The Capacity Region of 2-Receiver Multiple-Input Broadcast Packet Erasure Channels with Channel Output Feedback

Chih-Chun Wang, *Member, IEEE*, Jaemin Han, *Student Member, IEEE*

***Abstract*—This work studies the capacity of the 2-receiver multiple-input broadcast packet erasure channels (PECs) with channel output feedback, which is in contrast with the single-input setting of the existing works. Motivated by the immense success of linear network coding (LNC) in theory and in practice, this work first focuses on LNC schemes and characterizes the LNC feedback capacity region of 2-receiver multiple-input broadcast PECs. A new linear-space-based approach is proposed, which unifies the problems of finding a capacity outer bound and devising the achievability scheme into a single linear programming (LP) problem. Specifically, *an LP solver is used to exhaustively search for the LNC scheme(s) with the best possible throughput, the result of which is thus guaranteed to attain the LNC feedback capacity.* It is then proven by pure algebraic arguments that the LNC capacity region matches a simple capacity region outer bound, which proves that the derived LNC capacity region is indeed the true capacity. A byproduct of the above results is a complete LNC capacity region characterization for 2-receiver partially Markovian and partially controllable broadcast PECs.**

***Index Terms*—Broadcast capacity; broadcast channels; channel output feedback; linear network coding; multiple-input/multiple-output (MIMO) channels; packet erasure channels.**

I. Introduction

It is well known that channel output feedback can potentially enlarge the capacity region of broadcast channels [2], [13]. Recently the feedback capacity region of K-receiver broadcast *packet erasure channels* (PECs) has been fully characterized for $K \leq 3$ [6], [15]. For the case of $K > 3$, the full feedback broadcast capacity region has been characterized for the setting of perfectly symmetric channels and for the setting of *one-sidedly-fair capacity* with spatially independent erasure events, respectively [6], [15]. Numerically tight capacity inner and outer bounds have also been proposed in [15], which can be computed efficiently by linear programming (LP) solvers for arbitrary K and arbitrary channel characteristics. Specifically, a 1-to-K PEC takes an input symbol (also known as a packet) W from some (large) finite field $\mathsf{GF}(q)$ and each destination d_k (out of K destinations) receives either the input packet $Z_k = W$ or an erasure $Z_k = *$, depending on whether the packet W has successfully arrived at d_k. The single-input broadcast PEC model in [6], [15] captures closely the network coding capacity for the downlink transmission from a single access point to multiple clients with one antenna and simple modulation schemes [9].

On the other hand, such a single-input broadcast PEC model does not take into account several commonly used modern communication schemes. For example, 2 antennas may be used at both the source s and the destinations d_k, which corresponds to a multiple-input broadcast PEC [4] that takes $(W^{[1]}, W^{[2]}) \in (\mathsf{GF}(q))^2$ as input, and each d_k may receive one of the four possible outcomes $(W^{[1]}, *)$, $(*, W^{[2]})$, $(W^{[1]}, W^{[2]})$, and $(*, *)$ depending on whether the packet $W^{[m]}$ sent by antenna m, $m = 1, 2$, is decodable or not. Even when only a single antenna is used, source s may use Orthogonal Frequency Division Multiple Access (OFDMA), which, in each time slot, can send out multiple streams of packets over different sub-carriers. Each sub-carrier may experience different random erasure events. Each d_k constantly scans *all* subcarriers and records any overheard packets. The multiple sub-carriers in OFDMA can again be modeled as a multiple-input broadcast PEC.

This works considers the multiple-input broadcast PEC with 2 receivers and *channel output feedback*. That is, in each time slot t, the source s sends M symbols $(W^{[1]}, W^{[2]}, \cdots, W^{[M]})$. Depending on the channel realization, each destination d_i, $i = 1, 2$, may hear a random set of $W^{[\cdot]}$ symbols. At the end of time t, each d_i reports back to s which subset of symbols it has received in the current time slot through the use of ACK or NACK. The setting of interest generalizes the existing work [4] by further considering the throughput benefits of channel output feedback. Motivated by the immense success of linear network coding (LNC) [11], this work first focuses on LNC schemes and characterizes the full LNC feedback capacity region (R_1^*, R_2^*). A new framework is proposed that unifies the problems of finding an LNC capacity outer bound and designing the corresponding bound-achieving solution into a single linear-programming (LP) problem. Namely, we first drastically simplify the LNC design choices in a lossless way by leveraging upon the underlying linear space structures. Then we use an LP solver to search exhaustively over all possible LNC design choices and find the LNC solution that achieves the highest throughput. The exhaustiveness guarantees that the resulting LNC scheme is throughput optimal (among all LNC solutions) and thus achieves the LNC capacity.

This work was supported in parts by NSF Grants CCF 0845968 and CNS 0905331.

Part of this paper was presented in the IEEE International Symposium on Information Theory, 2012.

C.-C. Wang and J. Han are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47906 USA; e-mail: {chihw,han83}@purdue.edu

Manuscript received in May 2012; revised in May 2014.

Remark 1: Such a constructive optimality proof has been widely used in the networking community but not in the information theory community. For example, in the networking society, *the optimal multi-path routing throughput is found by simply searching over all possible routing decisions that obey the flow-conservation law,* which is in contrast with the traditional information-theoretic approach that first finds a cut and an achievability scheme and later proves that the achievable throughput meet the cut value. This exhaustive-search-based approach was previously not possible since there are too many LNC design choices. With a new framework that leverages upon the underlying linear space structure of LNC, we can greatly reduce the number of design choices and are thus able to design provably optimal LNC schemes without the need of finding any cut condition!

Remark 2: This work considers the so-called *inter-flow NC problem* with two coexisting traffic flows. When there is only one flow in the network, termed the intra-flow NC problem, [3] proves that channel output feedback does not increase the end-to-end capacity of any arbitrary erasure networks, and random linear network coding [8] achieves the capacity.

After characterizing the LNC capacity region of 2-receiver multi-input broadcast PECs, we will prove by some pure algebraic arguments that the LNC capacity region matches a simple capacity region outer bound. The results thus prove that the newly derived LNC capacity region is indeed the capacity.

