} + R_{k \rightarrow j_2} = R_i. 
    \end{align*}
\]

Solving the above equations and using the following notation

\[
    \forall T' \in 2^{|M|}, \forall i \in [M] \setminus T, \quad \mathcal{R}_{i:T'} \triangleq (R_i - p_{s_i,T'})^+, \quad (34)
\]

we thus have

\[
    \begin{align*}
        & R_{k \rightarrow j_1 j_2} = (R_i - p_{s_i,j_1,j_2})^+ = \mathcal{R}_{i:(j_1,j_2)} \\
        & R_{k \rightarrow j_1 j_2} = (R_i - p_{s_i,j_1,j_2})^+ = \mathcal{R}_{i:(j_1,j_2)} \\
        & R_{k \rightarrow j_1 j_2} = (R_i - p_{s_i,j_1,j_2})^+ = \mathcal{R}_{i:(j_1,j_2)} \\
        & R_{k \rightarrow j_1 j_2} = R_i - \mathcal{R}_{i:(j_1,j_2)} + \mathcal{R}_{i:(j_1,j_2)} \quad (35)
    \end{align*}
\]

With the above definition of the subspaces, let \( v_k^{i \rightarrow T_1 T_2} \) denote the basis vectors of \( \mathcal{V}_{i \rightarrow T_1 T_2} \) for all \( T \subseteq \{1, 2\} \).

Step 3: Relay \( r \) constructs the following intersession-coded packet \( W_r(t) \) for all \( t \in [n] \):

\[
    W_r(t) = \sum_{i=1}^{3} \sum_{k=1}^{n R_{k \rightarrow s_i}} c_k^{i \rightarrow T_1} T_2(t) v_k^{i \rightarrow T_1 T_2} X_i^T
\]

based on the new basis vectors \( v_k^{i \rightarrow T_1 T_2} \). Our task is to design the corresponding coefficients \( c_k^{i \rightarrow T_1 T_2} \) for all \( t \in [n] \).
\[ \sum_{k=1}^{i-1} L_k < t \leq \sum_{k=1}^{i} L_k. \]

With different choices of \( i \), \( T_1 = \{1, 2, 3\} \) \( \backslash i \), and \( T_2 = \{1, 2, 3\} \) \( \backslash \{i\} \), we have 12 different types of \( V_{1-23} \) to consider. Similar to the discussion of the case \( M = 2 \), we can describe the coefficients by their corresponding \( n \times (n \sum_{i=1}^{3} R_i) \) matrix form. Namely, for any \( t \in [n] \), we treat the corresponding coefficients \( \{c_{k}^{\{i-T_1 T_2\}}(t): \forall i, T_1, T_2, k\} \) as a row vector and vertically concatenate all \( n \) row vectors to form a coding matrix.

Fig. 5 illustrates the high-level matrix-based description of a capacity-achieving network coding solution. For the following, we will explain in details how to construct the coding matrix in Fig. 5.

We first notice that in Fig. 5 the \( n R_{k_{1-23}} \) columns corresponding to \( V_{1-23} \) are further partitioned into two groups of \( \Delta_1 \) and \( \Delta_2 \) columns, respectively. The values of \( \Delta_1 \) and \( \Delta_2 \) will be described shortly. Similarly, the columns corresponding to \( V_{1-12} \) (resp. \( V_{3-12} \)) are partitioned into two groups of sizes \( (\Delta_3, \Delta_4) \) (resp. \( (\Delta_5, \Delta_6) \)).

We then set the durations \( L_1 = \frac{n R_{k_{1-23}}}{p_{r1}}, \) \( L_2 = \frac{\Delta_1 + \Delta_3}{p_{r1}} \),

\[ L_3 = n \cdot \max \left( \frac{R_{k_{2-12}} + R_{k_{2-12}}}{p_{r2}}, \frac{R_{k_{1-23}} + R_{k_{1-23}}}{p_{r1}} \right), \]

and \( L_4 = n - L_1 - L_2 - L_3 \). As a result, to fully describe the structure of the coding matrix, we only need to describe three parameters \( \Delta_1, \Delta_4, \) and \( \Delta_5 \) since \( \Delta_2 = n R_{k_{1-23}} - \Delta_1, \) \( \Delta_3 = n R_{k_{3-12}} - \Delta_4, \) and \( \Delta_6 = n R_{k_{3-12}} - \Delta_5 \). The choices of \( \Delta_1, \Delta_4, \) and \( \Delta_5 \) depend on the values of \( R_{k_{1-23}}, R_{k_{3-12}}, \) and \( R_{k_{3-12}} \). Totally, we have 4 different cases when deciding the values of \( \Delta_1, \Delta_4, \) and \( \Delta_5 \).

**Case 1:** \( R_{k_{3-12}} \geq R_{k_{1-23}} \). In this case, we set \( \Delta_1 = \Delta_4 = 0 \).

**Case 2:** \( R_{k_{1-23}} \geq R_{k_{3-12}} \geq \frac{p_{r1}}{p_{r3}} R_{k_{1-23}} \). In this case, we set

\[ \Delta_1 = n \frac{p_{r1} \left(R_{k_{1-23}} - R_{k_{3-12}}\right)}{p_{r1} - p_{r3}} \]

\[ \Delta_4 = n \frac{p_{r3} \left(R_{k_{1-23}} - R_{k_{3-12}}\right)}{p_{r1} - p_{r3}}, \] and \( \Delta_5 = 0 \). (39)

**Case 3:** \( \frac{p_{r3}}{p_{r1}} R_{k_{1-23}} \geq R_{k_{3-12}} \) and

\[ R_{k_{3-12}} + R_{k_{3-12}} \geq \frac{p_{r3}}{p_{r1}} \left(R_{k_{1-23}} + R_{k_{3-12}}\right). \] (40)

In this case, we set

\[ \Delta_1 = n R_{k_{1-23}}, \ \Delta_4 = n R_{k_{3-12}}, \] and \( \Delta_5 = n \frac{p_{r3} R_{k_{1-23}} - p_{r1} R_{k_{3-12}}}{p_{r1} - p_{r3}}. \) (41)

**Case 4:** \( \frac{p_{r3}}{p_{r1}} \left(R_{k_{1-23}} + R_{k_{3-12}}\right) \geq R_{k_{3-12}} + R_{k_{3-12}} \). In this case, we set \( \Delta_1 = n R_{k_{1-23}}, \Delta_4 = n R_{k_{3-12}} \) and \( \Delta_5 = n R_{k_{3-12}} \).

The following lemma ensures the legitimacy of the above choices of \( \Delta_1, \Delta_4, \) and \( \Delta_5 \).

**Lemma 4:** For any \( R_{k_{1, R_{k_{2, R_{k_{3}}}}}} \) in the interior of the outer bound of Corollary 3, all \( \Delta_1 \) to \( \Delta_6 \) are non-negative and \( L_4 \) is also non-negative. Moreover, the following inequality always holds.

\[ \frac{\Delta_1 + \Delta_5}{p_{r1}} \geq \frac{\Delta_4 + \Delta_5}{p_{r3}}. \] (42)

To prove this lemma, we need to use the assumption that \( R_{k_{1, R_{k_{2, R_{k_{3}}}}}} \) are in the interior of (25) to (27). A detailed proof is relegated to Appendix D.

With the structure of the coding matrix fixed, we now describe how to choose the entries of the submatrices. In Fig. 5, we use “0” to denote the zero submatrix in the overall coding matrix. For example, in Phases 2 to 4, all \( c_{k}^{\{i-T_1 T_2\}}(t) \), \( k = 1, \cdots, n R_{k_{1-23}} \) are set to zero. Similarly, in Phases 1, 3, and 4, the coefficients \( c_{k}^{\{\bar{T}_1 T_2\}}(t) \) of the first \( \Delta_1 \) columns of \( V_{1-23} \), i.e., \( k = 1, \cdots, \Delta_1 \), are set to zero.

We now consider the shaded boxes in Fig. 5 that are not in the columns corresponding to \( V_{1-23}, V_{3-12} \), the \( \Delta_2 \) portion of \( V_{1-23} \), and the \( \Delta_3 \) portion of \( V_{3-12} \). Namely, consider those boxes in solid colors. For those submatrices, we choose the corresponding coefficient \( c_{k}^{\{i-T_1 T_2\}}(t) \) independently and uniformly randomly from \( GF(q) \). For example, in Phase 3, we choose \( c_{k}^{\{i-T_1 T_2\}}(t) \), \( k = 1, \cdots, n R_{k_{1-23}} \) independently and uniformly randomly from \( GF(q) \). Similarly, in Phase 2,
we choose the coefficients $c_k^{[3−12]}(t)$ of the last $\Delta_4$ columns of $\mathcal{V}_{3−12}$, i.e., $k = (nRk_{3−12} − \Delta_4) + 1, \ldots, nRk_{3−12}$, independently and uniformly randomly from $GF(q)$.

The construction of the four groups of columns (i) the last $\Delta_2$ columns in $\mathcal{V}_{1−3}$, (ii) $\mathcal{V}_{2−3}$, (iii) $\mathcal{V}_{3−2}$, and (iv) the first $\Delta_3$ columns in $\mathcal{V}_{3−12}$ is special, and we use the code alignment technique, which was also used in [38], [40], [43] and is in parallel with the interference alignment technique for additive Gaussian channels [4].

Using the code alignment technique, we first construct an $n \times \max(\Delta_2, \Delta_3)$ matrix $A$ by randomly generating each entry of $A$ independently and uniformly randomly from $GF(q)$. We then copy the first $\Delta_2$ columns of $A$ to the last $\Delta_2$ columns of $\mathcal{V}_{1−3}$, and copy the first $\Delta_3$ columns of $A$ to the first $\Delta_3$ columns of $\mathcal{V}_{3−12}$. Note that with this construction, the last $\Delta_2$ columns of $\mathcal{V}_{1−3}$ and $\mathcal{V}_{3−12}$ are independently and uniformly randomly distributed, and so are the first $\Delta_3$ columns of $\mathcal{V}_{3−12}$. On the other hand, jointly these columns are highly correlated as they are the overlapped copies of the common matrix $A$. More explicitly, for all $t \in [n]$ and for all $1 \leq k \leq \min(\Delta_2, \Delta_3)$, we have

$$c_k^{[1−2]}(t) = c_k^{[3−12]}(t). \quad (43)$$

Since the entries of these two groups of columns are perfectly aligned, it gives the name code alignment.

Similarly, code alignment is also applied to columns of $\mathcal{V}_{2−3}$ and $\mathcal{V}_{3−2}$. Namely, we construct an $n \times \max(Rk_{2−3}, Rk_{3−2})$ matrix $B$ by randomly generating each entry of $B$ independently and uniformly randomly from $GF(q)$. We then copy the first $nRk_{2−3}$ columns of $B$ to the $nRk_{2−3}$ columns of $\mathcal{V}_{2−3}$, and copy the first $nRk_{3−2}$ columns of $B$ to the $nRk_{3−2}$ columns of $\mathcal{V}_{3−2}$. For all $t \in [n]$ and for all $1 \leq k \leq n \cdot \min(Rk_{2−3}, Rk_{3−2})$, we thus have

$$c_k^{[2−3]}(t) = c_k^{[3−2]}(t). \quad (44)$$

In Fig. 5, the code aligned submatrices are illustrated with linear hatching. The description of the entire network code is thus complete.

The case of $d_2$: We first consider the Phase-3 and Phase-4 packets received by $d_1$, which form a $(p_{r;1}(L_3 + L_4)) \times \left(\sum_{i=1}^{3} R_i\right)$ matrix. We rearrange the columns so that the columns corresponding to $\mathcal{V}_{1−23}$ and $\mathcal{V}_{2−23}$ are the first $n \cdot (Rk_{1−23} + Rk_{2−23})$ columns. We call these columns the Group-1 columns and use $nC_{G1} \triangleq n \cdot (Rk_{1−23} + Rk_{2−23})$ to denote the number of Group-1 columns. We also rearrange the rows so that the Phase-3 packets correspond to the first $p_{r;1}L_3$ rows while the Phase-4 packets correspond to the last $p_{r;1}L_4$ rows. After rearrangement, the coding matrix becomes

$$
\begin{bmatrix}
A_1 & A_2 \\
0 & A_3
\end{bmatrix},
$$

where the four corners correspond to the intersections of the Phase-3/Phase-4 rows (packets) and the Group-1/non-Group-1 columns.

We now consider the “degrees of freedom” in terms of the ranks of unknown spaces. For Group-1 columns, the corresponding linear spaces $\mathcal{V}_{1−27}$ and $\mathcal{V}_{2−23}$ are unknown spaces from the perspective of $d_1$. Totally, they contribute to $nC_{G1}$ degrees of freedom. We then turn our focus to the non-Group-1 columns. We first notice that in our construction of Fig. 5, the columns corresponding to $\mathcal{V}_{1−27}$, the $\Delta_1$ portion of $\mathcal{V}_{1−3}$, the $\Delta_4$ portion of $\mathcal{V}_{3−17}$, and the $\Delta_5$ portion of $\mathcal{V}_{3−17}$ must be all-zero. Moreover, from $d_1$‘s perspective, the subspaces $\mathcal{V}_{2−13}$, $\mathcal{V}_{2−12}$, $\mathcal{V}_{3−12}$, and $\mathcal{V}_{1−12}$ are transparent since $d_1$ can decode these subspaces by the side information $Z_{2−1}$ and $Z_{3−1}$. As a result, from a decoding’s perspective, among all non-Group-1 columns, only the following subspaces are unknown to $d_1$ and participate in the submatrices $A_A$ and $A_3$: $\mathcal{V}_{1−23}, \mathcal{V}_{2−3}, \mathcal{V}_{3−2}, \mathcal{V}_{1−3}$, the $\Delta_2$ portion of $\mathcal{V}_{1−3}$, and the $\Delta_6$ portion of $\mathcal{V}_{3−17}$. Totally, there are thus

$$nRk_{1−23} + nRk_{2−3} + nRk_{3−2} + \Delta_2 + \Delta_6 \quad (46)$$
degrees of freedom for the unknown subspaces corresponding to the non-Group-1 columns.

On the other hand, since we use the code alignment technique to construct the coding matrix, see (44), the contribution of the $v_k^{[2−3]}$ and $v_k^{[3−2]}$ basis vectors must satisfy

$$
\begin{align*}
&\sum_{k_1=1}^{nRk_{1−23}} c_{k_1}^{[2−3]}(t) v_k^{[2−3]} X_2^T + \sum_{k_2=1}^{nRk_{2−3}} c_{k_2}^{[3−2]}(t) v_k^{[3−2]} X_3^T \\
&\quad = \sum_{k=1}^{n\cdot \min(Rk_{1−23}, Rk_{2−3})} c_k^{[2−3]}(t) \left( v_k^{[2−3]} X_2^T + v_k^{[3−2]} X_3^T \right) \\
&\quad + \sum_{k_1=n\cdot \min(Rk_{1−23}, Rk_{2−3})+1}^{nRk_{3−2}} c_{k_1}^{[2−3]}(t) v_{k_1}^{[2−3]} X_2^T \\
&\quad + \sum_{k_2=n\cdot \min(Rk_{2−3}, Rk_{3−2})+1}^{nRk_{3−2}} c_{k_2}^{[2−3]}(t) v_{k_2}^{[2−3]} X_3^T.
\end{align*}
\quad (47)
$$

Therefore, from the perspective of decoding at $d_1$, the effective degrees of freedom of the undesired spaces $\mathcal{V}_{2−3}$ and $\mathcal{V}_{3−2}$ is reduced by the code alignment technique from the original $nRk_{2−3} + nRk_{3−2}$ to $n \cdot \max(Rk_{1−23}, Rk_{2−3})$. Namely, instead of decoding $v_k^{[2−3]} X_2^T$ and $v_k^{[3−2]} X_3^T$ individually, we simply need to decode the “aligned interference” $(v_k^{[2−3]} X_2^T + v_k^{[3−2]} X_3^T)$ for $k = 1, \ldots, n \cdot \min(Rk_{1−23}, Rk_{2−3})$, as expressed in (47). Based on the above reasoning, the effective degrees of freedom of all non-Group-1 columns are reduced from (46) to

$$nRk_{1−23} + n \cdot \max(Rk_{1−23}, Rk_{2−3}) + \Delta_2 + \Delta_6. \quad (48)$$

As a result, by the MDS property of RLNC, if we can prove

$$nC_{G1} \leq p_{r;1}L_3 \quad (49)$$

and $nC_{G1} + (48) \leq p_{r;1}(L_3 + L_4), \quad (50)$

where $p_{r;1}L_3$ and $p_{r;1}L_4$ are the numbers of Phase-3 and Phase-4 packets received by $d_1$, respectively, then we can solve the linear equations described in (45) and decode the spaces of $\mathcal{V}_{1−27}$, $\mathcal{V}_{2−23}$, $\mathcal{V}_{1−23}$, the $\Delta_2$ portion of $\mathcal{V}_{1−3}$, the $\Delta_6$
portion of $\mathcal{V}_{9-12}$, and the aligned interfering space of $\mathcal{V}_{2-13}$ and $\mathcal{V}_{3-12}$. To that end, we first notice that (49) is satisfied by the construction of $L_3$ (see Fig. 5). Eq. (50) can be proven by substituting (35) to (38) into (48) and by (25) of Corollary 3.

We now turn our attention to the Phase-1 and Phase-2 packets received by $d_1$. There are totally $p_{r;1}(L_1 + L_2)$ such packets. We observe that (i) The side information subspaces $\mathcal{V}_{2-13}$, $\mathcal{V}_{2-13}$, and $\mathcal{V}_{1-12}$ are transparent to $d_1$; (ii) According to the construction in Fig. 5, for Phases-1 and -2 packets, the columns corresponding to $\mathcal{V}_{1-27}$, $\mathcal{V}_{2-13}$, and $\mathcal{V}_{2-13}$ must be all-zero; (iii) The following subspaces have been decoded by $d_1$ using the Phases-3 and -4 packets and thus are transparent to $d_1$ as well, including $\mathcal{V}_{1-23}$, the code-aligned space $(\mathcal{V}_{2-13}, \mathcal{V}_{3-12})$, the $\Delta_2$ portion of $\mathcal{V}_{1-23}$, and the $\Delta_5$ portion of $\mathcal{V}_{3-12}$. By omitting the above three types of columns that are transparent to $d_1$, the remaining coding matrix for the Phases-1 and -2 packets can be expressed as

$$
\begin{pmatrix}
A_4 & 0 & 0 \\
0 & A_5 & A_6
\end{pmatrix},
$$

where $A_4$ corresponds to the intersection of the Phase-1 packets and the columns of $\mathcal{V}_{1-27}$, $A_5$ corresponds to the intersection of the Phase-2 packets and the columns of the $\Delta_1$ portion of $\mathcal{V}_{1-23}$, and $A_6$ corresponds to the intersection of the Phase-2 packets and the columns of the $\Delta_5$ portion of $\mathcal{V}_{3-12}$. Note that all the entries of $A_4$ to $A_6$ are chosen independently and uniformly randomly from $\text{GF}(q)$. By the MDS property of RLNC, if we can prove

$$
n \cdot \text{RK}_{1-23} \leq p_{r;1}L_1
$$

and

$$
\Delta_1 + \Delta_5 \leq p_{r;1}L_2,
$$

then we can solve the linear equations described in (51) and decode the spaces of $\mathcal{V}_{1-27}$, the $\Delta_1$ portion of $\mathcal{V}_{1-23}$, and the $\Delta_5$ portion of $\mathcal{V}_{3-12}$. We then notice that (52) and (53) are satisfied by the construction of $L_1$ and $L_2$, respectively, see Fig. 5.

Based on the above discussion, $d_1$ can decode all its desired information subspaces.

**The case of $d_2$:** Decoding at $d_2$ only uses the Phases-3 and -4 packets received by $d_2$, which form a $(p_{r;2}(L_3 + L_4)) \times (n \sum_{i=1}^3 R_i)$ matrix. We rearrange the columns so that the columns corresponding to $\mathcal{V}_{2-13}$ and $\mathcal{V}_{2-13}$ are the first $n \cdot \text{RK}_{2-13} + \text{RK}_{2-13}$ columns. We call these columns the Group-2 columns and use $n_{G2} \triangleq n \cdot \text{RK}_{2-13} + \text{RK}_{2-13}$ to denote the number of Group-2 columns. We also rearrange the rows so that the Phase-3 packets correspond to the first $p_{r;2}L_3$ rows while the Phase-4 packets correspond to the last $p_{r;2}L_4$ rows. After rearrangement, the coding matrix becomes

$$
\begin{pmatrix}
A_7 & A_8 \\
0 & A_9
\end{pmatrix},
$$

where the four corners correspond to the intersections of the Phase-3/Phase-4 rows (packets) and the Group-2/non-Group-2 columns.

Similar to the case of $d_1$, we consider the degrees of freedom of unknown spaces. For Group-2 columns, the corresponding linear spaces $\mathcal{V}_{2-13}$ and $\mathcal{V}_{2-13}$ contribute to $n_{G2}$ degrees of freedom. We then turn our focus to the non-Group-2 columns. Again we can exclude from our consideration the columns that do not participate in Phases 3 and 4 and those transparent columns corresponding to the overhead side information $Z_{1-2}$ and $Z_{3-2}$. As a result, from a decoding’s perspective, among all non-Group-2 columns, only the following subspaces are unknown to $d_2$ and participate in the submatrices $A_8$ and $A_9$: $\mathcal{V}_{2-13}$, $\Delta_2$ portion of $\mathcal{V}_{1-23}$, $\Delta_3$ portion of $\mathcal{V}_{1-23}$, and $\Delta_6$ portion of $\mathcal{V}_{3-13}$. Totally, there are thus

$$
n_{R_{k-23}} + n_{R_{k-13}} + \Delta_2 + \Delta_3 + \Delta_6
$$

degrees of freedom for the subspaces corresponding to the non-Group-2 columns.

Since we use the code alignment technique to construct the coding matrix, see (43), the last $\Delta_2$ basis vectors of $\mathcal{V}_{1-23}$, $v_{[1-23]}$, and the first $\Delta_3$ basis vectors of $\mathcal{V}_{3-13}$, $v_{[3-13]}$, are aligned. The effective degrees of freedom of all non-Group-2 columns are thus reduced from (55) to

$$
n_{R_{k-23}} + n_{R_{k-13}} + \text{max}(\Delta_2, \Delta_3) + \Delta_6.
$$

As a result, by the MDS property of RLNC, if we can prove

$$
n_{G2} \leq p_{r;2}L_3
$$

and

$$
n_{G2} + (56) \leq p_{r;2}(L_3 + L_4),
$$

then we can solve the linear equations described in (54) and $d_2$ can successfully decode its desired subspaces: $\mathcal{V}_{2-13}$, $\mathcal{V}_{2-13}$, $\mathcal{V}_{2-13}$, $\Delta_2$ of $\mathcal{V}_{1-23}$, $\Delta_3$ of $\mathcal{V}_{3-13}$, and $\Delta_6$ of $\mathcal{V}_{3-13}$.

To that end, we first notice that (57) is satisfied by the construction of $L_3$ (see Fig. 5). A detailed proof of (58) is provided in Appendix E, which uses the assumption that (26) is satisfied.

**Based on the above discussion, $d_2$ can decode all its desired information subspaces.**

**The case of $d_3$:** Decoding at $d_3$ only uses the Phase-2 and Phase-4 packets received by $d_3$, which form a $(p_{r;3}(L_2 + L_4)) \times (n \sum_{i=1}^3 R_i)$ matrix. We rearrange the columns so that the columns corresponding to the last $\Delta_4$ columns of $\mathcal{V}_{1-13}$ and the first $\Delta_5$ columns of $\mathcal{V}_{3-13}$ are moved to the first $\Delta_4 + \Delta_5$ columns. We call these columns the Group-3 columns and use $n_{G3} \triangleq \Delta_4 + \Delta_5$ to denote the number of Group-3 columns. We also rearrange the rows so that the Phase-2 packets correspond to the first $p_{r;3}L_2$ rows while the Phase-4 packets correspond to the last $p_{r;3}L_4$ rows. After rearrangement, the coding matrix becomes

$$
\begin{pmatrix}
A_{10} & A_{11} \\
0 & A_{12}
\end{pmatrix},
$$

where the four corners correspond to the intersections of the Phase-2/Phase-4 rows (packets) and the Group-3/non-Group-3 columns.

Similar to the cases of $d_1$ and $d_2$, we consider the degrees of freedom in terms of the ranks of unknown spaces. Specifically, Group-3 columns contribute to $n_{G3}$ degrees of freedom. We then turn our focus to the non-Group-3 columns. By similar
arguments as used in the cases of \( d_1 \) and \( d_2 \), among all non-Group-3 columns, only the following subspaces are unknown to \( d_3 \) and participate in the submatrices \( \mathbf{A}_{11} \) and \( \mathbf{A}_{12} \): \( V_{3 \rightarrow T_2} \), \( V_{3 \rightarrow T_2} \), the \( \Delta_3 \) portion of \( V_{3 \rightarrow T_1} \), and the \( \Delta_6 \) portion of \( V_{3 \rightarrow T_2} \). Totally, there are thus

\[
nR_{\Delta_3} + nR_{\Delta_6} + \Delta_3 + \Delta_6
\]

(60)
degrees of freedom for the subspaces corresponding to the non-Group-3 columns.

As a result, by the MDS property of RLNC, if we can prove

\[
nG_3 \leq p_{r,3}L_2
\]

(61)

and

\[
nG_3 + (60) \leq p_{r,3}(L_2 + L_4),
\]

(62)

then we can solve the linear equations described in (59) and \( d_3 \) can successfully decode its desired subspaces: \( V_{3 \rightarrow T_1} \), \( V_{3 \rightarrow T_2} \), \( V_{3 \rightarrow T_2} \), and \( V_{3 \rightarrow T_2} \).

To that end, we first notice that Lemma 4 implies (61). Eq. (62) can be proven by substituting (35) to (38) into (60) and by (27) of Corollary 3. Based on the above discussion, \( d_3 \) can decode all its desired information subspaces.

The proof of Proposition 3 is thus complete.

VI. THE ACHIEVABILITY RESULTS FOR ARBITRARY \( M \) VALUES

The capacity-achieving network code in the previous section is deliberately designed for the \( M = 3 \) case. Such an optimal design is difficult to generalize for the cases of \( M \geq 4 \). In this section, we provide a slightly suboptimal but systematic code design that attains the capacity inner bound of Proposition 4.

A. High-Level Description of The Achievability Scheme

In this subsection, we first present the skeleton of our code construction while the detailed description of a central step is relegated to Section VI-B.

Before proceeding, we first convert Proposition 4 to an equivalent LP problem by the alternative formulation of a ZPPLCIF in Lemma 1.

Proposition 4: Following the same settings and definitions of Proposition 4, a rate vector \( (R_1, \cdots, R_M) \) can be achieved by a linear network code if there exist \( M(M + 1)! \) non-negative variables \( \{x_{\pi, S_j^T, h} : \forall \pi, \forall j \in [M + 1], \forall h \in [M]\} \) and \( M(2^{M+1} - M - 2) \) non-negative variables:

\[
\{y_{k, T, (\pi, S_j^T)} : \forall k \in [M], \forall T \in 2^{[M]}, \forall \pi, \forall j \in [M + 1],
\]

satisfying \( k \notin S_j^T \) and \( T \subseteq S_j^T \},

and \( M2^{M-1} \) non-negative variables:

\[
\{z_{k, T} \geq 0 : \forall T \in 2^{[M]}, \forall k \in [M]\} - \{T\},
\]

such that jointly the rate \( R_k \) and the \( x, y, z \) variables satisfy (12), (14), (15), (16), and the following inequalities:

\[
\forall \pi, \forall i, j, k \in [M],
\]

\[
\sum_{h=1}^{M} x_{\pi, S_j^T, h} \cdot \min(p_{r,i}, p_{r,h}) \geq \sum_{h=1}^{M} y_{k, T, (\pi, S_j^T)} \cdot \min(p_{r,i}, p_{r,h})
\]

(63)

\[
\sum_{\forall \pi} \sum_{h=1}^{M} x_{\pi, g, h} \cdot p_{r,M} < p_{r,M} \iff \sum_{\forall \pi} \sum_{h=1}^{M} x_{\pi, g, h} < 1.
\]

(64)

Our construction scheme is based on the above alternative formulation of the inner bound.

For any rate vector \( (R_1, \cdots, R_M) \) satisfies the inner bound of Proposition 6, perform the following four steps.

Step 1: For each time slot \( t \in [n] \), each source \( s_i, i \in [M] \), sends out the coded symbol \( W_i(t) \) by RLNC, i.e.,

\[
W_i(t) = \sum_{k=1}^{nR_i} c_{i,k}(t)X_{i,k},
\]

for which we choose the mixing coefficients \( c_{i,k}(t) \) independently and uniformly randomly from \( GF(q) \). \( c_i(t) \triangleq (c_{i,1}(t), \cdots, c_{i,nR_i}(t)) \) denotes the corresponding \( (nR_i) \)-dimensional coding vector used in time \( t \).

After Step 1, relay \( r \) receives \( n_{R_{u,r}} \) coded packets, which is larger than \( nR_r \), the number of information packets. By the MDS property of RLNC, \( r \) can successfully decode \( X_{i,1} \) to \( X_{i,nR_i} \) for all \( i \in [M] \).

Step 2: Each destination \( d_j, j \in [M], \) sends to \( r \) its own overhearing channel status vector \( 1_{(Z_{i,i}(t) \neq *)} \) for \( i \neq j \). Based on \( 1_{(Z_{i,i}(t) \neq *)} \), the relay \( r \) derives the following linear subspaces \( V_{i,j} \) for all \( i, j \in [M], i \neq j \):

\[
V_{i,j} \triangleq \text{span}\{(c_i(t) : \forall t \text{ s.t. } Z_{i,j}(t) \neq *)\}
\]

That is, \( V_{i,j} \) is the linear subspace spanned by all the coding vectors of the user-\( i \) packets that are overheard by \( d_j \). Based on \( \{V_{i, i} : \forall i \neq j\} \), the relay \( r \) further derives multiple subspaces \( V_{i, T_1 T_2} \) for all \( i \in [M], T_1 \subseteq [M]\} i, \) and \( T_2 = [M]\} \{i \} \) such that the following three properties are satisfied:

(i) \( \forall j \in T_1, \quad V_{i, T_1 T_2} \subseteq V_{i,j}, \)

ii) \( \text{span}\{V_{i, T_1 T_2} : \forall T_1 \subseteq [M]\} i, \) and \( T_2 = [M]\} \{i \} \} \) equals to the feasible solution \( z_{i,T_2} \) satisfying (15) and (16) of Propositions 4 and 6. Eqs. (15) and (16) can thus be viewed as the space partitioning inequalities.

Lemma 5: Construction of \( \{V_{i, T_1 T_2}\} \) is always possible.

A proof of Lemma 5 is provided in Appendix F.
With the above construction, let \( \mathbf{v}_k^{[i-T_1, T_2]} \), \( k = 1, \ldots, n \cdot z_{i, T_2} \) denote the basis vectors of \( \mathcal{V}_{i-T_1, T_2} \).

**Step 3:** We then partition the \( n \) time slots into \( M! \) subgroups. Each group, indexed by a permutation \( \pi \), contains \( n \cdot (\sum_{h=1}^{M} x_{\pi, h, h}) \) time slots. Inequality (64) ensures that such time slot partition is always possible. We will perform network coding for each subgroup separately.

Note that partitioning in time is equivalent to partitioning the rows of the coding matrix. We now turn our focus to partitioning the basis vectors (the columns). With the above construction, let \( \mathbf{v}_k^{[i-T_1, T_2]} \), \( k = 1, \ldots, n \cdot z_{i, T_2} \). To that end, we rely on new \( \nu \) variables such that

\[
\forall T_2 \text{ satisfying } i \notin T_2, \quad \sum_{\pi} z_{i, T_2, \pi} = z_{i, T_2},
\]

(65)

On the other hand, any vector \( \mathbf{v} \in \mathcal{V}_{i-T_1, T_2} \) can also be encoded as if it has only been overheard by users in a subset \( S_1 \subseteq T_1 \), i.e., we forgo the fact that \( \mathbf{v} \) is also known to all \( j \in T_1 \setminus S_1 \). As a result, by defining \( S_2 = [M] \setminus (S_1 \cup \{i\}) \), (65) can be revised as follows.

\[
\forall T_2 \text{ satisfying } i \notin T_2, \quad \sum_{\pi, S_2 \supseteq T_2, i \notin S_2} y'_{i, T_2, (\pi, S_2)} = z_{i, T_2},
\]

(66)

where \( y'_{i, T_2, (\pi, S_2)} \) takes into account the redistribution of \( \mathbf{v}_k^{[i-T_1, T_2]} \) vectors into partitions indexed by \( (\pi, S_2) \).

To facilitate the actual encoding operation in the subsequent Step 4, we do not allow full degrees of freedom when redistributing the \( \mathbf{v}_k^{[i-T_1, T_2]} \) vectors. More explicitly, we only allow \( y'_{i, T_2, (\pi, S_2)} \) to be non-zero when \( S_2 \) is of the form \( S_2 = S^T_j \) for some \( j \in [M+1] \) and \( i \notin S^T_j \). Combining this restriction with (66), the final form of the redistribution equation becomes

\[
\forall T_2 \text{ satisfying } i \notin T_2, \quad \sum_{\forall \pi, \forall j \in [M+1]: S^T_j \supseteq T_2, i \notin S^T_j} y_{i, T_2, (\pi, S^T_j)} = z_{i, T_2},
\]

(67)

which is a restatement of (14). Eq. (14) is thus a subspace redistribution inequality.

**Step 4:** Following Step 3, for any permutation \( \pi \), the goal of Step 4 is to encode all the partitioned basis vectors, totally

\[
\sum_{\forall \pi \in [M], j \in [M+1], T \subseteq [2M]: \exists \pi, T^\pi \leq S^T_j} n \cdot y_{i, T, (\pi, S^T_j)}
\]

of them, using the allocated time slots \( n \cdot (\sum_{h=1}^{M} x_{\pi, h, h}). \)

Step 4 is the central step of the construction, which will be elaborated in Section VI-B. We conclude this subsection by introducing the concept of “chunks of time slots” such that for each \( \pi \), we can partition the group of totally \( \sum_{h=1}^{M} x_{\pi, h, h} \) time slots into multiple chunks of time slots. Each chunk has

\[n \cdot \chi_{\pi, l} \] time slots for a distinct \( l \). The detailed properties of the “chunks” are described by the following lemma and corollary.

**Lemma 6:** Without loss of generality, for any \( x, y, \) and \( z \) values satisfying (63), (64), (12), (14), (15), and (16) of Proposition 6, we can assume that

- \( \forall \pi, \forall h \in [M], x_{\pi, [M], h} = 0. \)
- \( \forall \pi, \forall k \in [M], y_{j, \pi} \in [M], \)

\[
\sum_{h=1}^{k} x_{\pi, S^T_{h-1}, h} \geq \sum_{h=1}^{k} x_{\pi, S^T_{h}, h}.
\]

(68)

The following is a corollary of this lemma.