The rest of this paper is organized as follows. Section II introduces the problem formulation. Section III discusses several existing results. The main results of this work are the full LNC feedback capacity characterization and the equivalence between the LNC capacity region and the true capacity, which are presented in Section IV. Section V contains the proof for the case of $M = 1$ input, which illustrates several key concepts of this work. Section VI generalizes the proof for the case of arbitrary M inputs. Section VII discusses in details a concrete multi-input broadcast PEC example and some further implications of the capacity results, including the LNC feedback capacity of the partially Markovian and partially controllable broadcast PECs. Section VIII concludes this work.

II. Problem Formulation

Given a finite field $\mathsf{GF}(q)$, the M-input 2-receiver broadcast PEC is defined as follows. For any time slot, source s sends M symbols $\mathbf{W} \triangleq (W^{[1]}, W^{[2]}, \cdots, W^{[M]}) \in (\mathsf{GF}(q))^M$ and each d_i receives a random subset $\mathsf{rx}_i \subseteq \{1, 2, \cdots, M\}$ of the $W^{[\cdot]}$ symbols for $i = 1, 2$. Each of the M input symbols $W^{[m]}$ takes values in $\mathsf{GF}(q)$. The randomness of an M-input 2-receiver broadcast PEC can be described by the joint reception probability $p_{\mathsf{rx}_1,\mathsf{rx}_2}$ such that $\sum_{\forall \mathsf{rx}_1,\mathsf{rx}_2} p_{\mathsf{rx}_1,\mathsf{rx}_2} = 1$. For example, when $M = 3$, $p_{\{1,2\},\{2,3\}}$ is the probability that d_1 receives $\mathbf{Z}_1 \triangleq (W^{[1]}, W^{[2]}, *)$ and d_2 receives $\mathbf{Z}_2 \triangleq (*, W^{[2]}, W^{[3]})$. We consider only stationary and memoryless channels, i.e., $\{p_{\mathsf{rx}_1,\mathsf{rx}_2}\}$ remains constant (over time) and the reception events for any distinct time slots $t_1, t_2, \cdots$ are independent. For any given joint probability distribution $p_{\mathsf{rx}_1,\mathsf{rx}_2}$, two independence conditions can be defined as follows.

Definition 1: An M-input 2-receiver broadcast PEC is *cross-input independent* if for any $m \neq \tilde{m}$, the two 2-dimensional random vectors $(1_{\{m\in\mathsf{rx}_1\}}, 1_{\{m\in\mathsf{rx}_2\}})$ and $(1_{\{\tilde{m}\in\mathsf{rx}_1\}}, 1_{\{\tilde{m}\in\mathsf{rx}_2\}})$ are independent where $1_{\{\cdot\}}$ is the indicator function.

Definition 2: An M-input 2-receiver broadcast PEC is *cross-receiver* independent[1] if the two random sets rx_1 and rx_2 are independently distributed.

Throughout this work we focus mostly only on M-input 2-receiver broadcast PECs that are cross-input independent (but may or may not be cross-receiver independent). For comparison, the $(M, 2)$ *erasure broadcast channel* defined in [4] is equivalent to the M-input 2-receiver model that is both cross-input and cross-receiver independent.

We use $p^{[m]}_{a_1a_2}$ to denote the (marginal) reception probabilities for the m-th symbol $W^{[m]}$ where each bit a_i indicates whether d_i receives $W^{[m]}$ or not for $i = 1, 2$. For example, by definition

$$p^{[m]}_{10} = \sum_{\forall \mathsf{rx}_1,\mathsf{rx}_2 \text{ s.t. } m\in\mathsf{rx}_1, m\notin\mathsf{rx}_2} p_{\mathsf{rx}_1,\mathsf{rx}_2}$$

is the probability that the m-th input symbol $W^{[m]}$ is successfully received by d_1 but not by d_2.

Consider the following communication problem. For any rate vector (R_1, R_2), within n time slots source s would like to send two independent packet streams $\mathbf{X}_i \triangleq (X_{i,1}, X_{i,2}, \cdots, X_{i,nR_i}) \in (\mathsf{GF}(q))^{nR_i}$ to destination d_i for $i = 1, 2$, respectively. At the end of each time slot, each d_i reports back to s which subset of symbols it has received (the rx_i value) through the use of ACK or NACK. This channel output feedback setting was not considered in the existing work [4].

If we use the input argument "(t)", $t = 1, 2, \cdots, n$, to distinguish the n channel usages, a network code can be described by n encoding functions: for all $t = 1, \cdots, n$,

$$\mathbf{W}(t) = f_t(\mathbf{X}_1, \mathbf{X}_2, [\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}) \tag{1}$$

and two decoding functions: for all $i = 1, 2$,

$$\hat{\mathbf{X}}_i = g_i\left([\mathbf{Z}_i]_1^n, \{f_t(\cdot, \cdot, [\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}) : t = 1, \cdots, n\}\right), \tag{2}$$

where $[\mathbf{Z}_i]_1^n$ denotes what d_i has received from time 1 to n, and $[\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}$ denotes the channel output information from time 1 to $(t - 1)$. In (2) we assume that each d_i knows[2] how the coded symbols are generated (the functions $f_t(\cdot, \cdot, [\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1})$) but does not know the actual information symbols $\mathbf{X}_1$ and $\mathbf{X}_2$ used to generate $\mathbf{W}(t)$.

A network code is linear if the encoders f_t are linear with respect to $\mathbf{X}_1$ and $\mathbf{X}_2$, i.e., when (1) can be written as

$$\mathbf{W}(t) = \overline{\mathbf{X}} \cdot \mathbf{C}_t,$$

[1] Cross-receiver independence is also termed "spatial independence" in [15].

[2] In general, d_i always knows $f_t(\cdot, \cdot, \cdot)$, the overall communication scheme that is agreed upon before transmission. But d_i does not know $f_t(\cdot, \cdot, [\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1})$ since it depends on the (random) channel realization of the other destination d_j. As a result, a more realistic definition of the decoder should be

$$\hat{\mathbf{X}}_i = g_i\left([\mathbf{Z}_i]_1^n\right). \tag{3}$$

where $\mathbf{C}_t$ is an $(nR_1 + nR_2) \times M$ matrix in $\mathsf{GF}(q)$ and $\overline{\mathbf{X}} \triangleq (\mathbf{X}_1, \mathbf{X}_2)$ is an $n(R_1 + R_2)$-dimensional row vector consisting of all information symbols. The choice of $\mathbf{C}_t$ depends on $[\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}$ but not on $\overline{\mathbf{X}}$. In practice [1], the *coding coefficients* $\mathbf{C}_t$ are often embedded in the header of the packets so that upon the successful reception of a packet $W^{[m]}(t)$, the corresponding coding coefficients used to generate $W^{[m]}(t)$ are known to the receiver. Hence a decoder of the form of (2) can be used.