**Corollary 4:** Following Lemma 6, we can further assume that there exists a finite set of (variable, function) pairs \( \{ (\chi_{\pi, l}, \theta_{\pi, l} (\cdot)) : \forall \pi, \forall l \} \), such that each variable \( \chi_{\pi, l} \) is in the range [0, 1], each function \( \theta_{\pi, l} (\cdot) \) is a non-increasing map from \([M]\) to \([M+1]\), and jointly they satisfy (i) \( \theta_{\pi, l} (M) \leq M \), and (ii)

\[
\forall \pi, \forall j, h \in [M], x_{\pi, S^T_{h-1}, h} = \sum_{\forall l \cdot \chi_{\pi, l} (\cdot) = h} \chi_{\pi, l}.
\]

(69)

We term \( \chi_{\pi, l} \) and \( \theta_{\pi, l} (\cdot) \) as the (normalised) size and the label function of the \( l \)-th chunk.

The proofs of Lemma 6 and Corollary 4 are relegated to Appendix G.

**Remark 1:** We first note that since the label function satisfies \( \theta_{\pi, l} (M) \leq M \), we must have

\[
\{ \chi_{\pi, l} : \forall \pi, \forall l \}
\]

\[
= \{ \chi_{\pi, l} : \forall \pi, \forall h \in [M], \forall l, \text{ s.t. } \theta_{\pi, l} (M) = h \}.
\]

By choosing \( j = M \) in (69), we have

\[
\forall \pi, \forall h \in [M], x_{\pi, h, h} = \sum_{\forall l \cdot \theta_{\pi, l} (M) = h} \chi_{\pi, l}.
\]

As a result, the summation of all \( \chi_{\pi, l} \) equals to \( \sum_{h=1}^{M} x_{\pi, h, h} \). Therefore, the term \( \chi_{\pi, l} \) can indeed be viewed as the normalized size of the \( l \)-th chunk.

**Remark 2:** One special property of a chunk is the associated label function \( \theta_{\pi, l} (\cdot) \) that is non-increasing. Such a property is critical to our code construction.

### B. The Central Step (Step 4) of The Achievability Scheme

In this subsection, we concentrate on how to perform network coding over the partition \( \{ \chi_{\pi, l} : \forall \pi, \forall l \} \). For the \( l \)-th chunk, we will design the corresponding \( (n \chi_{\pi, l}) \times (n \sum_{h=1}^{M} R_{h}) \) coding matrix \( \mathbf{A}_{\pi, l} \). After finising the creation for each chunk, the overall coding matrix is an \( n \times (n \sum_{i=1}^{M} R_{i}) \) matrix that vertically concatenates all the coding matrices of the partition \( \{ \chi_{\pi, l} : \forall \pi, \forall l \} \).

More explicitly, we define

\[
\tilde{y}_{i, T, (\pi, S^T_j)} \triangleq \sum_{\forall \pi \in [M], T \subseteq [2M]: \exists \pi, T^\pi \leq S^T_j} y_{i, T, (\pi, S^T_j)}
\]

for any given permutation \( \pi, i \in [M] \), and \( S^T_j \) satisfying \( i \notin S^T_j \). Namely, \( n \cdot \tilde{y}_{i, T, (\pi, S^T_j)} \) is the number of basis vectors \( \mathbf{v} \) that have been reassigned to a special group indexed by \( (\pi, S^T_j) \),
through \((67)\). If we define \(T^\pi_j \overset{\Delta}{=} |M| \langle S^\pi_j \cup \{i\} \rangle\), and use \(v_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle, k = 1, \ldots, n \cdot \tilde{y}_{i,0,(\pi, S^\pi_j \rangle}) \) to denote those vectors, then by our discussion in the previous subsection,

\[
\left\{ v_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle} : \forall \pi, \forall j, \forall k \right\}
\]

form the basis vectors of the entire message space \(\Omega_t\). Using the new basis vectors, for each time slot \(t\), the relay \(r\) transmits a coded packet \(W_r(t)\) by

\[
W_r(t) = \sum_{i=1}^{M} \sum_{t \in [n]} n_{B_{i,\pi,S^\pi_j}} \sum_{k=1}^{c_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle}} v_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle}(t) v_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle} X_T^T,
\]

Our task is to design the coefficients \(c_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle}(t)\) for all \(t \in [n]\). Similar to the discussion in Sections V-A and V-B, we can express the coefficients \(c_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle}(t)\) by the equivalent \(n \cdot \left( n \sum_{i=1}^{M} R_i \right)\) coding matrix. Our systematic construction then focuses on the individual \((n \chi_{\pi,l}) \cdot \left( n \sum_{i=1}^{M} R_i \right)\) coding sub-matrices \(A_{\pi,l}\). For future reference, we also use \(\nu_{\pi,l}^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle}\) to denote the space spanned from \(\nu_k^{\pi, \langle i \rightarrow T^\pi_j S^\pi_j \rangle}\).

Our construction will rely on the code alignment technique first introduced in Section V-B. For each chunk \((\pi, l)\), we create an \((n \chi_{\pi,l}) \times \left( n \sum_{i=1}^{M} R_i \right)\) coding sub-matrix \(A_{\pi,l}\) by choosing each entry independently and uniformly randomly from GF\((q)\). The \(B_{\pi,l}\) matrix will later be used to create the \((n \chi_{\pi,l}) \times \left( n \sum_{i=1}^{M} R_i \right)\) coding matrix \(A_{\pi,l}\). For each permutation \(\pi\), the forking subroutine will construct all \(\{A_{\pi,l} : \forall \pi\}\) simultaneously.

1: **Initialization**: All entries of \(A_{\pi,l}\) are set to zero for all \(l\).
2: for \(j_0 = 1, \ldots, M\) do
3: for each \(i_0\) that is not in \(S^\pi_{j_0+1}\) do
4: set \(cnt_{\text{assigned}} \leftarrow 0\).
5: set the status of all chunks \((\pi, l, \theta_{\pi,l}(\cdot))\) to “not-processed.”
6: while \(cnt_{\text{assigned}} < n \cdot \tilde{y}_{i_0,(\pi, S^\pi_{j_0+1})}\) do
7: Arbitrarily choose one “not-processed” chunk and denote it by \((\chi_{\pi,l_0}, \theta_{\pi,l_0}(\cdot))\) for some \(l_0\).
8: set \(\Delta_{\text{io,cum}} \leftarrow n \cdot \chi_{\pi,l_0} \cdot \min(p_{r;i}, p_{r;\theta_{\pi,l}(j_0)})\).
9: if \(j_0 > 1\) then
10: set \(\Delta_{\text{io,prev}} \leftarrow n \cdot \chi_{\pi,l_0} \cdot \min(p_{r;i}, p_{r;\theta_{\pi,l}(j_0-1)})\).
11: else
12: \(\Delta_{\text{io,prev}} \leftarrow 0\).
13: end if
14: set \(\Delta_{\text{io,extr}} \leftarrow (\Delta_{\text{io,cum}} - \Delta_{\text{io,prev}})\).
15: set \(cnt_{\text{new}} \leftarrow \min(cnt_{\text{assigned}} + \Delta_{\text{io,extr}}, n \tilde{y}_{i_0,(\pi, S^\pi_{j_0+1})})\).
16: set \(\Delta \leftarrow cnt_{\text{new}} - cnt_{\text{assigned}}\).
17: Consider the \((\Delta_{i_0,\text{prev}} + 1)\)-th to the \((\Delta_{i_0,\text{prev}} + \Delta)\)-th columns of matrix \(B_{\pi,l_0}\). Copy that portion of \(B_{\pi,l_0}\) to the columns of \(A_{\pi,l_0}\) corresponding to

\[
v_k^{\pi, \langle i_0 \rightarrow T^\pi_j S^\pi_j \rangle}(t)\] for all \(k \in (\text{cnt}_{\text{assigned}}, \text{cnt}_{\text{new}})\).
18: set the status of the chunk \((\chi_{i_0, l_0}, \theta_{\pi,l_0}(\cdot))\) to “processed.”
19: set \(cnt_{\text{assigned}} \leftarrow \text{cnt}_{\text{new}}\).
20: end while
21: end for
22: end for

Intuitively, during the FOR loop of the given \(j_0\), the above sub-routine “assigns” the \(n \cdot \tilde{y}_{i_0,(\pi, S^\pi_{j_0+1})}\) columns corresponding to vectors \(v_k^{\pi, \langle i_0 \rightarrow T^\pi_j S^\pi_j \rangle}\) to different chunks \((\chi_{\pi,l, \theta_{\pi,l}(\cdot)})\) for all \(i_0 \notin S^\pi_{j_0+1}\). Although the assignment, Lines 3 to 21, is made separately for distinct \(i_0 \notin S^\pi_{j_0+1}\), a common \(B_{\pi,l}\) is used in Line 17, which results in “aligned codeword spaces” that is necessary for approaching the optimal throughput. Lines 4 and 5 initialize the FOR loop. Line 6 stops the inner loop whenever all columns have been assigned, i.e., when \(cnt_{\text{assigned}} = n \cdot \tilde{y}_{i_0,(\pi, S^\pi_{j_0+1})}\).

During the inner loop, Lines 6 to 20, each unprocessed chunk \((\chi_{i_0, l_0}, \theta_{\pi,l_0}(\cdot))\) has already been assigned \(\Delta_{\text{io,prev}}\) columns during the previous \(j\)-th steps for those \(j < j_0\). On the other hand, for the \(j_0\)-th step, each chunk can still carry \(\Delta_{\text{io,extr}}\) extra columns, which we term the “extra rank capacity” and is computed by Lines 8, 10, and 14 of the subroutine. It is worth noting that the computation of the extra rank capacity in Lines 8, 10, and 14 depends on \(i_0\), the user of interest. The reason is that each user \(i_0\) is facing different success probability \(p_{r;\theta_{\pi,l}}\), which affects how many columns a chunk can carry for each user \(i_0\).

Now we need to prove that after processing all chunks, we can successfully assign all \(n \cdot \tilde{y}_{i_0,(\pi, S^\pi_{j_0+1})}\) columns. To that end, we notice that when \(j_0 = 1\), by Corollary 4 with \(j = 1\), the first part of Lemma 6, and by (63) with \(j = 1\), we have

\[
\sum_{\forall \ell} (\chi_{i_0, l_0} \cdot (\min(p_{r;i}, p_{r;\theta_{\pi,l}(i_0)}(1)) - 0)) = \sum_{h=1}^{M} \left( \sum_{\forall \ell: \ell(i)=h} \chi_{i_0, l_0} \cdot (\min(p_{r;i}, p_{r;h})) \right)
\]

\[
= \sum_{h=1}^{M} x_{\pi, S^\pi_{j_0+1}, h} \cdot (\min(p_{r;i}, p_{r;h})) 
\]

\[
\geq 1 \cdot i_0 \cdot \tilde{y}_{i_0, (\pi, S^\pi_{j_0+1})} \quad (71)
\]

for all \(i_0 \notin S^\pi_{j_0+1}\). Similarly, for \(j_0 \geq 2\), by Corollary 4 and by
for the four chunks, respectively. The subroutine for $j_0 = 1$ then copies the first $n\chi_{\pi,2} \cdot p_{r,2}$ columns and the first $n\chi_{\pi,3} \cdot p_{r,3}$ columns of $B_{\pi,2}$ and $B_{\pi,3}$ to their corresponding columns of $V_{2-T3}^{\pi}$, respectively. See Fig. 6 for illustration. For simplicity of the figure, we actually treat the first $n\chi_{\pi,2} \cdot p_{r,3}$ columns of $B_{\pi,2}$ and the next $n\chi_{\pi,2} \cdot (p_{r,2} - p_{r,3})$ columns of $B_{\pi,2}$ as two separate groups of columns. Therefore the subroutine copies two groups of columns of $B_{\pi,2}$ to their corresponding positions in $A_{\pi,2}$. In Fig. 6, we use the dashed lines to represent the “copy” operations.

For $j_0 = 2$, since $S_{2+1}^\pi = \{1\}$ we consider $i_0 = 2$ and 3, and the corresponding columns of $V_{2-T3}^{\pi}$ and $V_{3-T2}^{\pi}$ separately. First consider $i_0 = 2$ and the corresponding $V_{2-T3}^{\pi}$. With the specified label functions, the extra rank capacities that can be used for carrying $V_{2-T3}^{\pi}$ columns are computed by Line 14:

for the four chunks, respectively. The subroutine then copies the first $n\chi_{\pi,1} \cdot p_{r,3}$ columns of $B_{\pi,1}$ to their corresponding columns of $V_{2-T3}^{\pi}$. If we treat the first $n\chi_{\pi,3} \cdot p_{r,3}$ columns of $B_{\pi,3}$ as the first group, and the next $n\chi_{\pi,3} \cdot (p_{r,2} - p_{r,3})$ columns of $B_{\pi,3}$ as the second group, then the subroutine copies the second group of columns of $B_{\pi,3}$ to their corresponding columns of $V_{2-T3}^{\pi}$. (See Fig. 6.)

The subroutine then turns to the case of $i_0 = 3$ and the corresponding $V_{3-T2}^{\pi}$. The extra rank capacities that can be used for carrying $V_{3-T2}^{\pi}$ columns are computed as follows.

for the four chunks, respectively. The subroutine then copies the first $n\chi_{\pi,1} \cdot p_{r,3}$ columns of $B_{\pi,1}$ to their corresponding columns of $V_{2-T3}^{\pi}$. If we treat the first $n\chi_{\pi,3} \cdot p_{r,3}$ columns of $B_{\pi,3}$ as the first group, and the next $n\chi_{\pi,3} \cdot (p_{r,2} - p_{r,3})$ columns of $B_{\pi,3}$ as the second group, then the subroutine copies the second group of columns of $B_{\pi,3}$ to their corresponding columns of $V_{2-T3}^{\pi}$. (See Fig. 6.)

When $j_0 = 3$, since $S_{3+1}^\pi = \emptyset$ we consider $i_0 = 1$ and 2, and the corresponding columns $V_{2-T3}^{\pi}, V_{2-13}^{\pi}$, and $V_{3-12}^{\pi}$ separately. First consider $i_0 = 1$ and the corresponding $V_{2-T3}^{\pi}$. With the specified label functions, the extra rank capacities that can be used for carrying $V_{2-T3}^{\pi}$ columns are computed as
$M = 3, \ \pi = (2, 3, 1)$

<table>
<thead>
<tr>
<th>$\theta_x(1, 2, 3) = (4, 3, 1)$</th>
<th>$\theta_x(1, 2, 3) = (2, 2, 1)$</th>
<th>$\theta_x(1, 2, 3) = (3, 2, 1)$</th>
<th>$\theta_x(1, 2, 3) = (4, 4, 2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{\chi, 1} \cdot p_{r, 1}$</td>
<td>$n_{\chi, 2} \cdot p_{r, 2}$</td>
<td>$n_{\chi, 3} \cdot p_{r, 3}$</td>
<td>$n_{\chi, 4} \cdot p_{r, 4}$</td>
</tr>
<tr>
<td>$n_{\chi, 1} \cdot p_{r, 1}$</td>
<td>$n_{\chi, 2} \cdot p_{r, 2}$</td>
<td>$n_{\chi, 3} \cdot p_{r, 3}$</td>
<td>$n_{\chi, 4} \cdot p_{r, 4}$</td>
</tr>
<tr>
<td>$n_{\chi, 1} \cdot (p_{r, 3} - p_{r, 2})$</td>
<td>$n_{\chi, 2} \cdot (p_{r, 3} - p_{r, 2})$</td>
<td>$n_{\chi, 3} \cdot (p_{r, 3} - p_{r, 2})$</td>
<td>$n_{\chi, 4} \cdot (p_{r, 3} - p_{r, 2})$</td>
</tr>
<tr>
<td>$n_{\chi, 1} \cdot (p_{r, 3} - p_{r, 2})$</td>
<td>$n_{\chi, 2} \cdot (p_{r, 3} - p_{r, 2})$</td>
<td>$n_{\chi, 3} \cdot (p_{r, 3} - p_{r, 2})$</td>
<td>$n_{\chi, 4} \cdot (p_{r, 3} - p_{r, 2})$</td>
</tr>
</tbody>
</table>

$V_{1-12}^\pi, V_{13-12}^\pi, V_{13-23}^\pi, V_{23-12}^\pi, V_{23-13}^\pi, V_{13-23}^\pi, V_{13-12}^\pi$ 

$A_{x, 1}$ 

$A_{x, 2}$ 

$A_{x, 3}$ 

$A_{x, 4}$ 

Fig. 6. Illustration of the systematic code construction in Section VI-B.

follows.

\[
\begin{align*}
 n_{\chi, 1} \cdot \min(p_{r, 1} - p_{r, 3}) &= n_{\chi, 1} \cdot (p_{r, 1} - p_{r, 3}), \\
 n_{\chi, 2} \cdot \min(p_{r, 1} - p_{r, 2}) &= n_{\chi, 2} \cdot (p_{r, 1} - p_{r, 2}), \\
 n_{\chi, 3} \cdot \min(p_{r, 1} - p_{r, 2}) &= n_{\chi, 3} \cdot (p_{r, 1} - p_{r, 2}), \\
 n_{\chi, 4} \cdot \min(p_{r, 1} - p_{r, 2}) &= n_{\chi, 4} \cdot (p_{r, 1} - p_{r, 2}), \\
\end{align*}
\]

for the four chunks, respectively. Using the same concept of “groups” as discussed in the cases of $j_0 = 1$ and 2, the subroutine copies the second and the third groups of columns of $B_{\pi, 1}$, the third group of columns of $B_{\pi, 2}$, the third group of columns of $B_{\pi, 3}$, and the first and the second groups of columns of $B_{\pi, 4}$ to their corresponding columns of $V_{12}^\pi$, respectively. (See Fig. 6.)