Definition 3: Given a finite field $\mathsf{GF}(q)$, a rate vector (R_1, R_2) is achievable if for any $\epsilon > 0$ there exists a network code of length n such that $\mathsf{Prob}(\hat{\mathbf{X}}_i \neq \mathbf{X}_i) < \epsilon$ for all $i = 1, 2$. The capacity region is defined as the closure of all (R_1, R_2) that are achievable.

Definition 4: A rate vector (R_1, R_2) is LNC-achievable if it can be achieved by a linear network code. The LNC capacity region is the closure of all LNC-achievable (R_1, R_2).

Note that the unit of the above capacity definition is *packets per time slot*. It can be converted to the traditional unit *bits per time slot* by multiplying a factor of $\log_2(q)$ since each packet can carry $\log_2(q)$ bits of information. Throughout this paper, we use exclusively *packets per time slot* as the unit.

III. Discussion Of The Existing Results

A. A Pair of Simple Inner and Outer Bounds

A simple inner bound on the above capacity problem can be derived by performing LNC over the M individual single-input broadcast PECs (one for each of the M inputs) separately. Therefore, we have the following lemma:

Lemma 1: For any fixed $\mathsf{GF}(q)$, a rate vector (R_1, R_2) is LNC-achievable if there exist $2M$ non-negative variables $R_1^{[m]}$ and $R_2^{[m]}$, for all $m = 1, \cdots, M$, such that the following conditions are satisfied.

$$\forall i = 1, 2, \quad \sum_{m=1}^{M} R_i^{[m]} = R_i \tag{4}$$

$$\forall m = 1, \cdots, M, \quad \begin{cases} \frac{R_1^{[m]}}{p_{10}^{[m]}+p_{11}^{[m]}} + \frac{R_2^{[m]}}{p_{10}^{[m]}+p_{01}^{[m]}+p_{11}^{[m]}} < 1 \\ \frac{R_1^{[m]}}{p_{10}^{[m]}+p_{01}^{[m]}+p_{11}^{[m]}} + \frac{R_2^{[m]}}{p_{01}^{[m]}+p_{11}^{[m]}} < 1 \end{cases}. \tag{5}$$

Proof: (4) follows from summing up the per-input LNC rates and (5) follows from the feedback capacity region results for 1-input 2-receiver broadcast PECs [7]. ■

Following similar steps as in [7] we can derive the following capacity outer bound.

Lemma 2: For any fixed $\mathsf{GF}(q)$, consider any M-input 2-receiver broadcast PEC that is cross-input independent. A rate vector (R_1, R_2) is achievable only if there exist $4M$ non-negative variables $R_i^{[m,k]}$ for all $i, k \in \{1, 2\}$ and $m \in \{1, \cdots, M\}$ such that the following conditions are satisfied.

$$\forall i, k \in \{1, 2\}, \quad \sum_{m=1}^{M} R_i^{[m,k]} = R_i \tag{6}$$

and for all $m \in \{1, \cdots, M\}$,

$$\frac{R_1^{[m,1]}}{p_{10}^{[m]} + p_{11}^{[m]}} + \frac{R_2^{[m,1]}}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}} \leq 1 \tag{7}$$

$$\frac{R_1^{[m,2]}}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}} + \frac{R_2^{[m,2]}}{p_{01}^{[m]} + p_{11}^{[m]}} \leq 1. \tag{8}$$

Proof: The proof simply combines the capacity outer bound construction in [13] and the zero-feedback capacity results in [4].

Specifically, we first construct a new M-input 2-receiver broadcast PEC from the original broadcast PEC in the following way. Namely, we allow d_1 to observe both $\mathbf{Z}_1$ and $\mathbf{Z}_2$, the latter of which is what d_2 receives in each time slot. Since d_1 can observe additional information $\mathbf{Z}_2$, the feedback capacity of the new broadcast PEC is an outer bound of the feedback capacity of the original PEC.

Further, for the new broadcast PEC, d_2 is a physically degraded receiver when compared to d_1. By [5], channel output feedback does not improve the performance of a physically degraded broadcast channel. As a result, the feedback capacity of the new broadcast PEC equals to the feedback-free capacity of the new broadcast PEC.

By the capacity expression[3] in [4], the feedback-free capacity of the new broadcast PEC is described by (6) with $k = 2$ and by (8), which, by our aforementioned reasoning, serves as an outer bound for the feedback capacity of the original broadcast PEC. If we swap the roles of d_1 and d_2 and repeat the same argument, we can show that (6) with $k = 1$ and (7) must also be an outer bound for the feedback capacity of the original broadcast PEC. The proof of this lemma is completed by taking the intersection of the two outer bounds. ■

B. An Improved Inner Bound

The capacity inner bound in Lemma 1 turns out be be suboptimal due to the fact that it performs network coding separately for each of the M sub-channels. To tighten the inner bound, one can follow the ideas of [7], [10] and devise a simple LNC scheme that encodes jointly over all M sub-channels and achieves the following improved inner bound.

Lemma 3: For any fixed $\mathsf{GF}(q)$, a rate vector (R_1, R_2) is LNC-achievable if there exist $2M$ non-negative variables $R_1^{[m]}$ and $R_2^{[m]}$, for all $m = 1, \cdots, M$, such that the following

However, the capacity difference between the stronger (2) and the weaker (3) decoders is negligible when a sufficient large $\mathsf{GF}(q)$ is used. The reason is that in addition to the allotted n time slots, s can simply use a few extra time slots to "broadcast" the binary channel status $[\mathsf{rx}_1, \mathsf{rx}_2]_1^n$ to both destinations so that a more powerful decoder in (2) can be used. The overhead of using extra time slots to convey the *binary* receptions status $[\mathsf{rx}_1, \mathsf{rx}_2]_1^n$ to $\{d_1, d_2\}$ diminishes to zero when a sufficiently large $\mathsf{GF}(q)$ is used. As a result, the capacity gap is negligible for large q. In this work we focus exclusively on decoders of the form of (2). Also see [1] and our discussion of the practical LNC implementation.

[3] The newly constructed broadcast PEC is cross-input independent but cross-receiver dependent. Although the results in [4] were stated for the setting of cross-input and cross-receiver independence, we can use Sato's argument [14] to relax the condition of cross-receiver independence.

conditions are satisfied:

$$\forall i = 1, 2, \quad \sum_{m=1}^{M} R_i^{[m]} = R_i \tag{9}$$

$$\forall m = 1, \cdots, M, \quad \frac{R_1^{[m]} + R_2^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} < 1 \tag{10}$$

$$R_1 + \sum_{m=1}^{M} \left(\frac{R_2^{[m]} \left(p_{10}^{[m]} + p_{11}^{[m]} \right)}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}} \right) < \sum_{m=1}^{M} \left(p_{10}^{[m]} + p_{11}^{[m]} \right) \tag{11}$$

$$R_2 + \sum_{m=1}^{M} \left(\frac{R_1^{[m]} \left(p_{01}^{[m]} + p_{11}^{[m]} \right)}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}} \right) < \sum_{m=1}^{M} \left(p_{01}^{[m]} + p_{11}^{[m]} \right). \tag{12}$$

The main idea behind Lemma 3 is to first divide the information packets among the M inputs, see (9), and send the packets uncodedly until each one is received by at least one of d_1 and d_2, see (10). Then collect all the newly created *network coding opportunities* [7], [10], [12], redistribute the coding opportunities among all M inputs, and use the remaining time slots (of each of the M sub-channels) to send linear combinations of the packets that have not arrived at their intended destinations. The proof of Lemma 3 and detailed discussion are relegated to Appendix A. One can also verify from the proof that the inner bound in Lemma 3 always contains the inner bound in Lemma 1.

C. An Illustrative Example

Both the inner bounds in Lemmas 1 and 3 are suboptimal in the following sense. For some special choices of the parameters $p_{a_1a_2}^{[m]}$, the two inner bounds in Lemmas 1 and 3 coincide with the outer bound in Lemma 2. However, for some $p_{a_1a_2}^{[m]}$ values even the improved inner bound in Lemma 3 is strictly contained in the outer bound in Lemma 2.

Take the following 2-input, 2-receiver MIMO broadcast PEC for example. Suppose the joint reception probability $\{p_{\mathsf{rx}_1,\mathsf{rx}_2}\}$ are

$$\begin{aligned} &p_{00}^{[1]} = 0, \quad p_{01}^{[1]} = 0.125, \quad p_{10}^{[1]} = 0, \quad p_{11}^{[1]} = 0.875; \\ &p_{00}^{[2]} = 0.04, \quad p_{01}^{[2]} = 0.16, \quad p_{10}^{[2]} = 0.16, \quad p_{11}^{[2]} = 0.64, \end{aligned} \tag{13}$$

and the channel is cross-input independent. This is basically a setting of cross-input and cross-receiver independence with the m-th-input-to-d_i sub-channel having success probability 0.875, 1, 0.8, and 0.8 for $(m, i) = (1, 1)$, $(1, 2)$, $(2, 1)$, and $(2, 2)$, respectively. We plot in Fig. 1 the inner and outer bounds in Lemmas 1 and 2 for this example. As can be seen, there is a non-zero gap. Further, one can verify that for these particular $p_{a_1a_2}^{[m]}$ values, the rate vector $(R_1, R_2) = (0.875, 0.96)$, a point within the outer bound in Lemma 2, is strictly outside (the closure of) the improved inner bound in Lemma 3.

To close the gap between the inner and outer bounds, we need to either design a more powerful achievability scheme or further sharpen the outer bounding arguments. Both are non-trivial tasks since (i) for arbitrary M, there is an exponentially large number of ways of designing a coding scheme and we can no longer rely on the ad-hoc design methodology as used in Lemma 3; (ii) We do not have many analytical tools for feedback capacity analysis other than the existing approach discussed in Lemma 2.

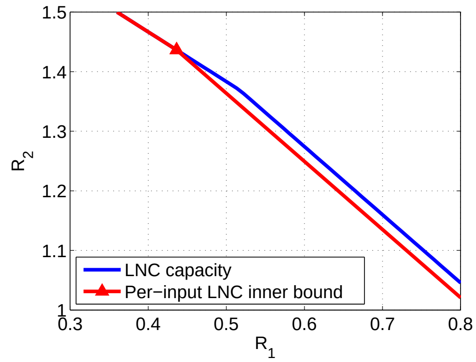

Fig. 1. The LNC capacity computed by Proposition 1 versus the per-input LNC inner bound in Lemma 1. The LNC capacity also coincides with the capacity outer bound in Lemma 2.

In this work, we circumvent the above difficulties by first focusing on LNC. We characterize the LNC capacity region for *cross-input independent M-input 2-receiver broadcast PECs*. As will be seen shortly after, this work derives a new systematic, constructive LNC capacity proof that does not rely on the ad-hoc design nor on the outer bounding arguments in the existing results.

After characterizing the LNC capacity, we use some pure algebraic arguments to prove that the LNC capacity region is identical to the outer bound in Lemma 2. Therefore, the LNC capacity region is the true capacity of the M-input 2-receiver broadcast PECs.

IV. Main Results

Consider a finite index set FTs of 18 elements:

$$\mathsf{FTs} \triangleq \{0, 1, 2, 3, 7, 9, 11, 15, 18, 19, 23, 27, 31, 47, 63, 87, 95, 127\}. \tag{14}$$

It will be clear in Section V why we consider such an index set. Let $\mathbf{b} = b_1b_2b_3 \cdots b_7 \in \{0, 1\}^7$ denote a 7-bit string. We can also view $\mathbf{b}$ as a base-2 expression with the leftmost bit being the most significant bit (MSB). We append zeros in the prefix to make the length always 7. For example, the statement "$\mathbf{b} = 9$" is equivalent to "$\mathbf{b} = 0001001$" and the statement "$\mathbf{b} \in \mathsf{FTs}$ and $b_7 = 0$" is equivalent to "$\mathbf{b} \in \{0, 2, 18\}$."

The main results of this work can now be stated as follows.