For $i_0 = 2$, the extra rank capacities that can be used for carrying $V_{12}^\pi$ are

\[
\begin{align*}
 n_{\chi, 1} \cdot \min(p_{r, 2} - p_{r, 1}) &= n_{\chi, 1} \cdot (p_{r, 2} - p_{r, 1}), \\
 n_{\chi, 2} \cdot \min(p_{r, 2} - p_{r, 1}) &= n_{\chi, 2} \cdot (p_{r, 2} - p_{r, 1}), \\
 n_{\chi, 3} \cdot \min(p_{r, 2} - p_{r, 1}) &= n_{\chi, 3} \cdot (p_{r, 2} - p_{r, 1}), \\
 n_{\chi, 4} \cdot \min(p_{r, 2} - p_{r, 1}) &= n_{\chi, 4} \cdot (p_{r, 2} - p_{r, 1}), \\
\end{align*}
\]

for the four chunks, respectively. The subroutine then copies the second group of columns of $B_{\pi, 1}$, and the first and the second groups of columns of $B_{\pi, 4}$ to their corresponding columns of $V_{13}^\pi$, respectively. (See Fig. 6.)

For $i_0 = 3$, the extra rank capacities that can be used for carrying $V_{13}^\pi$ are

\[
\begin{align*}
 n_{\chi, 1} \cdot \min(p_{r, 3} - p_{r, 1}) &= n_{\chi, 1} \cdot (p_{r, 3} - p_{r, 1}), \\
 n_{\chi, 2} \cdot \min(p_{r, 3} - p_{r, 1}) &= n_{\chi, 2} \cdot (p_{r, 3} - p_{r, 1}), \\
 n_{\chi, 3} \cdot \min(p_{r, 3} - p_{r, 1}) &= n_{\chi, 3} \cdot (p_{r, 3} - p_{r, 1}), \\
 n_{\chi, 4} \cdot \min(p_{r, 3} - p_{r, 1}) &= n_{\chi, 4} \cdot (p_{r, 3} - p_{r, 1}), \\
\end{align*}
\]

for the four chunks, respectively. The subroutine then copies the first group of columns of $B_{\pi, 4}$ to their corresponding columns of $V_{12}^\pi$. (See Fig. 6.)

To prove the decodability of the above network code construction, we first notice that in the above construction, each column participates in one and only one chunk $(\chi_{\pi, l}, \theta_{\pi, l})$ for some $\pi$ and $l$ (as can also be seen in Fig. 6). The following lemma then guarantees the decodability of the above network code construction.

**Lemma 7:** For any chunk index $(\pi, l)$, any destination $d_i$ can successfully decode all the user-$i$ columns that participate in chunk $(\chi_{\pi, l}, \theta_{\pi, l})$ by the observations of all time slots $t$ that correspond to the coding submatrix $A_{x, l}$.

We conclude this section by the proof of Lemma 7.

**Proof of Lemma 7:** Consider any chunk with index $(\pi, i_0)$. For any user $i$, we use $V_{i-j_0+1}^\pi(i-j_0+1)\overrightarrow{j_0+1}$ to denote the columns of $V_{i-j_0+1}^\pi(i-j_0+1)\overrightarrow{j_0+1}$ that participate in chunk $(\chi_{\pi, i_0}, \theta_{\pi, i_0})$ during the $j_0$-th iteration of the outer FOR loop, Lines 2 to 22. Proving Lemma 7 is thus equivalent to proving that any user $i_0$ can decode the subspace $V_{i_0-j_0+1}^\pi(i_0-j_0+1)\overrightarrow{j_0+1}$ for all $j_0 \in [M]$ satisfying $i_0 \notin S_j^\pi$.

For any $i_0$, let $j^*$ denote the unique index in $[M]$ satisfying $i_0 \notin S_j^\pi$ but $i_0 \in S_{j^*}^\pi$. By the definition $S_j^\pi$, we thus have $i_0 = \pi(j^*)$. We now compute an upper bound on the sum of the extra rank capacities of chunk $(\pi, i_0)$ during the $j_0$-th

\footnote{We use “decoding a user-$i$ column” as shorthand for “decoding a vector $v$ that corresponds to one of the user-$i$ columns.”}
iteration for \( j_0 = j^* \) to \( M \).

\[
\sum_{j_0 = j^*}^{M} n_{\chi, i_0} \cdot \left( \min(p_{r; i_0}, p_{r; \theta_{i_0}(j_0)}) - \min(p_{r; i_0}, p_{r; \theta_{i_0}(j_0 - 1)}) \right) \\
= n_{\chi, i_0} \left( \min(p_{r; i_0}, p_{r; \theta_{i_0}(j)}) - \min(p_{r; i_0}, p_{r; \theta_{i_0}(j - 1)}) \right) \\
\leq n_{\chi, i_0} \cdot \max \left( 0, p_{r; i_0} - p_{r; \theta_{i_0}(j^* - 1)} \right)
\]

(74)

Note that from user \( i_0 \)'s perspective, the columns of the desired information \( \tilde{V}_{i_0}^{(\pi, l_0)} \) are assigned to \( A_{\pi, l} \) through the concept of extra rank capacity during \( j_0 = j \). Therefore the total number of information columns \( \tilde{V}_{i_0}^{(\pi, l_0)} \) for \( j = j^* \) to \( M \) must be no larger than (74).

We note that since other user \( i, i \neq i_0 \), also participates in \( A_{\pi, l_0} \), decoding for user-\( i_0 \) has to take into account other users \( i \neq i_0 \). We first consider any \( i = \pi(h) \) for some \( h > j^* \). The corresponding subspaces of interest \( \tilde{V}_{i_0}^{(\pi, l_0)} \) must satisfy \( j \geq h \) otherwise we will have \( i \in S_{j+1}^\pi \). Since \( j \geq h > j^* \), we must have \( i_0 \in [M] \setminus S_{j+1}^\pi \). Therefore, from user-\( i_0 \)'s perspective, all the subspaces \( \tilde{V}_{i_0}^{(\pi, l_0)} \) are side information already known to user \( i_0 \). Therefore, the columns corresponding to \( \tilde{V}_{i_0}^{(\pi, l_0)} \) are transparent from user-\( i_0 \)'s perspective.

We then consider any \( i = \pi(h) \) for some \( h < j^* \). The corresponding subspaces of interest \( \tilde{V}_{i_0}^{(\pi, l_0)} \) must satisfy \( j \geq h \) otherwise we will have \( i \in S_{j+1}^\pi \). By the same argument as in the previous case, those \( \tilde{V}_{i_0}^{(\pi, l_0)} \) for some \( j \geq j^* \) are side information already known to user \( i_0 \) and can be omitted in our discussion. We thus focus only on the subspaces \( \tilde{V}_{i_0}^{(\pi, l_0)} \) for all \( j \in [h, j^* - 1] \). Recall that all the columns of the subspaces \( \tilde{V}_{i_0}^{(\pi, l_0)} \) are assigned to \( A_{\pi, l} \) during \( j_0 = j \). Therefore, the columns of all those non-transparent subspaces \( \tilde{V}_{i_0}^{(\pi, l_0)} \) are copied from the \( k \)-th column of \( B_{\pi, l_0} \), where \( k \) is in the range

\[
k \in \left[ n_{\chi, i_0} \cdot \min(p_{r; i_0}, p_{r; \theta_{i_0}(j - 1)}) + 1, \right.
\]

\[
\left. n_{\chi, i_0} \cdot \min(p_{r; i_0}, p_{r; \theta_{i_0}(j)}) \right]
\]

(75)

where the “\( \leq \)” relationship is due to the fact that the range of \( j \) for non-transparent subspaces is \( [h, j^* - 1] \) and \( \theta_{i_0}(j) \) is a non-increasing function. Due to our code-alignment-based construction, all columns of the non-transparent subspaces \( \tilde{V}_{i_0}^{(\pi, l_0)} \) are copied from (a subset of) the first \( n_{\chi, i_0} \cdot p_{r; \theta_{i_0}(j^* - 1)} \) columns of the same matrix \( B_{\pi, l_0} \). Therefore, from the decoding perspective of user \( i_0 \), the effective degrees of freedom of the non-transparent columns is no larger than

\[
n_{\chi, i_0} \cdot p_{r; \theta_{i_0}(j^* - 1)}.
\]

For simplicity, we use the convention that \( \min(p_{r; i_0}, p_{r; \theta_{i_0}(0)}) = 0 \) in (74) and in the following discussion.

Also note that the total number of successful reception of user \( i_0 \) for the chunk \( (\chi_{i_0}, \theta_{i_0}(\cdot)) \) is

\[
n_{\chi, i_0} \cdot p_{r; i_0}.
\]

(76)

For the non-trivial case in which the total number of the to-be-decoded columns of \( \tilde{V}_{i_0}^{(\pi, l_0)} \) is non-zero, we can remove the \( \max(0, \cdot) \) operation of (74). Then by comparing (74), (75), and (76), we must have that the “total number of to-be-decoded columns” plus “the effective degrees of freedom of the non-transparent interference columns” is no larger than “the total number of observed time slots.” By the MDS property of RLNC, it is thus guaranteed that user \( i_0 \) can successfully decode all the desired information columns \( \tilde{V}_{i_0}^{(\pi, l_0)} \) from the observation of the \( A_{\pi, l_0} \) coding sub-matrix. The proof is thus complete.

VII. NUMERICAL EXPERIMENTS AND SOME OBSERVATIONS

A. Numerical Experiments

1) Capacity Region for \( M \leq 3 \): For the simplest case of \( M = 2 \), the capacity region is described by Corollary 2, which can be expressed by six linear inequalities once we expand the \( \min \) and the projection operators.

For \( M = 3 \), we plot the 3-D capacity region for an example network. To simplify the illustration, we assume all the PECs are spatially independent. Namely, the joint success probability \( p_{s \rightarrow sT} \) can be expressed as the product of the marginals. We consider an example network with the marginals being

\[
\begin{bmatrix}
p_{s_1, r} & p_{s_1, 2} & p_{s_1, 3} \\
p_{s_2, 1} & p_{s_2, r} & p_{s_2, 3} \\
p_{s_3, 1} & p_{s_3, 2} & p_{s_3, r}
\end{bmatrix} = \begin{bmatrix}
1 & 0.12 & 0.1 \\
0.1 & 1 & 0.15 \\
0.05 & 0.1 & 1
\end{bmatrix}
\]

and \((p_{r, 1}; p_{r, 2}; p_{r, 3}) = (0.9, 0.8, 0.7)\).

Fig. 7 illustrates the corresponding 3-D capacity region of the above wireless 1-hop network. As seen in Fig. 7, the capacity region of \( M = 3 \) is non-trivial and contains many facets that are governed by the linear inequalities in Fig. 3.
deficiency is defined as the capacity inner bound in Proposition 4. The deficiency quantities the suboptimality of the inner bound of Proposition 4 when compared to the true capacity region. In our numerical evaluation (see Fig. 8), the difference defi is no larger than 1% for 98.9% of the time.

To explore the causes of the suboptimality of the inner bound, we first notice that the achievability scheme in Section VI consists of two components. The first component is to decide “which basis vector has been overheard by which destination” so that we can partition the message space \( \Omega_i \) into proper side information subspaces, which is achieved in Step 2 of the scheme. The second component is to redistribute the side information subspaces (as in Step 3) and then use the code alignment technique in Step 4 to “pack the side information subspaces” as compactly as possible.

First consider the second component. For each permutation \( \pi \), the corresponding side information subspaces \( V_{i \rightarrow T_{j+1}^{\pi}(j+1)} \) have an acyclic relationship in the sense that the corresponding non-overhearing sets \( S_{j+1}^{\pi} \), \( j \in [M] \), have a total ordering \( S_2^{\pi} \supseteq S_3^{\pi} \supseteq \cdots \supseteq S_{M+1}^{\pi} \). It can be proven that given the to-be-packed subspaces \( V_{i \rightarrow T_{j+1}^{\pi}(j+1)} \), \( \forall j \in [M], \forall i \in [M]\backslash S_{j+1}^{\pi} \), the proposed redistribution plus code-alignment coding scheme is the optimal. Namely, if we are only interested in packing subspaces \( V_{i \rightarrow T_{j+1}^{\pi}(j+1)} \) that have an acyclic relationship, the proposed redistribution plus code-alignment scheme can pack the subspaces using the minimal number of time slots. However, the suboptimality lies in the approach of performing coding on the partitioned subspaces separately. For comparison, the optimal coding solution in Section V-B performs coding with holistic consideration of all subspaces. That is, the coding matrix in Fig. 5 does not have the clean separation as in the coding matrix of Fig. 6. On the other hand, a holistic network coding scheme may be difficult to design for \( M > 3 \) due to the exponential number of different types of overhearing patterns for large \( M \).

Further, even the first component that partitions the message space \( \Omega_i \) into proper side information subspaces turns out to be highly non-trivial for the cases of \( M \geq 3 \). For example, consider an \((n, R)\)-dimensional message vector \( \mathbf{X} \), and we use RLNC to mix all \( nR \) symbols by randomly generated \( n \) coding vectors \( c(1) \) to \( c(n) \). Consider two receivers \( d_1 \) and \( d_2 \) and the corresponding success probabilities being \( p_1 \) and \( p_2 \), respectively. By the MDS property of RLNC, the linear subspace received by \( d_1 \), denoted by \( V_1 \), has rank \( \text{Rank}(V_1) = \min(nR, np_1) \). The linear subspace received by \( d_2 \), denoted by \( V_2 \), has rank \( \text{Rank}(V_2) = \min(nR, np_2) \). The question is how large is the rank of the intersection, \( \text{Rank}(V_1 \cap V_2) \). The answer to this question is critical for partitioning the message space since it corresponds to how many basis vectors that are known to both \( d_1 \) and \( d_2 \). With some algebraic computation, it can be shown that

\[
\text{Rank}(V_1 \cap V_2) = \min(nR, np_1) + \min(nR, np_2) - \min(nR, np_{12})
\]

where \( p_{12} \) is the probability that a packet is received by at least one of \( d_1 \) and \( d_2 \). One can verify that for very
general choices of $p_1$, $p_2$, and $p_{1 \cup 2}$, say $(p_1, p_2, p_{1 \cup 2}) = (0.5, 0.5, 0.75)$, (77) is neither concave nor convex with respect to $R$. This thus presents a substantial challenge when partitioning the message space into its proper side information subspaces. In the broadest sense, the optimal construction in Section V-B directly uses the optimal expression (77) but our systematic inner bound construction in Section VI uses a conservative but more tractable lower bound

$$\operatorname{Rank}(\mathcal{V}_1 \cap \mathcal{V}_2) \geq \min (n R, n \cdot (p_1 + p_2 - p_{1 \cup 2})),$$

which is concave with respect to $R$. However, using the above lower bound induces capacity loss and part of it is reflected in the def of Fig. 8.

2) Complexity Of The Outer And Inner Bounds: Although checking the feasibility of LP problems is a P problem, the number of variables and linear constraints of Propositions 2 and 4 is exponential (or super exponential) with respect to $M$. On the other hand, for any fixed $M$ value, the complexity of computing the bounds does not depend on $n$ the block length of the network code, even though there are totally $\sum_{i=1}^{M} n R_i$ symbols, $X_{i_1}, \cdots, X_{i_n R_i}, i \in [M]$, to be coded together. This fractional setting thus presents a big complexity improvement over the integral settings of [27] and of [3], for which the INC complexity is exponential with respect to the number of symbols to be coded together.

Note that like any NP-hardness results, the NP-hardness of finding the index code capacity [1], [3] is a worst-case analysis. As can be seen in the cumulative distribution functions in Fig. 8, the gap between the outer and the inner bounds can be as large as 27% in the worst case, which fits the prediction of the NP-hardness results. On the other hand, the outer and inner bounds meet in the average cases and can thus be used effectively as the substitute of capacity from the practical perspective.

We would also like to emphasize that in a wireless network setting, it is unlikely\(^{10}\) that more than $M \geq 6$ sessions $(s_i, d_j), i \in [M]$, are using the same relay node $r$. For small $M \leq 5$, the outer and inner bounds can be easily evaluated by standard LP solvers.

VIII. CONCLUSION

In this work, we have studied the capacity region of wireless 1-hop relay networks with $M$ coexisting unicast sessions, based on the packet erasure channel model and a 2-staged network coding setting that captures the benefits of message side information. The 2-staged interseession network coding (2-staged INC) capacity region for $M \leq 3$ has been fully characterized. For $M \geq 4$, the 2-staged INC capacity has been bounded by a new pair of outer and inner bounds, the tightness of which is within 1% for at least 96.7% of the times for $M \in \{4, 5\}$. The outer and inner bounds can thus be used effectively as the substitute of 2-staged INC capacity from the practical perspective.

The proposed achievability scheme is based on the concept of code alignment, which was also used in [13], [38], [40], [43] and is in parallel with the interference alignment technique in the setting of Gaussian interference channel. The results in this paper can also be viewed as the generalization of index-coding capacity from wireline broadcast with binary alphabets to wireless broadcast with high-order alphabets.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their valuable suggestions and help on refining the proofs.

APPENDIX A

PROOF OF PROPOSITION 1

Proof of Proposition 1: Inequality (5) can be proven by the statistically degraded channel arguments. To prove inequality (6), we assume the broadcast PEC is physically degraded. Namely, the channel status $(1\{Z_{1,t} \neq *\}, 1\{Z_{2,t} \neq *\}, 1\{Z_{3,t} \neq *\})$ has the following distribution:

$$\operatorname{Prob} (1\{Z_{1,t} \neq *\} = a, 1\{Z_{2,t} \neq *\} = b, 1\{Z_{3,t} \neq *\} = c) = \begin{cases} 1 - p_1 & \text{if } a = b = c = 0 \\ p_1 - p_2 & \text{if } a = 1, b = c = 0 \\ p_2 - p_3 & \text{if } a = b = 1, c = 0 \\ p_3 & \text{if } a = b = c = 1 \\ 0 & \text{otherwise} \end{cases}$$

With the above construction, $U \rightarrow W \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3$ form a Markov chain.

We will first prove the following inequality:

$$I(U; Z_1 | Z_2 Z_3) \leq \frac{p_1 - p_2}{p_1 - p_3} I(U; Z_1 | Z_3).$$

(78)

To that end, we notice the following

$$I(U; Z_1 | Z_2 Z_3) = \sum_{t=1}^{n} I(U; Z_1(t) | Z_1(t)^{-1} Z_2(t) Z_3),$$

(79)

where $[Z_1(t)]^{-1} \Delta (Z_1(t) : \forall \tau \in [t-1])$. Inequality (79) follows from the facts $H(Z_1(t) | [Z_1(t)]^{-1} Z_2 Z_3) \leq H(Z_1(t) | [Z_1(t)]^{-1} Z_2 Z_3)$ and $H(Z_1(t) | [Z_1(t)]^{-1} Z_2 Z_3) = H(Z_1(t) | [Z_1(t)]^{-1} Z_2 Z_3)$. Each term of the summation in (79) can be further simplified by

$$I(U; Z_1(t) | [Z_1(t)]^{-1} Z_2(t) Z_3) = I(U; Z_1(t) \{Z_1(t) \neq *\} | [Z_1(t)]^{-1} Z_2(t) Z_3) = \operatorname{Prob}(Z_1(t) \neq *, Z_2(t) = *) \cdot I(U; Z_1(t) | [Z_1(t)]^{-1}, Z_1(t) \neq *, Z_3(t) = Z_3(t)) = (p_1 - p_2) \cdot I(U; Z_1(t) | [Z_1(t)]^{-1}, Z_1(t) \neq *, Z_3(t) = *) (80)$$

(81)
where \(|Z_3|_t \triangleq \{Z_3(\tau) : \tau \in [n]\}t\) and (80) follows from the fact that partition \(Z_3(\tau) = [Z_3(\tau)]_t\) and (81) follows from the fact that conditioning on \(Z_3(\tau) \neq [Z_3(\tau)]_t\), whether \(Z_3(\tau) = [Z_3(\tau)]_t\) or not does not change the entropy of \(Z_3(\tau)\). That is,
\[
H(Z_1(t))|Z|_t^{-1}, Z_1(t) \neq *, Z_2(t) = Z_3(t) = *, [Z_3]_t) = H(Z_1(t))|Z|_t^{-1}, Z_1(t) \neq *, Z_3(t) = *, [Z_3]_t) \text{ and } H(Z_1(t))|Z|_t^{-1}, Z_1(t) \neq *, Z_2(t) = Z_3(t) = *, [Z_3]_t) = H(Z_1(t))|Z|_t^{-1}, Z_1(t) \neq *, Z_3(t) = *, [Z_3]_t).
\]
We also notice that
\[
I(U; Z_1(t))|Z|_t^{-1}, Z_3) = I(U; Z_1(t))|Z|_t^{-1}1_{(Z_1(\tau)\neq\tau)} Z_3) = \text{Prob}(Z_1(t) \neq *, Z_3(t) = *)
\]
\[
\cdot I(U; Z_1(t))|Z|_t^{-1}, Z_1(t) \neq *, Z_3(t) = *, [Z_3]_t) = (p_1 - p_3)I(U; Z_1(t))|Z|_t^{-1}, Z_1(t) \neq *, Z_3(t) = *, [Z_3]_t).
\]
As a result,
\[
(81) = \frac{p_1 - p_2}{p_1 - p_3} I(U; Z_1(t))|Z|_t^{-1}, Z_3). \tag{82}
\]
Substituting (82) into (79), we have proven (78).
Since \(I(U; Z_1(t))|Z|_t Z_3 = I(U; Z_1(t))|Z_3)\), Eq. (78) implies
\[
I(U; Z_2|Z_3) \geq \frac{p_2 - p_3}{p_1 - p_3} I(U; Z_1|Z_3). \tag{83}
\]
The final step is to notice that \(U \rightarrow W \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3\) form a Markov chain. We thus have
\[
I(U; Z_2|Z_3) = I(U; Z_2) - I(U; Z_3)
\]
and
\[
I(U; Z_1|Z_3) = I(U; Z_1) - I(U; Z_3).
\]
Substituting the above two equalities into (83), we have proven (6). The proof of Proposition 1 is thus complete.

\section*{APPENDIX B
PROOF OF PROPOSITION 2}

In this proof, we assume the \((r, d_k)\) PEC is physically degraded with respect to the \((r, d_k)\) PEC for any \(i \leq k\).

By definition, for any \(\epsilon > 0\), let \(n, q, \) and \((g_{s_i}, g_{r_k}, g_{d_k})\) denote the block length, the underlying finite field size, and the corresponding network code satisfying \(\max_{i \in [M]} \text{Prob}(X_i \neq X_i) < \epsilon\). We first notice that for any \(i \in [M]\), \(X_i \rightarrow W_i \rightarrow Z_{i-r} \rightarrow X_i\) form a Markov chain. As a result, we have
\[
I(X_i; X_i) \leq I(W_i; Z_{i-r}) \leq n p_{s_i} \log(q). \tag{84}
\]
On the other hand, by Fano’s inequality, we have
\[
I(X_i; X_i) \geq n R_i(1 - \epsilon) \log(q) - H_2(\epsilon), \tag{85}
\]
where \(H_2(\cdot)\) is the binary entropy function. Jointly (84) and (85) imply that
\[
R_i(1 - \epsilon) - \frac{H_2(\epsilon)}{n \log(q)} \leq p_{s_i} r. \tag{86}
\]
By choosing an infinitesimal \(\epsilon > 0\), the above inequality implies (7).

For any \(\epsilon > 0\) and the corresponding network code, we define \(2^{|M|}\) functions \(f_{S_\epsilon}^{(p)}(\cdot)\) indexed by \(S \in 2^{|M|}\) such that for all \(i \in [M]\),
\[
f_{S_\epsilon}^{(p)}(p_{r;i}) = I(U; Z_{r-i}|Z_{r-i}, X_j : \forall j \in S, \forall l \in [M]\{j\}) \tag{87}
\]
and we interpolate the \(f_{S_\epsilon}^{(p)}(p)\) value for all \(p \neq p_{r;i}\), where
\[
U = \{X_i, Z_{i-r}, 1_{(Z_{r-i})\neq\tau) : \forall i, j \in [M], i \neq j\}
\]
denotes all the information available at the relay \(r\) plus all the information messages at all the sources; \(Z_{r-i}\) denotes the \(n\)-dimensional \((r, d_k)\) PEC output vector; \(\{Z_{r-i} : \forall j \in S, \forall l \in [M]\{j\}\}\) is all the side information received by \(\{d_j : \forall j \in S\}\); and \(\{X_j : \forall j \in S\}\) is the information messages for destination \(d_j, j \in S\). By Proposition 1, \(f_{S_\epsilon}^{(p)}(\cdot)\) must be a ZPCLCIF.

We first notice that when \(S = \emptyset\), by definition we have
\[
f_{\emptyset}^{(p)}(p_{r;i}) = I(U; Z_{r-M}) \leq \frac{n p_{r-M} \log(q)}{n \log(q)} = p_{r-M}. \tag{88}
\]
For notational simplicity, for any \(j \in [M]\), we use \(Z_{r-i} \triangleq \{Z_{r-i} : \forall l \in [M]\{j\}\}\) to denote all the side information overheard by destination \(d_j\). We then notice that for all \(i, k \in [M], S, T \in 2^{|M|}\) satisfying \(k \neq S\) and \(T = S \cup \{k\}\), we have
\[
(n \log(q)) \cdot f_{S_\epsilon}^{(p)}(p_{r;i}) = I(U; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{89}
\]
\[
= I(U; X_i, Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{90}
\]
\[
= I(U; X_i, Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} \in \{S\} \}) \tag{91}
\]
\[
= (n \log(q)) \cdot f_{S_{r-i}}^{(p_{r;i})}
\]
\[
+ I(X_i, Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{92}
\]
where (89) follows from definition, (90) follows from that \((X_i, Z_{S_{k-R-k}})\) is a function of \(U\), (91) follows from the chain rule, and (92) follows from definition. We can further simplify the second term of (92) as follows.
\[
I(X_i, Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{93}
\]
\[
= I(X_i; Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{94}
\]
\[
+ I(Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{95}
\]
\[
= I(X_i; Z_{S_{k-R-k}}; Z_{r-M}|\{X_j, Z_{S_{r-i}} : \forall j \in S\}) \tag{96}
\]
where (94) follows from the information equality \(I(X; Y; Z) = I(X; Y; Z) + I(Y; Z) - I(X; Y)\); (95) follows from that conditioning on \(\{X_j, Z_{S_{r-i}} : \forall j \in S\}\); \(X_k\), the information at source \(s_k\), is independent from the side information \(Z_{S_{k-R-k}} = \{Z_{l-k} : \forall l \in [M]\{k\}\}\) at destination \(d_k\); and (96) follows from the non-negativeness of mutual information.
We then notice that for any $i \leq k$,
\[
I(X_k; Z_{SL-k} Z_{-i} | \{X_j, Z_{SL-j}: \forall j \in S \})
\]
\[= I(X_k; \{Z_{k-j} : \forall j \in S \}| \{X_j : \forall j \in S \})
\]
\[+ I(X_k; Z_{SL-k} Z_{-i} | \{X_j, Z_{SL-j}: \forall j \in S \})
\]
\[= I(X_k; \{Z_{SL-i} : \forall j \in S \}| \{X_j : \forall j \in S \})
\]
\[+ I(X_k; Z_{SL-k} Z_{-i} | \{X_j, Z_{SL-j}: \forall j \in S \})
\]
\[≥ I(X_k; Z_{SL-k} Z_{-i} | \{X_j : \forall j \in S \})
\]
\[≥ I(X_k; Z_{SL-k} Z_{-i} | \{X_j : \forall j \in S \})
\]
\[≥ I(X_k; Z_{SL-k})
\]
\[≥ (nR(k-\epsilon) - H_2(\epsilon) - I(X_k; \{Z_{k-j} : \forall j \in S \}))^+, \]
\[\text{where } (\cdot)^+ \text{ is the projection to non-negative reals. Since}
\]
\[I(X_k; \{Z_{k-j} : \forall j \in S \}) \leq n \log(q)(R_k - \epsilon) - H_2(\epsilon) - I(X_k; \{Z_{k-j} : \forall j \in S \})^+, \]
\[\text{combining (92), (104), and (105) we thus have that } \forall i, k \in [M], \forall S, T \in 2^{[M]} \text{ satisfying } k \notin S \text{ and } T = S \cup \{k\},
\]
\[f_S^{(1)}(p_{r;i}) \geq f_T(p_{r;i})
\]
\[+ 1_{\{i \leq k\}} \left(R_k - \epsilon - \frac{H_2(\epsilon)}{n \log(q)} - p_{k;S} \right)^+
\]
\[\text{For any strictly positive, infinite, decreasing sequence } (\epsilon_1, \epsilon_2, \cdots) \text{ satisfying } \lim_{t \to \infty} \epsilon_t = 0, \text{ we construct}
\]
\[f_S(p) \triangleq \lim_{t \to \infty} \inf \{f_S^{(1)}(p) : \forall \tau \geq t\},
\]
\[\text{for any } S \in 2^{[M]} \text{ as the point-wise infinite limit of } \{f_S^{(1)}(p)\}_{t}. \text{ Obviously, } f_S(\cdot) \geq 0. \text{ Since each } f_S^{(1)}(p) \text{ is ZPPLCIF, it can be shown that } f_S(\cdot) \text{ must be a zero-passing, concave, increasing function. Moreover, since } \lim_{t \to \infty} \epsilon_t = 0, \text{ (88) and (106) ensure that}
\]
\[f_S(p_{r;i}) \geq f_T(p_{r;i}) + 1_{\{i \leq k\}} (R_k - p_{k;S})^+,
\]
\[\text{Such } f_S(\cdot), \forall S \in 2^{[M]} \text{ thus satisfy (8) and (9) in Proposition 2. Note that } f_S(\cdot) \text{ may not be piece-wise linear. On the other hand, we can "linearize" } f_S(\cdot) \text{ by constructing another } f_{S,PL}(\cdot) \text{ from } f_S(\cdot) \text{ satisfying}
\]
\[f_{S,PL}(p_{r;i}) \triangleq f_S(p_{r;i})
\]
\[\text{and by interpolating the } f_{S,PL}(p) \text{ value for all } p \notin \{p_{r;i} : \forall i \in [M]\}. \text{ The resulting } f_{S,PL}(\cdot) \text{ will still satisfy (8) and (9) and is guaranteed to be a ZPPLCIF. The proof of Proposition 2 is thus complete.}
\]
\[\text{APPENDIX C}
\]
\[\text{PROOF OF COROLLARY 3}
\]
\[\text{In this appendix, we prove the alternative form of Proposition 2 for the case of } M = 3. \text{ In this case, we have eight functions } f_S(\cdot) \text{ to consider for } S = \{1, 2, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1\}, \{2\}, \{3\}, \text{ and } \emptyset. \text{ Before discussing each function separately, we introduce the following self-explanatory lemma.}
\]
\[\text{Lemma 8: Consider } p_{1,2}, p_{2,3}, \text{ and } p_{3,1} \text{ satisfying } 0 \leq p_{1,2} < p_{2,3} < p_{3,1} \leq 1. \text{ Consider any ZPPLCIF } f(\cdot) \text{ that satisfies}
\]
\[f(p_{1,2}) \geq a, \quad f(p_{2,3}) \geq b, \quad \text{and } f(p_{3,1}) \geq c,
\]
\[\text{for some } a, b, c \geq 0. \text{ Then due to the concavity and the monotonicity of } f(\cdot), \text{ we must also have}
\]
\[f(p_{1,2}) \geq \max(a, b, c), \quad f(p_{2,3}) \geq \max \left(\frac{p_{2,3}}{p_1}, \frac{p_3}{b}, c\right), \quad \text{and } f(p_{3,1}) \geq \max \left(\frac{p_{3,1}}{p_{2,3}}, a, \frac{p_{3,1}}{p_{3,1}} - \frac{p_{2,3}}{p_3} - \frac{p_{3,1}}{p_{3,1}} - c\right).
\]
\[\text{For the following, we discuss the eight cases separately.}
\]
\[\text{Case 1: } S = \{1, 2, 3\}. \text{ Similar to the } M = 2 \text{ case, we can simply choose } f_{\{1,2,3\}}(p) \triangleq 0 \text{ for all } p \in [0, 1].
\]
\[\text{Case 2: } S = \{1, 2\}. \text{ (8) implies that}
\]
\[f_{\{1,2\}}(p_{r;i}) \geq 0 + (R_3 - p_{3,2})^+.
\]
\[\text{Case 3: } S = \{1, 3\}. \text{ (8) implies that}
\]
\[f_{\{1,3\}}(p_{r;i}) \geq 0 + (R_2 - p_{2,3})^+.
\]
\[\text{Case 4: } S = \{2, 3\}. \text{ (8) implies that}
\]
\[f_{\{2,3\}}(p_{r;i}) \geq 0 + (R_1 - p_{3,2})^+.
\]
Lemma 8 then implies that (108) and (109) can be strengthened and replaced by the following inequalities
\[
\begin{align*}
f_{2}(p_{r;2}) & \geq \frac{pr_{2}}{p_{r;1}} \left( R_{1} - p_{s_{1};1 \cup \{2, 3\}} \right) + \\
f_{2}(p_{r;3}) & \geq \frac{pr_{3}}{p_{r;1}} \left( R_{1} - p_{s_{1};1 \cup \{2, 3\}} \right) + \\
\end{align*}
\]

The discussion of Cases 5 to 8 will use the notation \( R_{r;T} \), which was first introduced in (34).