Proposition 1: For any fixed $\mathsf{GF}(q)$, consider any cross-input independent M-input 2-receiver broadcast PEC. A rate vector (R_1, R_2) is in the LNC capacity region if and only if there exist $18M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $\mathbf{b} \in \mathsf{FTs}$ and $m = 1, \cdots, M$, and 7 non-negative variables y_1 to y_7 such that jointly they satisfy the following 4 groups of linear conditions:

• Group 1, termed the *time-sharing conditions*, has M equalities:

$$\forall m = 1, \cdots, M, \quad \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs}} x_{\mathbf{b}}^{[m]} \right) \leq 1. \tag{15}$$

• Group 2, termed the *rank-conversion conditions*, has 7 equalities:

$$y_1 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_1=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]} \right) \tag{16}$$

$$y_2 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_2=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{01}^{[m]} + p_{11}^{[m]} \right) \tag{17}$$

$$y_3 = R_1 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_3=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]} \right) \tag{18}$$

$$y_4 = R_2 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_4=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{01}^{[m]} + p_{11}^{[m]} \right) \tag{19}$$

$$y_5 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_5=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right) \tag{20}$$

$$y_6 = R_1 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_6=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right) \tag{21}$$

$$y_7 = R_2 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_7=0} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right) \tag{22}$$

• Group 3, termed the *rank-comparison conditions*, has 7 inequalities:

$$y_3 \leq y_6, \quad y_4 \leq y_7 \tag{23}$$

$$y_6 \leq (R_1 + R_2), \quad y_7 \leq (R_1 + R_2) \tag{24}$$

$$y_5 + y_3 - y_6 \geq y_1 \tag{25}$$

$$y_5 + y_4 - y_7 \geq y_2 \tag{26}$$

$$y_6 + y_7 - (R_1 + R_2) \geq y_5. \tag{27}$$

• Group 4, termed the *decodability conditions*, has 2 equalities:

$$y_3 = y_1 \text{ and } y_4 = y_2. \tag{28}$$

Obviously, the LNC capacity region in Proposition 1 is an inner bound of the true capacity region. A stronger result can then be proven as follows.

Proposition 2: For any fixed $\mathsf{GF}(q)$, the LNC capacity region in Proposition 1 matches the outer bound in Lemma 2. Therefore, the Shannon capacity region can be described either by the LP problem in Lemma 2 that contains $4M$ variables and $4 + 2M$ (in)equalities, or by the LP problem in Proposition 1 that contains $18M + 7$ variables and $M + 16$ (in)equalities.

The proof of Proposition 1 is relegated to Sections V and VI. The proof of Proposition 2 is by some pure algebraic arguments, which are provided in Appendix B. A byproduct of the proof of Proposition 2 is the following lemma that sheds further insights to the LP problem in Proposition 1.

Lemma 4: For any given (R_1, R_2) value, if the corresponding LP problem in Proposition 1 is feasible, then there exist $18M$ $x_{\mathbf{b}}^{[m]}$ and 7 y_i variables satisfying simultaneously (15) to (28) and the following additional property: $x_{\mathbf{b}}^{[m]} \neq 0$ only if $\mathbf{b} \in \{0, 9, 18, 27, 31, 63, 95\}$.

The proof of Lemma 4 is provided in Appendix B. Algebraically, the above lemma implies that we can, without loss of generality, reduce the number of $x_{\mathbf{b}}^{[m]}$ variables from $18M$ to $7M$ when solving the LP problem in Proposition 1. The physical meaning of Lemma 4 will be discussed in Section VI-D.

Proposition 1 characterizes the capacity of cross-input independent broadcast PECs, and it can also be used as an inner bound for cross-input dependent broadcast PECs through the following lemma.

Lemma 5: For any fixed $\mathsf{GF}(q)$, consider two M-input 2-receiver broadcast PECs CH_1 and CH_2 that have the same per-input marginal reception probabilities $\{p_{a_1 a_2}^{[m]} : \forall m, \forall a_1, a_2 \in \{0, 1\}\}$ while CH_1 is cross-input independent but CH_2 is not. Any (R_1, R_2) that is achievable (resp. LNC achievable) for CH_1 is also achievable (resp. LNC achievable) for CH_2.

This lemma can be proven by interleaving several CH_1-based constituent codes over the time axis and the M-symbols, and then apply the interleaved super code to CH_2. A detailed proof is provided in Appendix C.

V. Proof of Proposition 1 for $M = 1$

We first provide the proof of Proposition 1 for the case of $M = 1$, which can be viewed as an alternative proof for the existing $M = 1$ achievability results in [7]. The concepts and terminology used in the proof of $M = 1$ case will be useful when proving the general case of $M \geq 2$ in Section VI.

A. Basic Definitions

Consider the case of $M = 1$. That is, for each time slot, s sends one symbol $W(t) = \overline{\mathbf{X}} \cdot \mathbf{c}_t^{\mathrm{T}}$ where $\mathbf{c}_t$ is an $n(R_1 + R_2)$-dimensional (row) coding vector consisting of the coding coefficients and $\mathbf{c}_t^{\mathrm{T}}$ is the transpose of $\mathbf{c}_t$. When $M = 1$ the receiving set rx_i of d_i can take values as either $\mathsf{rx}_i = \emptyset$ or $\mathsf{rx}_i = \{1\}$ where the former indicates that d_i received an erasure while the latter indicates that d_i received the first (and also the only) transmitted symbol $W(t)$.

For each d_i, we define the *knowledge space* $S_i(t)$ in the end of time t by

$$S_i(t) \triangleq \mathsf{span}\{\mathbf{c}_\tau : \forall \tau \leq t \text{ s.t. } 1 \in \mathsf{rx}_i \text{ at time } \tau\}. \tag{29}$$

That is, $S_i(t)$ is the linear span of the vectors of those packets that have successfully arrived at d_i. By convention, we define the linear span of an empty vector set to be a linear sub-space containing only the all-zero vector $\mathbf{0}$. Namely, $\mathsf{span}(\emptyset) = \{\mathbf{0}\}$. For example, $S_i(0) = \{\mathbf{0}\}$ by this definition.

One can easily see that in the end of time t, d_i is able to compute the value of $\overline{\mathbf{X}} \cdot \mathbf{v}^{\mathrm{T}}$ for all $\mathbf{v} \in S_i(t)$ by linearly combining $Z_i(\tau)$ for all $\tau \leq t$.

For $j = 1$ to $n(R_1 + R_2)$, let δ_j denote an $n(R_1 + R_2)$-dimensional elementary delta (row) vector with its j-th coordinate being one and all the other coordinates being zero. Define $\Omega \triangleq \mathsf{span}\{\delta_j : j = 1, \cdots, n(R_1 + R_2)\}$ as the *overall message space* and define $\Omega_1 \triangleq \mathsf{span}\{\delta_j : j = 1, \cdots, nR_1\}$ and $\Omega_2 \triangleq \mathsf{span}\{\delta_j : j = (nR_1 + 1), \cdots, n(R_1 + R_2)\}$ as the *individual message spaces* for d_1 and d_2, respectively. Both $S_i(t)$ and Ω_i are linear subspaces of Ω for $i = 1, 2$. Sometimes we also call Ω the *overall coding vector space* since any inter-flow coding vector $\mathbf{c}_t$ must be in Ω.