Case 5: \( S = \{1\} \). We now have two different choices of \( T \): \( T = \{1, 2\} \) and \( T = \{1, 3\} \). By applying (8) for \( T = \{1, 2\} \) and for \( T = \{1, 3\} \), respectively, and by the discussion of Cases 2 and 3, we have
\[
\begin{align*}
f_{1}(p_{r;1}) & \geq \max \left( R_{2;1} + R_{3;1,2}, R_{2;1,3}, R_{3;1} \right) \\
f_{1}(p_{r;2}) & \geq \max \left( R_{2;1} + R_{3;1,2}, R_{2;1,3}, R_{3;1} \right) \\
f_{1}(p_{r;3}) & \geq \max \left( R_{3;1,2}, \frac{pr_{3}}{p_{r;2}} R_{2;1,3} + R_{3;1} \right),
\end{align*}
\]

where each term of the maximum operation corresponds to \( T = \{1, 2\} \) or \( T = \{1, 3\} \), respectively. Lemma 8 then implies that (110) can be strictly strengthened and replaced by the following inequality
\[
\begin{align*}
f_{1}(p_{r;3}) & \geq \\
\max \left( \frac{pr_{3}}{p_{r;2}} \left( R_{2;1} + R_{3;1,2} \right), \frac{pr_{3}}{p_{r;2}} R_{2;1,3} + R_{3;1} \right).
\end{align*}
\]

Case 6: \( S = \{2\} \). We now have two different choices of \( T \): \( T = \{1, 2\} \) and \( T = \{2, 3\} \). By applying (8) for \( T = \{1, 2\} \) and for \( T = \{2, 3\} \), respectively, and by the discussion of Cases 2 and 4, we have
\[
\begin{align*}
f_{2}(p_{r;1}) & \geq \max \left( R_{1;2} + R_{3;1,2}, R_{1;2,3} + R_{3;2} \right) \\
f_{2}(p_{r;2}) & \geq \max \left( R_{3;1,2}, \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{3;2} \right) \\
f_{2}(p_{r;3}) & \geq \max \left( R_{3;1,2}, \frac{pr_{3}}{p_{r;1}} R_{1;2,3} + R_{3;2} \right),
\end{align*}
\]

where each term of the maximum operation corresponds to either \( T = \{1, 2\} \) or \( T = \{2, 3\} \), respectively. Lemma 8 then implies that (111) and (112) can be strictly strengthened and replaced by the following inequalities
\[
\begin{align*}
f_{2}(p_{r;2}) & \geq \\
\max \left( \frac{pr_{2}}{p_{r;1}} \left( R_{1;2} + R_{3;1,2} \right), \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{3;2}, \frac{pr_{2} - pr_{3}}{pr_{1} - pr_{3}} \left( R_{1;2} + R_{3;1,2} \right) \right) \\
+f_{2}(p_{r;3}) & \geq \\
\max \left( \frac{pr_{3}}{p_{r;1}} \left( R_{1;2} + R_{3;1,2} \right), \frac{pr_{3}}{p_{r;1}} R_{1;2,3} + R_{3;2} \right).
\end{align*}
\]

Case 7: \( S = \{3\} \). We now have two different choices of \( T \): \( T = \{1, 3\} \) and \( T = \{2, 3\} \). By applying (8) for \( T = \{1, 3\} \) and for \( T = \{2, 3\} \), respectively, and by the discussion of Cases 3 and 4, we have
\[
\begin{align*}
f_{3}(p_{r;1}) & \geq \max \left( R_{1;3} + R_{2;1,3}, R_{1;2,3} + R_{2;3} \right) \\
f_{3}(p_{r;2}) & \geq \max \left( R_{2;1,3}, \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{2;3} \right) \\
f_{3}(p_{r;3}) & \geq \max \left( \frac{pr_{3}}{p_{r;2}} R_{2;1,3}, \frac{pr_{3}}{p_{r;1}} R_{1;2,3} \right).
\end{align*}
\]

where each term of the maximum operation corresponds to either \( T = \{1, 3\} \) or \( T = \{2, 3\} \), respectively. Lemma 8 then implies that (113) and (114) can be strictly strengthened and replaced by the following inequalities
\[
\begin{align*}
f_{3}(p_{r;2}) & \geq \\
\max \left( \frac{pr_{2}}{p_{r;1}} \left( R_{1;3} + R_{2;1,3} \right), \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{2;3} \right) \\
f_{3}(p_{r;3}) & \geq \\
\max \left( \frac{pr_{3}}{p_{r;1}} R_{1;3} + R_{2;1,3}, \frac{pr_{3}}{p_{r;2}} R_{1;2,3} + R_{2;3} \right).
\end{align*}
\]

Case 8: \( S = \emptyset \). We now have three different choices of \( T \): \( T = \{1\} \), \( \{2\} \), and \( \{3\} \). By applying (8) for all three choices, respectively, we have
\[
\begin{align*}
f_{0}(p_{r;1}) & \geq \max \left( f_{1}(p_{r;1}) + R_{1}, f_{2}(p_{r;1}) + R_{2}, f_{3}(p_{r;1}) + R_{3} \right) \\
f_{0}(p_{r;2}) & \geq \max \left( f_{1}(p_{r;2}) + f_{2}(p_{r;2}) + R_{2}, f_{3}(p_{r;2}) + R_{3} \right) \\
f_{0}(p_{r;3}) & \geq \max \left( f_{1}(p_{r;3}) + f_{2}(p_{r;3}) + f_{3}(p_{r;3}) + R_{3} \right).
\end{align*}
\]

By the discussion of Cases 5 to 7 and by Lemma 8, (115) can be strictly strengthened and replaced by the following inequality
\[
\begin{align*}
f_{0}(p_{r;3}) & \geq \\
\max \left( \frac{pr_{3}}{p_{r;1}} \left( R_{1} + R_{2;1,3} + R_{3;1} \right), \frac{pr_{3}}{p_{r;1}} \left( R_{2} + \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{3;2} \right), \frac{pr_{3}}{p_{r;1}} \left( R_{2} + \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{3;2} \right), \frac{pr_{3}}{p_{r;1}} \left( R_{2} + \frac{pr_{2}}{p_{r;1}} R_{1;2,3} + R_{3;2} \right) \right).
\end{align*}
\]

After further incorporating (7) and the inequality that \( f_{0}(p_{r;3}) \leq pr_{3} \), the proof of Corollary 3 is complete.
APPENDIX D
PROOF OF LEMMA 4

In this appendix, we first prove that all \( \Delta_1 \) to \( \Delta_6 \) are non-negative for all four cases, respectively. We then prove that the inequalities \( L_4 \geq 0 \) and \( \frac{\Delta_1 + \Delta_5}{p_{r;1}} \geq \frac{\Delta_4 + \Delta_5}{p_{r;3}} \) always hold.

**Case 1:** In this case, obviously all \( \Delta_1 \) to \( \Delta_6 \) are non-negative.

**Case 2:** In this case, since \( R_{k_1 - 23} \geq R_{k_3 - 17} \), obviously \( \Delta_1, \Delta_4, \Delta_5, \) and \( \Delta_6 \) are non-negative. By computing \( \Delta_2 = nR_{k_1 - 23} - \Delta_1 \) and \( \Delta_3 = nR_{k_3 - 17} - \Delta_4 \), we also have

\[
\Delta_2 = n \frac{p_{r;1}R_{k_1 - 23} - p_{r;3}R_{k_3 - 23}}{p_{r;1} - p_{r;3}} = \Delta_3.
\]

(116)

Since \( R_{k_3 - 12} \geq \frac{p_{r;3}}{p_{r;1}}R_{k_1 - 23} \), we must have \( \Delta_2 = \Delta_3 \geq 0 \).

**Case 3:** In this case, since \( \frac{p_{r;3}}{p_{r;1}}R_{k_1 - 23} \geq R_{k_3 - 17} \), obviously \( \Delta_1, \Delta_2, \Delta_3, \Delta_4, \) and \( \Delta_5 \) are non-negative. By computing \( \Delta_6 = nR_{k_3 - 17} - \Delta_5 \), we also have

\[
\Delta_6 = n \frac{p_{r;1}(R_{k_3 - 17} + R_{k_3 - 12}) - p_{r;3}(R_{k_1 - 23} + R_{k_3 - 12})}{p_{r;1} - p_{r;3}}
\]

(117)

By condition (40) of Case 3, we must have \( \Delta_6 \geq 0 \).

**Case 4:** In this case, obviously \( \Delta_1 \) to \( \Delta_6 \) are non-negative.

To prove \( L_4 \geq 0 \), we will prove that \( L_1 + L_2 + L_3 \leq n \). Note that \( \Delta_2 \geq 0 \) implies that \( \Delta_1 \leq nR_{k_3 - 17} \), and \( \Delta_6 \geq 0 \) implies that \( \Delta_5 \leq nR_{k_3 - 17} \). We thus will prove that

\[
\frac{L_1 + L_2 + L_3}{n} \leq \max \left( \frac{R_{k_1 - 23} + R_{k_1 - 23} + R_{k_3 - 17}}{p_{r;1}}, \frac{R_{k_2 - 17} + R_{k_3 - 17}}{p_{r;1}} \right) \leq 1.
\]

Equivalently, we will prove that the following two inequalities hold simultaneously.

\[
\frac{R_{k_1 - 23} + R_{k_1 - 23} + R_{k_3 - 17}}{p_{r;1}} + \frac{R_{k_2 - 17} + R_{k_3 - 17}}{p_{r;1}} \leq 1
\]

(118)

\[
\frac{R_{k_1 - 23} + R_{k_1 - 23} + R_{k_3 - 12} + R_{k_1 - 23} + R_{k_3 - 12}}{p_{r;1}} \leq 1
\]

(119)

Using (35) to (38), (118) is a straightforward result of (26) (using the second term of the max operation), and (119) is a straightforward result of (25) (using any of the two terms of the max operation).

Inequality (42) can be proven by plugging the values of \( \Delta_1, \Delta_4, \Delta_5 \) for different cases. In Case 1, \( \Delta_1 = \Delta_4 = \Delta_5 = 0 \) and (42) is satisfied. In Cases 2 and 3, plugging the values gives us

\[
\frac{\Delta_1 + \Delta_5}{p_{r;1}} = \frac{\Delta_4 + \Delta_5}{p_{r;3}}.
\]

In Case 4, (42) is guaranteed due to the condition of Case 4.

The proof of Lemma 4 is thus complete.

APPENDIX E
DECODING AT D_2 FOR THE CASE OF M = 3

By noticing \( L_1 + L_2 + L_3 + L_4 = n \), we can rewrite \( p_{r;2}(L_3 + L_4) - nc_{G2} \) by

\[
p_{r;2}(L_3 + L_4) - nc_{G2} = n \cdot \left( p_{r;2} - \frac{p_{r;2}}{p_{r;1}}R_{k_1 - 23} - R_{k_2 - 13} - R_{k_2 - 17} \right)
\]

\[
- \frac{p_{r;2}}{p_{r;1}}(\Delta_1 + \Delta_5).
\]

(120)

To prove (58), we will prove an equivalent statement that (120) is no less than (56). Since the values of \( \Delta_1 \) to \( \Delta_6 \) are chosen depending on the values of \( R_{k_1 - 23}, R_{k_3 - 17}, \) and \( R_{k_3 - 17} \) (Cases 1 to 4), we will establish the inequality for each case separately.

**Case 1:** \( R_{k_3 - 17} \leq R_{k_1 - 23} \). In this case, we have \( \Delta_1 = \Delta_4 = \Delta_5 = 0 \) and \( \Delta_2 = nR_{k_1 - 23}, \Delta_3 = nR_{k_3 - 17}, \) and \( \Delta_6 = nR_{k_3 - 17} \). Therefore (120) becomes

\[
n \cdot \left( p_{r;2} - \frac{p_{r;2}}{p_{r;1}}R_{k_1 - 23} - R_{k_2 - 13} - R_{k_2 - 17} \right),
\]

and (56) becomes

\[
n \cdot (R_{k_2 - 13} + R_{k_2 - 13} + R_{k_3 - 12} + R_{k_3 - 17}).
\]

The inequality (120) \( \geq (56) \) is thus established by (35) to (38) and by the first term of the max operation in (26) of Corollary 3.

**Case 2:** \( R_{k_1 - 23} \geq R_{k_3 - 17} \geq \frac{p_{r;3}}{p_{r;1}}R_{k_1 - 23} \). In this case, the \( \Delta_1 \) to \( \Delta_5 \) values are decided in (39) and (116), and \( \Delta_6 = R_{k_3 - 17} \). Therefore (120) becomes

\[
n \cdot \left( p_{r;2} - \frac{p_{r;2}}{p_{r;1}}R_{k_1 - 23} - R_{k_2 - 13} - R_{k_2 - 17} \right)
\]

\[
- \frac{p_{r;2}}{p_{r;1}}(R_{k_1 - 23} - R_{k_3 - 17})
\]

(120)

and (56) becomes

\[
n \cdot (R_{k_2 - 13} + R_{k_2 - 13} + R_{k_3 - 12} + R_{k_3 - 17}).
\]

The inequality (120) \( \geq (56) \) is thus established by (35) to (38) and by the third term of the max operation in (26) of Corollary 3.

**Case 3:** \( \frac{p_{r;3}}{p_{r;1}}R_{k_1 - 23} \geq R_{k_3 - 17} \) and

\[
R_{k_3 - 17} \geq \frac{p_{r;3}}{p_{r;1}}(R_{k_1 - 23} + R_{k_3 - 17}).
\]

In this case, the values of \( \Delta_1, \Delta_4, \Delta_5, \) and \( \Delta_6 \) values are decided in (41) and (117), and \( \Delta_2 = \Delta_3 = 0 \). Therefore (120) becomes

\[
n \cdot \left( p_{r;2} - \frac{p_{r;2}}{p_{r;1}}R_{k_1 - 23} - R_{k_2 - 13} - R_{k_2 - 17} \right)
\]

\[
- \frac{p_{r;2}}{p_{r;1}}(R_{k_1 - 23} + \frac{p_{r;3}R_{k_1 - 23} - p_{r;3}R_{k_3 - 17}}{p_{r;1} - p_{r;3}})
\]
and (56) becomes
\[ n \cdot \left( R_{k_2-3} + R_{k_2-13} + \frac{p_{r:1}}{p_{r:2}} \right) - \frac{p_{r:2}}{p_{r:1}} \left( R_{k_2-3} + R_{k_2-13} + \frac{p_{r:3}}{p_{r:2}} \left( R_{k_1-2} + R_{k_1-13} - R_{k_2-12} - R_{k_2-13} \right) \right) \]
\[ - \frac{p_{r:2}}{p_{r:1}} \left( R_{k_2-3} + R_{k_2-13} - R_{k_2-13} - R_{k_2-12} \right) \]
\[ n \cdot \left( R_{k_2-3} + R_{k_2-13} + \frac{p_{r:2}}{p_{r:1}} \left( R_{k_2-3} + R_{k_2-13} - R_{k_2-12} - R_{k_2-13} \right) \right) \]
and (56) becomes \[ n \cdot (R_{k_2-3} + R_{k_2-13}) \] and (56) is thus established by (35) to (38) and by the third term of the max operation in (26) of Corollary 3.

Case 4: \[ \frac{p_{r:3}}{p_{r:2}} \left( R_{k_1-2} + R_{k_2-4} + R_{k_1-13} + R_{k_2-12} \right) \geq R_{k_2-3} + R_{k_2-13} + R_{k_2-12} \] In this case, we have \[ \Delta_1 = nR_{k_1-2} + R_{k_2-4} \] \[ \Delta_4 = nR_{k_1-2} + R_{k_2-12} \] \[ \Delta_5 = nR_{k_2-13} \] \[ \Delta_6 = 0 \] Therefore (120) becomes
\[ n \cdot \left( \frac{p_{r:2}}{p_{r:1}} R_{k_1-2} + R_{k_2-12} - \frac{p_{r:2}}{p_{r:1}} \left( R_{k_1-2} + R_{k_2-13} - R_{k_2-12} - R_{k_2-13} \right) \right) \]
and (56) becomes \[ n \cdot (R_{k_2-3} + R_{k_2-13}) \] and (56) is thus established by (35) to (38) and by the second term of the max operation in (26) of Corollary 3.

APPENDIX F
PROOF OF LEMMA 5

Proof: In Step 2, we first notice that for any \( z \) values that satisfy (16) with strict inequality, we can repeatedly reduce the values of any \( z \) variables until (16) becomes an equality. The newly reduced \( z \) variables will still satisfy (14) and (15). Therefore, we can replace inequality (16) by an equality without loss of generality.

For any \( i \in [M], T_i \subseteq [M] \setminus \{i, T_2 = [M] \setminus (T_1 \cup \{i\}) \), consider the following set of coding vectors:
\[ \{ c_i(t) \} : \forall j \in [M], 1 \{ z_{j-1}(t) \neq \} = 1 \{ j \in T_i \} \]
i.e., the set of coding vectors that are overheard by and only by users in \( T_1 \). Since \( z_{j-1}(t) \) satisfies (15), we can construct the subspace \( V_{i-1} \setminus T_2 \) as the linear span of the first \( n_z \) vectors from the above set of coding vectors. By this construction, the MDS property of RLNC, and by (16), properties (i), (ii), and (iii) of \( V_{i-1} \setminus T_2 \) are satisfied. The proof is complete. □

APPENDIX G
PROOFS OF LEMMA 6 AND COROLLARY 4

Proof of Lemma 6: Since \( S_{r_i} = [M] \) for any permutation \( \pi \), we have \( x_{\pi, [M], h} = x_{\pi, S_{r_i}, h} \). We also notice that \( x_{\pi, S_{r_i}, h} \) only participates in (63) when \( j = 1 \). Since \( x_{\pi, S_{r_i}, h} \) is on the right-hand side of (63) when \( j = 1 \), for any \( x_{\pi, S_{r_i}, h} \) that is strictly positive, we can reset its value to zero while the new \( x \) values still satisfy (63). Therefore, we can set any strictly positive \( x_{\pi, [M], h} \) value to zero without loss of generality. The first part of Lemma 6 is proven.

To prove the second part of the lemma, we use the results in Lemma 1. Recall that we assume \( p_r < p_{r-1} > p_{r-2} > \cdots > p_{r-1}M > 0 \) for any ZPPLCIF \( f_{\pi, S_{r_i}} (\cdot) \) that corresponds to \( \{ x_{\pi, S_{r_i}, h} : \forall h \in [M] \} \) through (28), we first notice that \( \sum_{h=1}^{k} x_{\pi, S_{r_i}, h} \) is the first order derivative \( \frac{d}{dp} f_{\pi, S_{r_i}} (p) \) for all \( p \in (p_{r-1} + p_{r-1}k) \), using the convention \( p_{r-1}M + 1 = 0 \). We then prove the second part by contradiction.

Assume that (68) does not hold for some \( (\pi, k_0, j) \). Then for such \( (\pi, k_0, j) \), we have
\[ \forall p \in (p_{r-1} + p_{r-1}k_0), \quad \frac{d}{dp} f_{\pi, S_{r_i}} (p) < \frac{d}{dp} f_{\pi, S_{r_i}} (p) \] (121)
Define
\[ k^* = \inf \{ i : \{ k_0 + 1, k_0 + 2, \cdots, M \} : \] \[ \frac{d}{dp} f_{\pi, S_{r_i}} (p) \neq \frac{d}{dp} f_{\pi, S_{r_i}} (p) \] where \( p_{r-1} \Delta = p_{r-1} - \epsilon \) and \( p_{r-1} \Delta = p_{r-1} + \epsilon \) for some sufficiently small \( \epsilon > 0 \) are the left and right limits of \( p_{r-1} \), respectively. Namely, \( p_{r-1} \) is the largest \( p \in (0, p_{r-1}) \) at which the slope of \( f_{\pi, S_{r_i}} (\cdot) \) changes. There are two cases to discuss depending on whether such \( k^* \) exists or not.

Case 1: \( k^* \) exists. In this case, since \( f_{\pi, S_{r_i}} (\cdot) \) is a ZPPLCIF, we must have
\[ \frac{d}{dp} f_{\pi, S_{r_i}} (p_{r-1}k^*) < \frac{d}{dp} f_{\pi, S_{r_i}} (p_{r-1}k^*) \] (122)
Without loss of generality, we can also assume that
\[ f_{\pi, S_{r_i}} (p_{r-1}k^*) = f_{\pi, S_{r_i}} (p_{r-1}k^*) + \] \[ \max_{\forall k \in [M], k \neq k^*} \left\{ \sum_{\forall T \subseteq [M], T \subseteq S_{r_i}} y_{k, T, (\pi, S_{r_i})} \right\} \] (123)
i.e., (13) holds with equality (using one of the \( k \) values) when evaluated at \( p \Delta = p_{r-1}k^* \). The reason behind this assumption is as follows. For any \( \epsilon > 0 \), we define a function \( g_{\epsilon}(\cdot) \) by setting \( g_{\epsilon}(p) = f_{\pi, S_{r_i}} (p) \) for all \( p \in (1, \cdots, p_{r-1}M, 0) \) and setting \( g_{\epsilon}(p_{r-1}k^*) = f_{\pi, S_{r_i}} (p_{r-1}k^*) - \epsilon \) and by interpolating \( g_{\epsilon}(p) \) values for all \( p \) values not in \( (1, \cdots, p_{r-1}M, 0) \). Suppose the equality (123) does not hold. Then by (122), there exists a sufficiently small \( \epsilon > 0 \) such that
\[ g_{\epsilon}(p_{r-1}k^*) = f_{\pi, S_{r_i}} (p_{r-1}k^*) - \epsilon \] \[ \geq f_{\pi, S_{r_i}} (p_{r-1}k^*) + \] \[ \max_{\forall k \in [M], k \neq k^*} \left\{ \sum_{\forall T \subseteq [M], T \subseteq S_{r_i}} y_{k, T, (\pi, S_{r_i})} \right\} \] and
\[ \frac{d}{dp} f_{\pi, S_{r_i}} (p_{r-1}k^*) < \frac{d}{dp} g_{\epsilon}(p_{r-1}k^*) \] \[ \leq \frac{d}{dp} g_{\epsilon}(p_{r-1}k^*) < \frac{d}{dp} f_{\pi, S_{r_i}} (p_{r-1}k^*) \] Therefore \( g_{\cdot}(\cdot) \) is also a ZPPLCIF. Note that \( g_{\cdot}(\cdot) \) is point-wise no larger than \( f_{\pi, S_{r_i}} (\cdot) \) for all \( p \in (0, 1) \); is strictly smaller than \( f_{\pi, S_{r_i}} (\cdot) \) when evaluated at \( p_{r-1}k^* \); and satisfies the inequality (13). We can thus replace \( f_{\pi, S_{r_i}} (\cdot) \) by the newly constructed \( g_{\cdot}(\cdot) \) while still satisfying all the inequalities (13). By repeatedly applying the above construction of replacing \( f_{\pi, S_{r_i}} (\cdot) \) with a point-wise smaller \( g_{\cdot}(\cdot) \), we have thus shown that (123) must hold when taking the point-wise limit of the functions of the above repeated construction.
a result, we can assume that (123) holds without loss of generality.

Now we compare the two terms \( (f_{\pi,S^+_j}(p_{r;k^*}) - f_{\pi,S_j^+}(p_{r;k^*})) \) versus \( (f_{\pi,S^+_j}(p_{r;k_0}) - f_{\pi,S_j^+}(p_{r;k_0})) \). We first note that by our construction, the slope of \( f_{\pi,S^+_j}(\cdot) \) remains unchanged for all \( p \in (p_{r;k^*}, p_{r;k_0}) \). Since \( f_{\pi,S_j^+}(\cdot) \) is a ZPPLCIF, we must also have
\[
\frac{d}{dp} f_{\pi,S_j^+}(p) \geq \frac{d}{dp} f_{\pi,S_j^+}(p_{r;k_0})
\]
for all \( p \in (p_{r;k^*}, p_{r;k_0}) \), where the strict inequality follows from the assumption (121). As a result, we must have
\[
f_{\pi,S^+_j}(p_{r;k_0}) - f_{\pi,S_j^+}(p_{r;k_0}) < f_{\pi,S^+_j}(p_{r;k^*}) - f_{\pi,S_j^+}(p_{r;k^*})
\]
which contradicts the assumption (121). As a result, (68) holds in Case 1.

Case 2: \( k^* \) does not exist. In this case, we have \( \frac{d}{dp} f_{\pi,S^+_j}(p_{r;k_0}) = \frac{d}{dp} f_{\pi,S_j^+}(0^+) \). By noting that both \( f_{\pi,S^+_j}(\cdot) \) and \( f_{\pi,S_j^+}(\cdot) \) are ZPPLCIFs and \( f_{\pi,S^+_j}(\cdot) \) is pointwise no smaller than \( f_{\pi,S_j^+}(\cdot) \), we must also have
\[
\frac{d}{dp} f_{\pi,S^+_j}(0^+) \geq \frac{d}{dp} f_{\pi,S_j^+}(0^+) \geq \frac{d}{dp} f_{\pi,S_j^+}(p_{r;k_0}).
\]

Jointly we thus have
\[
\frac{d}{dp} f_{\pi,S^+_j}(p_{r;k_0}) \geq \frac{d}{dp} f_{\pi,S_j^+}(p_{r;k_0}),
\]
which contradicts the assumption (121). As a result, (68) holds in Case 2 as well. The proof of Lemma 6 is complete.

Proof of Corollary 4: We prove this corollary by explicit construction for each permutation \( \pi \).

Consider a permutation \( \pi \) that is fixed throughout the proof. We will first construct a partition \( \{\chi_{\pi,l} : \forall l\} \) and the corresponding label function \( \{\theta_{\pi,l}(\cdot) : \forall l\} \) that satisfies (69) for \( j = 1 \). Then we will iteratively refine the partition until (69) is satisfied for all \( j \in [M] \). More explicitly, we perform the following steps sequentially.

Step 1: We construct a partition of \( (M + 1) \) chunks in this step. For \( l = 0 \) to \( M \), choose \( (\chi_{\pi,l}, \theta_{\pi,l}(\cdot)) \) as follows:
\[
\begin{align*}
\chi_{\pi,l} &= x_{\pi,S_j^+}^l, \\
\theta_{\pi,l}(k) &= \begin{cases} 
1 & \text{if } k = 1 \\
-1 & \text{if } k \in [M] \setminus 1
\end{cases}
\end{align*}
\]
where the value \(-1\) indicates that the value of \( \theta_{\pi,l}(k) \) has not been assigned for \( k \geq 2 \). For \( l = M + 1 \), choose
\[
\begin{align*}
\chi_{\pi,l} &= \left( \sum_{h=1}^{M} x_{\pi,h} \right) - \left( \sum_{h=1}^{M} x_{\pi,S_h^+} \right), \\
\theta_{\pi,l}(k) &= \begin{cases} 
M + 1 & \text{if } k = 1 \\
-1 & \text{if } k \in [M] \setminus 1
\end{cases}
\end{align*}
\]
Eq. (68) of Lemma 6 guarantees the non-negativeness of \( \chi_{\pi,M+1} \). After Step 1, the partition contains \( (M + 1) \) different chunks. It can be easily verified that (69) is satisfied for \( j = 1 \) and \( \theta_{\pi,j}(\cdot) \) is non-increasing.

We perform the following “Step \( j_0 \)” sequentially for \( j_0 = 2, \cdots, M \). We use the induction assumption that after finishing the \( (j_0 - 1) \)-th step, the latest partition \( \{\theta_{\pi,j}(\cdot) : \forall l\} \) satisfies (69) for all \( j \leq j_0 - 1 \). For Step \( j_0 \), perform the following routine:

1. Set \( k \leftarrow 0 \).
2. for \( h = 1 \) to \( M \) do
3. do
4. while \( \Delta > 0 \) do
5. if there exists a chunk \( (\chi_{\pi,l}, \theta_{\pi,l}(\cdot)) \) satisfying \( \theta_{\pi,l}(j_0 - 1) = k \) and \( \theta_{\pi,l}(j_0) = -1 \) then
6. Choose one such chunk \( (\chi_{\pi,l}, \theta_{\pi,l}(\cdot)) \).
7. if \( \Delta \geq \chi_{\pi,l} \) then
8. Set \( \theta_{\pi,l}(j_0) \leftarrow h \) and \( \Delta \leftarrow (\Delta - \chi_{\pi,l}) \).
9. else
10. Replace the chosen chunk \( (\chi_{\pi,l}, \theta_{\pi,l}(\cdot)) \) by two new sub-chunks \( (\chi_{\pi,l_1}, \theta_{\pi,l_1}(\cdot)) \) and \( (\chi_{\pi,l_2}, \theta_{\pi,l_2}(\cdot)) \) such that
11. \( \chi_{\pi,l_1} = \Delta \)
12. \( \theta_{\pi,l_1}(k) = \theta_{\pi,l}(k), \forall k \in [M] \)
13. and
14. \( \chi_{\pi,l_2} = \chi_{\pi,l} - \Delta \)
15. \( \theta_{\pi,l_2}(k) = \theta_{\pi,l}(k), \forall k \in [M] \).
16. end if
17. end if
18. else
19. Set \( k \leftarrow k + 1 \).
20. end if
21. end while
22. end for
23. for all chunks \( (\chi_{\pi,l}, \theta_{\pi,l}(\cdot)) \) satisfying \( \theta_{\pi,l}(j_0) = -1 \) do
24. Set \( \theta_{\pi,l}(j_0) \leftarrow M + 1 \).
25. end for
The correctness of the above algorithm is proven as follows. In the beginning when \( h = 1 \), we notice that all chunks have been assigned in Step \((j_0 - 1)\) and thus have \( \theta_{\pi,l}(j_0 - 1) \neq -1 \). Therefore, all chunks have \( \theta_{\pi,l}(j_0 - 1) \geq 1 \). When assigned during the case of \( h = 1 \), and their newly assigned value is \( \theta_{\pi,l}(j_0) = h = 1 \). Therefore, the label function \( \theta_{\pi,l}(\cdot) \) remains non-increasing when \( h = 1 \). Note that Lines 7 to 12 also guarantee that when the WHILE loop moves to \( h = 2 \) for the first time, the equality of (69) is satisfied for \((j, h) = (j_0, 1)\).

Consider the scenario in which we move to \( h = h_0 \) for some \( h_0 \in [2, M] \) for the first time and assume by induction that the equality of (69) holds for all \((j, h)\) satisfying either (i) \( j = j_0 \) and \( h \leq h_0 - 1 \), or (ii) \( j \leq j_0 - 1 \) and \( h \in [M] \). Eq. (68) of Lemma 6 and the induction assumption then guarantees that

\[
\sum_{k'=1}^{h_0-1} \left( \sum_{\forall \ell: \theta_{\pi,l}(j_0-1)=k'} \chi_{\pi,l} \right) = \sum_{k'=1}^{h_0-1} \left( \sum_{\forall \ell: \theta_{\pi,l}(1)=k'} \chi_{\pi,l} \right) \leq \sum_{k'=1}^{h_0-1} \left( \sum_{\forall \ell: \theta_{\pi,l}(0)=k'} \chi_{\pi,l} \right).
\]

Therefore, when we move to \( h = h_0 \) for the first time, we must have exhaustively processed all chunks with \( \theta_{\pi,l}(j_0-1) = k' \), \( k' = 1, \ldots, h_0 - 1 \). Hence we must have \( k \geq h_0 \). As a result, all the \( \theta_{\pi,l} \) functions that are assigned during the period of \( h = h_0 \) must have \( \theta_{\pi,l}(j_0-1) = k \geq h_0 \) and thus remain non-increasing since their newly assigned value is \( \theta_{\pi,l}(j_0) = h_0 \). Suppose \( h_0 < M \). Lines 7 to 12 also guarantee that when we move to \( h = h_0 + 1 \) for the first time, (69) is satisfied for \( j = j_0 \) and \( h = h_0 \).

Consider the scenario when we finish the last iteration of the FOR loop with \( h = M \). By (126) and (127), the total sum of all \( \chi_{\pi,l} \) is

\[
\sum_{k'=1}^{M+1} \left( \sum_{\forall \ell: \theta_{\pi,l}(j_0-1)=k'} \chi_{\pi,l} \right) = \sum_{k'=1}^{M+1} \left( \sum_{\forall \ell: \theta_{\pi,l}(1)=k'} \chi_{\pi,l} \right) = \sum_{k'=1}^{M} \left( \sum_{\forall \ell: \theta_{\pi,l}(0)=k'} \chi_{\pi,l} \right) \leq \sum_{k'=1}^{M} \sum_{\forall \ell: \theta_{\pi,l}(0)=k'} \chi_{\pi,l} = \sum_{k'=1}^{M} \sum_{\forall \ell: \theta_{\pi,l}(0)=k'} \chi_{\pi,l} \leq \sum_{k'=1}^{M} \sum_{\forall \ell: \theta_{\pi,l}(0)=k'} \chi_{\pi,l},
\]

(128)

where the inequality follows from repeatedly using (68). Therefore, when \( \Delta = 0 \) for the iteration of \( h = M \), the \( k \) value is still bounded within \([M+1]\). On the other hand, by the induction assumption, we also have

\[
\sum_{k'=1}^{M} \left( \sum_{\forall \ell: \theta_{\pi,l}(j_0-1)=k'} \chi_{\pi,l} \right) = \sum_{k'=1}^{M} \sum_{\forall \ell: \theta_{\pi,l}(j_0-1)=k'} \chi_{\pi,l} \leq \sum_{k'=1}^{M} \sum_{\forall \ell: \theta_{\pi,l}(j_0-1)=k'} \chi_{\pi,l}.
\]

As a result, when we finish the last iteration (with \( h = M \) and \( \Delta = 0 \)), we have processed all chunks \((\chi_{\pi,l}, \theta_{\pi,l}(\cdot))\) with \( \theta_{\pi,l}(j_0-1) \leq M \). Therefore, when we move to the "clean-up" stage: Lines 18 to 20, all the remaining chunks have \( \theta_{\pi,l}(j_0 - 1) = M + 1 \). Therefore the non-increasing property of \( \theta_{\pi,l}(\cdot) \) still holds after the clean-up stage. Lines 7 to 12 then guarantee that (69) is satisfied for all \( j \leq j_0 \) and \( h \in [M] \).

When \( j_0 = M, (128) \) becomes an equality. Therefore, when we have finished the last iteration (with \( h = M \) and \( \Delta = 0 \)) of \( j_0 = M \), we must have processed all the chunks. Line 19 will not be executed. As a result, \( \theta_{\pi,l}(M) \leq M \) for all \( \pi \) and \( l \). The above construction of \( \{ \chi_{\pi,l} : \forall \ell \} \) completes the proof of Corollary 4.

**APPENDIX H**

**A MORE GENERAL NON-CANONICAL SETTING**

In this appendix, we generalize the canonical setting in Section II to take into account both opportunistic routing and cross-layer optimization, which further enlarge the capacity of erasure-channel-based 2-staged INC.

We consider the following 1-hop relay network, which consists of \( M \) source/destination pairs \((s_i, d_i)\), \( \forall i \in [M] \) and a common relay node \( r \). Source \( s_i \) would like to send \( nR_i \) information packets \( X_{i,1} \rightarrow X_{i,nR_i} \rightarrow d_i \) through the common relay \( r \) for some positive integer \( n \) and fractional rate \( R_i \). Each \( s_i \) is associated with a 1-to-(\( M + 1 \)) PEC that sends a symbol \( W_i \in GF(q) \) from \( s_i \) to \( \{ d_j : \forall j \in [M] \} \cap \{ r \} \). Namely, we allow direct communication between \( s_i \) and \( d_i \) for each \( s_i \), the outputs of its 1-to-(\( M + 1 \)) PEC are denoted by \( Z_{i, \rightarrow j} \), \( j \in [M] \) and \( Z_{i, \rightarrow r} \), which refer to the outputs arriving at destination \( d_j \) and the relay \( r \), respectively. The common relay \( r \) is associated with a 1-to-\( M \) PEC that sends a symbol \( W_r \in GF(q) \) from \( r \) to \( \{ d_j : \forall j \in [M] \} \), and the corresponding outputs are denoted by \( Z_{r, \rightarrow j} \) for all \( j \in [M] \).