For any two linear subspaces $A, B \subseteq \Omega$, define $A \oplus B \triangleq \mathsf{span}\{\mathbf{v} : \forall \mathbf{v} \in A \cup B\}$ as the *linear sum space* of A and B. From the discussion in the beginning of this subsection, d_i can decode the desired $X_{i,1}, \cdots, X_{i,nR_i}$ symbols if and only if in the end of time n we have $\Omega_i \subseteq S_i(n)$, or equivalently

$$(S_i(n) \oplus \Omega_i) = S_i(n). \tag{30}$$

B. Break Down The Design Choices

We now demonstrate how to use the knowledge spaces to break down the design choices of LNC.

In the beginning of time t (or equivalently in the end of time $t-1$), there are $q^{n(R_1+R_2)}$ different ways of designing the coding vector $\mathbf{c}_t \in \Omega$. To simplify the design choices, we consider the following 7 linear subspaces:

$$A_1 \triangleq S_1; \qquad A_2 \triangleq S_2; \tag{31}$$

$$A_3 \triangleq S_1 \oplus \Omega_1; \quad A_4 \triangleq S_2 \oplus \Omega_2; \quad A_5 \triangleq S_1 \oplus S_2; \tag{32}$$

$$A_6 \triangleq S_1 \oplus S_2 \oplus \Omega_1; \qquad A_7 \triangleq S_1 \oplus S_2 \oplus \Omega_2, \tag{33}$$

for which we use S_1 and S_2 as shorthand for $S_1(t-1)$ and $S_2(t-1)$, the knowledge spaces in the end of time $t-1$. In the subsequent discussion, we often drop the input argument "$(t-1)$" when the time instant of interest is clear in the context.

We can now partition the overall coding vector space Ω into $2^7 = 128$ disjoint subsets depending on whether $\mathbf{c}_t$ is in A_k or not, for $k = 1, \cdots, 7$. Each subset is termed a *coding type* and can be indexed by a 7-bit string $\mathbf{b} = b_1 b_2 \cdots b_7$ where each b_k indicates whether $\mathbf{c}_t \in A_k$ or not. For example, type-0010111 contains the coding vectors that are in $A_3 \cap A_5 \cap A_6 \cap A_7$, but not in any of A_1, A_2, and A_4. Those coding vectors are now denoted by

$$\begin{aligned}\mathsf{TYPE}_{23} &= \mathsf{TYPE}_{0010111} \\ &\triangleq (A_3 \cap A_5 \cap A_6 \cap A_7) \backslash (A_1 \cup A_2 \cup A_4) \qquad (34) \\ &= (A_3 \cap A_5) \backslash (A_1 \cup A_4) \qquad (35)\end{aligned}$$

where (35) follows from the fact that by (31) to (33) we have $A_5 \subseteq (A_6 \cap A_7)$ and $A_2 \subseteq A_4$. Note that some of the 128 coding types are always empty, which are termed the *infeasible types*. For example, type-1000000 is infeasible since there cannot be any $\mathbf{v} \in \Omega$ that is in $A_1 = S_1$ but not in $A_3 = S_1 \oplus \Omega_1 \supseteq A_1$. Overall, there are only 18 *Feasible Types* (FTs) and the list of them is the FTs defined in (14).

This new framework allows us to focus on the "types" of the coding choices without worrying about designing the exact values of the individual coordinates of $\mathbf{c}_t \in \Omega$. Specifically, we will focus on the following design problem: From which one of the 18 FTs should we choose $\mathbf{c}_t$ in order to maximize the throughput? We will also analyze the performance of any given scheme by quantifying how frequently a coding vector $\mathbf{c}_t$ of type-$\mathbf{b}$ is sent.

C. The "Only If" Analysis of Proposition 1 For The Case of $M = 1$

Fix any given linear network code such that d_i can decode all $X_{i,1}$ to X_{i,nR_i} in the end of time n for all $i = 1, 2$ with close-to-one probability. Since the 18 disjoint FTs fully cover[4] Ω, for each time t we can always label the coding choice $\mathbf{c}_t$ of the given LNC scheme as one of the 18 FTs. Define $x_{\mathbf{b}}^{[1]} \triangleq \frac{1}{n}\mathsf{E}\left\{\sum_{t=1}^{n} 1_{\{\mathbf{c}_t \in \mathsf{TYPE}_{\mathbf{b}}\}}\right\}$ as the normalized expected number of $\mathbf{c}_t$ of type $\mathbf{b}$. Since the total number of time slots is n, (15) holds for the case of $M = 1$ by the *time-sharing* argument. In the following, we will establish additional linear equalities/inequalities that govern the values of $x_{\mathbf{b}}$.

Consider the linear spaces A_k, $k = 1$ to 7, in the beginning of time 1 and in the end of time n, and denote them by $A_k(0)$ and $A_k(n)$, respectively. Consider $A_6 = S_1 \oplus S_2 \oplus \Omega_1$ for example. By (29) and (33) we have $\mathrm{Rank}(A_6(0)) = \mathrm{Rank}(\Omega_1) = nR_1$. We then note that when source s sends a $\mathbf{c}_t \in \mathsf{TYPE}_{\mathbf{b}}$ for some $\mathbf{b}$ with b_6 being 0, then that $\mathbf{c}_t$ is not in $A_6 = S_1 \oplus S_2 \oplus \Omega_1$. *Therefore, whenever one of d_1 and d_2 receives $W(t) = \overline{\mathbf{X}} \cdot \mathbf{c}_t^{\mathrm{T}}$ successfully, the rank of A_6 will increase by one.* We thus have

$$\mathrm{Rank}(A_6(0)) + \sum_{\forall \mathbf{b} \text{ w. } b_6 = 0} \left(\sum_{t=1}^{n} 1_{\left\{ \substack{\mathbf{c}_t \in \mathsf{TYPE}_{\mathbf{b}}, \text{ and} \\ \text{one of } \{d_1, d_2\} \text{ receives it}} \right\}} \right) = \mathrm{Rank}(A_6(n)) \tag{36}$$

Define $y_k \triangleq \frac{1}{n}\mathsf{E}\{\mathrm{Rank}(A_k(n))\}$ as the normalized expected rank of $A_k(n)$. Taking the normalized expectation of (36), counting only the FTs, and by the linearity of expectation and the stationarity and memorylessness of the channel, we have proven (21) for $M = 1$. By similar *rank-conversion* arguments, we can also prove (16) to (22) for $M = 1$. Detailed derivation of (16) to (22) is relegated to Appendix D.