We also assume that the sources \( s_i \) and the relay \( r \) share the time resources (potentially due to underlying interference constraints). More specifically, consider a set of \((M + 1)\) time allocation variables \((\beta_1, \ldots, \beta_M, \beta_r)\) that satisfy \( \beta_r + \sum_{i=1}^{M} \beta_i = M + 1 \). Each of the sources \( s_i \) to \( s_M \) and the relay \( r \) can use their associated PECs for exactly \( n\beta_i \) and \( n\beta_r \) times.

We use the following vector notation to denote the \((M + 1)n\) transmitted symbols and their corresponding outputs. For all \( i \in [M] \), we define

\[
W_{i,r} = (W_{i,1}(1), W_{i,2}(2), \ldots, W_{i,n}(n\beta_i)).
\]

\[
Z_{i, \rightarrow j} = (Z_{i, \rightarrow j}(1), Z_{i, \rightarrow j}(2), \ldots, Z_{i, \rightarrow j}(n\beta_i))
\]

\[
W_{r} = (W_{r,1}(1), W_{r,2}(2), \ldots, W_{r,n}(n\beta_r)).
\]

\[
Z_{r, \rightarrow j} = (Z_{r, \rightarrow j}(1), Z_{r, \rightarrow j}(2), \ldots, Z_{r, \rightarrow j}(n\beta_r)),
\]

and for all \( i, j \in [M] \), we define

\[
Z_{i, \rightarrow j} = (Z_{i, \rightarrow j}(1), Z_{i, \rightarrow j}(2), \ldots, Z_{i, \rightarrow j}(n\beta_j)),
\]

for which we use the input argument \( \langle t \rangle \), to distinguish each channel usage. All the PECs are memoryless and stationary and described by the joint success probabilities \( p_{rs} \) and \( p_{s \rightarrow s} \), for all \( i \in [M] \) as discussed in Section II-A. The channel statistics \( p_{rs} \) and \( p_{s \rightarrow s} \) are assumed to be known to all network nodes. An illustration of this general 1-hop relay network with direct path is illustrated in Fig. 9 for the case of \( M = 2 \).

The **channel status vector** of the PEC outputs \( Z_{i, \rightarrow j} \) is defined by

\[
1_{\{Z_{i, \rightarrow j}(\cdot) \neq x\}} \triangleq (1_{\{Z_{i, \rightarrow j}(1) \neq x\}}, \ldots, 1_{\{Z_{i, \rightarrow j}(n\beta_j) \neq x\}}).
\]
A 2-staged 1-hop intersession network code is then defined by \((M + 1)\) encoding functions \(g_s(\cdot)\) and \(g_r(\cdot)\):

\[
\forall i \in [M], \quad W_i = g_s(X_i)
\]

\[
W_r = g_r(\{Z_i \rightarrow r, 1_{(Z_i \rightarrow r)(\cdot) \neq 1} : \forall i, j \in [M]\}),
\]

and \(M\) decoding functions \(g_d(\cdot)\):

\[
\forall j \in [M], \quad \hat{X}_j = g_d(\{Z_i \rightarrow j, Z_r \rightarrow j : \forall i \in [M]\}).
\]

The encoding and decoding functions can be linear or nonlinear.

Assume that the information symbols \(X_i = (X_{1i}, \cdots, X_{ni})\) are uniformly and independently distributed over GF(\(q\)) for all \(i \in [M]\). We define the achievability of any 2-staged 1-hop intersession network code as follows.

**Definition 6:** Given the values of the time allocation variables \((\beta_1, \cdots, \beta_M, \beta_r)\), a rate vector \((R_1, \cdots, R_M)\) is achievable if for any \(\epsilon > 0\), there exists a 1-hop intersession network code satisfying

\[
\forall i \in [M], \quad P(\hat{X}_i \neq X_i) < \epsilon,
\]

for sufficiently large \(n\) and sufficiently large underlying finite field GF(\(q\)).

**Definition 7:** The capacity region of a 2-staged 1-hop relay network of \(M\) sessions is defined as the closure of all achievable rate vectors \((R_1, \cdots, R_M)\) of all possible assignment of the time allocation variables \((\beta_1, \cdots, \beta_M, \beta_r)\).

Note that the previous setting in Section II is a special case of the above new setting by choosing the success probabilities \(p_{s,i} \rightarrow i\) with zero marginal probabilities \(p_{s,i} \rightarrow i = 0\) for all \(i \in [M]\), and by considering only uniform time allocation variables \((\beta_1, \cdots, \beta_M, \beta_r) = (1, \cdots, 1)\). The outer and inner bounds in Section III can then be generalized as follows, for which we assume that \(p_{r,1} > p_{r,2} > \cdots > p_{r,M} > 0\).

**Proposition 7:** A rate vector \((R_1, \cdots, R_M)\) is in the capacity region only if there exist \((M + 1)\) time allocation variables \((\beta_1, \cdots, \beta_M, \beta_r)\) with the total sum being \((M + 1)\), and \(2^M\) PPLCIFs \(\{f_S(\cdot) : \forall S \in 2^{[M]}\} \) indexed by \(S \in 2^{[M]}\) such that the following \((M + M^22^{M-1} + 1)\) inequalities are satisfied:

\[
\forall i \in [M], \quad R_i \leq \beta_1 \cdot p_{s,i \cup (d, r)} \quad (129)
\]

\[
\forall i, k \in [M], \forall S, T \in 2^{[M]} \text{ satisfying } k \notin S \text{ and } T = S \cup \{k\},
\]

\[
f_S(p_{r,i}) \geq f_T(p_{r,i}) + \sum_{1 \leq k \leq i} (R_k - \beta_k p_{s,k \cup (d, T)}) + f_0(p_{r,M}) \quad (130)
\]

and

\[
\beta_r \leq \sum_{k \in S} p_{r,M} \quad (131)
\]

**Proposition 8:** A rate vector \((R_1, \cdots, R_M)\) can be achieved by a linear network code if there exist \((M + 1)\) time allocation variables \((\beta_1, \cdots, \beta_M, \beta_r)\) with the total sum being \((M + 1)\), \(M!(2^{M+1} - M - 2)\) non-negative variables:

\[
\{y_k, T, (\pi, S) : \forall \pi, \forall j \in [M + 1]\},
\]

and \(M2^{M-1}\) non-negative variables:

\[
\{z_k, T \geq 0 : \forall T \in 2^{[M]}, \forall k \in [M] \setminus T\},
\]

and \((M + 1)!\) PPLCIFs \(\{f_{\pi, S}^{k} : \forall \pi, \forall j \in [M + 1]\}\), such that jointly the following \((M + M^3 + M^2 - M - 2)\) inequalities are satisfied:

\[
\forall i \in [M], \quad R_i < \beta_1 \cdot p_{s,i \cup (d, r)} \quad (132)
\]

\[
\forall \pi, \forall i, j, k \in [M], \quad f_{\pi, S}^{k}(p_{r,i}) \geq f_{\pi, S}^{k}(p_{r,i}) + 1_{\{i < k, k \notin S\}} + \sum_{T \in 2^{[M]} : T \subseteq S} y_{k, T, (\pi, S)_{k, T}} \quad (133)
\]

\[
\forall k \in [M], \forall T \in 2^{[M]}, k \notin T,
\]

\[
\sum_{\forall \pi, \forall j \in [M + 1] : S_j \supseteq T, k \notin S_j} y_{k, T, (\pi, S)_{k, T}} \geq z_{k, T} \quad (134)
\]

\[
\forall S \in 2^{[M]}, \forall k \in [M] \setminus S, \forall T = S \cup \{k\},
\]

\[
z_{k, S} \leq \beta_k \cdot p_{s,k \cup (\{k\} \cup (\pi, S))} \quad (135)
\]

\[
\forall k \in [M], \sum_{S \subseteq 2^{[M]} : k \notin S} z_{k, S} \geq (R_k - \beta_k \cdot p_{s,k \cup (d, r)})^+ \quad (136)
\]

\[
\sum_{\forall \pi} f_{\pi, s}(p_{r,M}) < \beta_r \cdot p_{r,M} \quad (137)
\]

Both Propositions 7 and 8 can be expressed as a LP feasibility problem with the help of Lemma 1, in a similar way as in Propositions 5 and 6. The proofs of Propositions 7 and 8 are provided in Appendix I.

**APPENDIX I**

**PROOFS OF PROPOSITIONS 7 AND 8**

**Proof of Proposition 7:** This proof follows closely the proof of Proposition 2 in Appendix B.

To prove (129), we first notice that for any \(i \in [M]\), \(X_i \rightarrow W_i \rightarrow (Z_i \rightarrow r, Z_i \rightarrow i) \rightarrow \hat{X}_i\) form a Markov chain. As a result, we have

\[
I(X_i; \hat{X}_i) \leq I(W_i; Z_i \rightarrow r, Z_i \rightarrow i) \leq n_i \cdot p_{s,i \cup (d, r)} \log(q).
\]

Following the same remaining steps as in (85) and (86), we can thus prove (129).
The remaining proof again closely follows that of Proposition 2. The main difference is that for any $\epsilon > 0$ and the corresponding network code, we define a new set of $2^M$ functions $f_{S}^{[\epsilon]}(\cdot)$ by

$$
\frac{f_{S}^{[\epsilon]}(p_{r;i})}{I(U;Z_{r\rightarrow i}^{[\epsilon]}(S_{r\rightarrow j}^{[\epsilon]} \mid j \in \mathcal{S}_{\text{self}})} - \frac{n \log(q)}{n \log(q)} (138)
$$

and by interpolating the $f_{S}^{[\epsilon]}(p)$ value for all $p \neq p_{r;i}$. Comparing (87) and (138), the difference is the conditioning of a new set of random vectors $Z_{\text{self}} \triangleq \{Z_{i \rightarrow i} : \forall i \in [M]\}$. Namely, we consider only the conditional mutual information given what has already been received by $d_i$ through the 2-hop jump from $s_i$ to $d_i$.

When $S = \emptyset$, by definition we have

$$
\frac{f_{S}^{[\epsilon]}(p_{r;i})}{I(U;Z_{r\rightarrow i}^{[\epsilon]}(S_{r\rightarrow j}^{[\epsilon]} \mid j \in \mathcal{S}_{\text{self}})} \leq \beta_r \cdot p_{r;i},
$$

which will be used to prove (131).

Using the new definition of $f_{S}^{[\epsilon]}(p)$, we can derive the following equality by similar steps as in (89) to (92):

$$
(n \log(q)) \cdot f_{S}^{[\epsilon]}(p_{r;i}) = (n \log(q)) \cdot f_{T}^{[\epsilon]}(p_{r;i}) + I(X_k;Z_{S\rightarrow k} \mid \{X_j, Z_{S\rightarrow j} : \forall j \in S\} \mid Z_{\text{self}}), (139)
$$

Define $T = S \cup \{k\}$. Using the following equality,

$$
I(X_k;\{Z_{k\rightarrow j} : \forall j \in T\}) = I(X_k;Z_{k\rightarrow k}) + I(X_k;\{Z_{k\rightarrow j} : \forall j \in S\} \mid Z_{k\rightarrow k})
$$

we can carry out similar steps as in (93) to (103) and prove that

$$
I(X_k;Z_{S\rightarrow k} \mid \{X_j, Z_{S\rightarrow j} : \forall j \in S\} \mid Z_{\text{self}}) \geq (n \log(q))(R_k - \epsilon) - H(2\epsilon) - I(X_k;\{Z_{k\rightarrow j} : \forall j \in T\})^+. (140)
$$

Since

$$
I(X_k;\{Z_{k\rightarrow j} : \forall j \in T\}) \leq n \beta_k \log(q)p_{s_k;i,T}, (141)
$$

combining (139), (140), and (141), we thus have that $\forall i, k \in [M], \forall S, T \in 2^{[M]}$ satisfying $k \notin S$ and $T = S \cup \{k\}$,

$$
\begin{align*}
&f_{S}^{[\epsilon]}(p_{r;i}) \geq f_{T}^{[\epsilon]}(p_{r;i}) - I(X_k;\{Z_{k\rightarrow j} : \forall j \in T\})^+ + \beta_k \cdot p_{s_k;i,T}.
\end{align*}
$$

which will be used to prove (130). By repeating the pointwise functional limit construction in Appendix B, the proof of Proposition 7 is complete.

**Proof of Proposition 8:** This proof follows closely that of Proposition 4 in Sections VI-A and VI-B. To that end, we first convert the inner bound of Proposition 8 to an equivalent LP problem by the alternative formulation of a ZPPLCIF in Lemma 1.

**Proposition 9:** Following the same settings and definitions of Proposition 8, a rate vector $(R_1, \ldots, R_M)$ can be achieved by a linear network code if there exist $(M + 1)$ time allocation variables $(\beta_1, \ldots, \beta_M, \beta_r)$ with the total sum being $(M + 1)$, and $(M + 1)$ non-negative variables $(x_{\pi,S^*_\pi h} : \forall \pi, \forall j \in [M], \forall h \in [M])$ and $M(2^{M+1} - M - 2)$ non-negative variables:

$$
\begin{align*}
\left\{ y_{k,T} : (\pi, S^*_\pi) \right\} &\ni: \forall k \in [M], \forall T \in 2^{[M]}, \forall \pi, \forall j \in [M + 1], \\
\text{satisfying } k &\notin S^*_\pi \text{ and } T \subseteq S^*_\pi,
\end{align*}
$$

and $M2^{M-1}$ non-negative variables:

$$
\begin{align*}
\left\{ z_{k,T} \right\} &\ni: \forall T \in 2^{[M]}, \forall k \in [M], \\
\text{satisfying the joint rate } R_t \text{ and the } x, y, z, \beta \text{ variables satisfy } (132), (134), (135), (136), (63), \text{ and the following inequality:}
\end{align*}
$$

$$
\sum_{\forall \pi, \forall h = 1}^{M} x_{\pi,0,h} \cdot p_{r;i,M} \leq \beta_r \cdot p_{r;i} \Leftrightarrow \sum_{\forall \pi}^{M} \sum_{\forall h = 1}^{M} x_{\pi,0,h} < \beta_r.
$$

For any rate vector $(R_1, \ldots, R_M)$ satisfies the inner bound of Proposition 9, perform the following four steps.

**Step 1:** For each time slot $t \in [n]$, each source $s_i, i \in [M]$, sends out the coded symbol $W_i(t)$ by RLNC, i.e.,

$$
W_i(t) = \sum_{k = 1}^{nR_i} c_{i,k}(t)X_{i,k},
$$

for which we choose the mixing coefficients $c_{i,k}(t)$ independently and uniformly randomly from GF($q$), $c_{i,k}(t) \Delta (c_{i,1}(t), \ldots, c_{i,nR_i}(t))$ denotes the corresponding $(nR_i)$-dimensional coding vector used in time $t$.

**Step 2:** The relay $r$ derives the following linear subspaces $V_{i;r}$ and $V_{i;j}$ for all $i, j \in [M], i \neq j$:

$$
\begin{align*}
V_{i;r} &\Delta \text{span}(\{c_{i,t} : \forall t \text{ s.t. } Z_{i\rightarrow t}(t) \neq \ast\}) \\
V_{i;j} &\Delta \text{span}(\{c_{i,t} : \forall t \text{ s.t. } Z_{i\rightarrow t}(t) = \ast\})
\end{align*}
$$

Namely, $V_{i;j}$ is the subspace of user $i$ that is decodable by $r$, not received by $d_i$ directly, but is overhead by $d_j$. Based on $\{V_{i;j} : i \neq j\}$, the relay $r$ further derives multiple subspaces $V_{i=1:T_2,}$ for all $i \in [M], T_1 \subseteq [M]\{i\}$, and $T_2 = [M]\{T_1 \cup \{i\}\}$ in a similar way as in Step 2 of Section VI-A, such that the following three properties are satisfied:

(i) $\forall j \in T_1,$ $V_{i=1:T_1,} \subseteq V_{i;j},$

(ii) $\text{span}(\{V_{i;i}, V_{i=1:T_1,} : \forall T \subseteq [M]\{i\}, T_2 = [M]\{T_1 \cup \{i\}\}\}) = \Omega_i,$

(iii) $\frac{n}{\text{Rank}(V_{i=1:T_2,})} = z_{i,T_2}.$

Namely, the message space $\Omega_i$ of user-$i$ is partitioned into multiple subspaces $V_{i=1:T_2,}$ such that each subspace $V_{i=1:T_2,}$ is known to the relay $r$ and all the users $j \in T_1,$ and the normalized rank of $V_{i=1:T_1,}$ equals to the feasible solution $z_{i,T_2}$ satisfying (135) and (136) of Proposition 8.

With the above construction of the subspaces, let $V_{k}^{[i=1:T_1]}$ denote the basis vectors of $V_{i=1:T_1,}$. We can then perform the regrouping operation and the code-alignment-based encoding method in the same way as in Steps 3 and 4.
of Sections VI-A and VI-B. It is then guaranteed that all the information subspaces $V_{i-j-T_1-T_2}$ at the relay $r$ can be conveyed to destination $d_i$. Property (ii) of $V_{i-j-T_1-T_2}$ then guarantees the decodability at $d_i$. The proof is thus complete.

References