In the following, we will derive the *rank comparison* inequalities in Group 3. By (31) to (33), in the end of time n we must have

$$A_3 \subseteq A_6, \quad A_4 \subseteq A_7, \quad A_6 \subseteq \Omega, \text{ and } A_7 \subseteq \Omega. \tag{37}$$

Considering the normalized expected ranks of the above inequalities in the end of time n, we have proven (23) and (24). Before continuing, we present the following self-explanatory lemma.

Lemma 6: For any two linear spaces B_1 and B_2, we have $\mathrm{Rank}(B_1 \oplus B_2) + \mathrm{Rank}(B_1 \cap B_2) = \mathrm{Rank}(B_1) + \mathrm{Rank}(B_2)$.

We then consider the following inequality:

$$\begin{aligned}&\mathrm{Rank}(S_1 \oplus S_2) + \mathrm{Rank}(S_1 \oplus \Omega_1) - \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_1) \\ &= \mathrm{Rank}((S_1 \oplus S_2) \cap (S_1 \oplus \Omega_1)) \qquad (38) \\ &\geq \mathrm{Rank}(S_1) \qquad (39)\end{aligned}$$

[4]The actual set of vectors in a type, e.g., (34), evolves over time since the A_k definitions in (31) to (33) depend on the knowledge spaces S_1 and S_2 in the end of time $t-1$. However, the 18 FTs always cover Ω for any t value.

where (38) follows from Lemma 6, and (39) follows from simple set operations. By taking the normalized expectation of (39) in the end of time n, we have proven (25). Similarly, we can derive the following inequalities:

$$\begin{aligned}&\mathrm{Rank}(S_1 \oplus S_2) + \mathrm{Rank}(S_2 \oplus \Omega_2)\\&\quad - \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2) \geq \mathrm{Rank}(S_2) \quad (40)\\&\mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_1) + \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2)\\&\quad - \mathrm{Rank}(\Omega) \geq \mathrm{Rank}(S_1 \oplus S_2), \quad (41)\end{aligned}$$

and use them to prove (26) and (27), respectively.

Finally, by definition we have

$$\begin{aligned}0 &\leq \mathrm{Rank}(S_i(n) \oplus \Omega_i) - \mathrm{Rank}(S_i(n))\\&\leq \mathrm{Rank}(S_i(n) \oplus \Omega_i) \leq \mathrm{Rank}(\Omega) = n \cdot (R_1 + R_2).\end{aligned}$$

By the *decodability* condition (30), $\mathsf{Prob}(\mathrm{Rank}(S_i(n) \oplus \Omega_i) \neq \mathrm{Rank}(S_i(n))) < \epsilon$. We thus have

$$\begin{aligned}0 &\leq \mathsf{E}\{\mathrm{Rank}(S_i(n) \oplus \Omega_i)\} - \mathsf{E}\{\mathrm{Rank}(S_i(n))\}\\&\leq \epsilon \cdot n \cdot (R_1 + R_2).\end{aligned}$$

Taking the normalized expectation and by (31) and (32), we have

$$\begin{aligned}&0 \leq y_3 - y_1 \leq \epsilon \cdot (R_1 + R_2)\\ \text{and} \quad &0 \leq y_4 - y_2 \leq \epsilon \cdot (R_1 + R_2). \quad (42)\end{aligned}$$

The above construction shows that given any $\epsilon > 0$, we can use the corresponding LNC scheme to compute the $x_{\mathbf{b}}^{[1]}$ and y_k values that satisfy (15)–(27) and (42). Since the above construction holds for arbitrarily small $\epsilon > 0$, it is guaranteed that the feasible region of the linear (in)equalities (15) to (28) is non-empty. The above discussion thus proves that for the case of $M = 1$, if (R_1, R_2) is LNC-achievable, there exist $x_{\mathbf{b}}^{[1]}$ and y_1 to y_7 values satisfying Proposition 1.

D. The "If" Analysis of Proposition 1 For the Case of $M = 1$ — Part I: A Polytope-Based View

For the achievability proofs in this section, we temporarily assume that the underlying $\mathsf{GF}(q)$ satisfies $q \geq 3$. In Section VI-C, we will relax this assumption and allow for the case of $\mathsf{GF}(2)$.

A critical difference between Lemma 2 and the outer bounding part of Proposition 1 is that the latter is a constructive approach while the former is implicitly a cut condition. As will be demonstrated, any $\{x_{\mathbf{b}}^{[1]} : \forall \mathbf{b} \in \mathsf{FTs}\}$ and y_1 to y_7 satisfying Proposition 1 can be directly translated to a rate-(R_1, R_2) LNC scheme. On the other hand, it is not clear how the auxiliary variables $R_i^{[m,k]}$ in Lemma 2 could guide the LNC design.

We first introduce some notation before elaborating the main ideas of the new achievability scheme. Consider $M = 1$ and any rate vector (R_1, R_2) that is in the interior[5] of the capacity region in Proposition 1. Denote the companying variable values in Proposition 1 by $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b} \in \mathsf{FTs}\}$ and $\ddot{y}_1$ to $\ddot{y}_7$ such that jointly these variable values and (R_1, R_2) satisfy (15) to (28). Note that the difference between $\ddot{x}_{\mathbf{b}}^{[1]}$ and $x_{\mathbf{b}}^{[1]}$ is that the former is a constant scalar value corresponding to the (R_1, R_2) of interest while the latter is a variable of a linear programming problem. For simplicity, oftentimes we use the notation $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$ that implicitly assumes that $\mathbf{b}$ must be in FTs.

Our main idea is as follows. For any t, in the beginning of time t source s first uses the previous reception status $[\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}$ to determine the subspaces A_1 to A_7. Then we simply let s choose the $\mathbf{c}_t$ from[6] one of the 18 FTs as discussed in Section V-B. To better describe the operations, we define random processes $X_{\mathbf{b}}^{[1]}(t)$, $\forall \mathbf{b} \in \mathsf{FTs}$, as follows:

$$\forall \mathbf{b} \in \mathsf{FTs}, \quad X_{\mathbf{b}}^{[1]}(t) = \frac{1}{n} \sum_{\tau=1}^{t} 1_{\{\mathbf{c}_\tau \in \mathsf{TYPE}_{\mathbf{b}}\}} \quad (43)$$

being the cumulative frequency[7] of using type-$\mathbf{b}$ coding choices until time t.

We then have the following critical observation:

> *If we can choose the coding type $\mathbf{b}$ used at time t for all $t = 1, \cdots, n$ in a way that ensures that in the end of time n, the cumulative frequencies $X_{\mathbf{b}}^{[1]}(n) = \ddot{x}_{\mathbf{b}}^{[1]}$ for all $\mathbf{b} \in \mathsf{FTs}$, then the outer bound analysis in Section V-C guarantees that the resulting LNC scheme achieves rate (R_1, R_2) with close-to-one probability.*

The reason is as follows. If we can attain the desired frequency $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$ with probability one, then the proof of the rank-conversion equalities in Section V-C ensures that for this particular scheme, for any $k = 1, \cdots, 7$, the normalized expected rank of $A_k(n)$ is the same as the $\ddot{y}_k$ value corresponding to the target frequency $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$. Therefore, in the end of time n, we will have $\mathsf{E}\{\mathrm{Rank}(A_1(n))\} = \mathsf{E}\{\mathrm{Rank}(A_3(n))\}$. Note that by definition we always have $\mathrm{Rank}(A_1(n)) \leq \mathrm{Rank}(A_3(n))$. This thus implies $\mathrm{Rank}(A_1(n)) = \mathrm{Rank}(A_3(n))$ with probability one and d_1 can decode its desired messages. By symmetry, so does d_2. Note that in practice, we may not be able to attain the target frequency $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$ with probability one. However, as long as the $\{X_{\mathbf{b}}^{[1]}(n) : \forall \mathbf{b}\}$ can be made *sufficiently close* to $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$, then we can achieve $\geq (1 - \epsilon)$ fraction of the target rate pair (R_1, R_2) with probability $\geq (1 - \epsilon)$ for arbitrary $\epsilon > 0$ by the continuous nature of the LP problem and by the law of large numbers. As can be seen from this argument, the decodability condition (the Group 4 equalities in Proposition 1) is now converted to a new goal of attaining the target relative frequency $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$.

[5]Being in the interior of the capacity region means that both R_1 and R_2 are strictly positive and there exists a small $\delta > 0$ such that rate $(R_1 + \delta, R_2 + \delta)$ is also in the capacity region.

[6]If there are more than one vector in a FT of interest, then s chooses arbitrarily one of them.

[7]The cumulative frequency up to time t is often defined with $\frac{1}{t}$ as the normalization factor. However, in (43) we use $\frac{1}{n}$ as the normalization factor instead. The benefit of using $\frac{1}{n}$ as the normalization factor is that we can concentrate on the increments over time without worrying about the time-dependent normalization factor $\frac{1}{t}$.

Also, for comparison, $X_{\mathbf{b}}^{[1]}(t)$ is the cumulative frequency until time t while $X_{i,1}$ to X_{i,nR_i} are the message symbols.

Before proceeding, we first describe *the polytope of the cumulative frequency,* which, for the case of $M = 1$, is a polytope in the 18-dimensional Euclidean space. We now specify how to construct this polytope. Fix (R_1, R_2). We first notice that among the four groups of conditions, Group 2 the rank conversion conditions are equalities that convert $\{x^{[1]}_{\mathbf{b}} : \forall\mathbf{b}\}$ to $\{y_1, \cdots, y_7\}$. As a result, we can substitute all the y variables in Group 3 by the equalities of Group 2. In the end, Groups 1 and 3 become $(1 + 7)$ linear inequalities governing the 18 non-negative variables $\{x^{[1]}_{\mathbf{b}} : \forall\mathbf{b}\}$. We now define Γ, the polytope of the cumulative frequency, as the polytope[8] in the first quadrant of the 18-dimensional Euclidean space governed by the inequalities of Groups 1 and 3.

We then note that in the sense of the first-order analysis, for any time t the empirical cumulative frequency $\{X^{[1]}_{\mathbf{b}}(t) : \forall\mathbf{b}\}$ must be inside the polytope Γ since $\sum_{\forall\mathbf{b}} X^{[1]}_{\mathbf{b}}(t) = \frac{t}{n} \leq 1$ and the rank-comparison inequalities are derived by the relationships among the linear subspaces A_1 to A_7 and thus hold for any time instant t. Also note that by the definition in (43), we have $X^{[1]}_{\mathbf{b}}(0) = 0$ for all $\mathbf{b} \in \mathsf{FTs}$. As a result, from this polytope interpretation *the new goal becomes to make sure that as time t increases, the cumulative frequency $\{X^{[1]}_{\mathbf{b}}(t) : \forall\mathbf{b}\}$ until time t has a trajectory within the polytope Γ that starts from the origin and ends in the 18-dimensional point $\vec{\ddot{x}} \triangleq \{\ddot{x}^{[1]}_{\mathbf{b}} : \forall\mathbf{b}\}$.*

Since there are infinitely many ways of designing a trajectory $\{X^{[1]}_{\mathbf{b}}(t) : \forall\mathbf{b}\}$ in Γ with the start point being the origin and the end point being $\vec{\ddot{x}}$, there are infinitely many ways of designing an LNC scheme achieving rates (R_1, R_2). We propose to use the following "tunneling approach," which is named after the fact that its goal is to create *a tunnel in the polytope* Γ.

More specifically, the tunneling approach takes the most straightforward approach, which ensures that the trajectory $\{X^{[1]}_{\mathbf{b}}(t) : \forall\mathbf{b}\}$ follows a straight "tunnel" connecting the origin and the end point $\vec{\ddot{x}}$. Note that in general the trajectory cannot be a straight *line* connecting $\vec{\mathbf{0}}$ and $\vec{\ddot{x}}$. The reason is that for every time t, we can only choose one of the 18 different coding types. Therefore, the trajectory can only zigzag along the 18 orthogonal axes and cannot follow a straight line connecting the origin and $\vec{\ddot{x}}$. As will be formalized in the later discussion, *the tunneling approach first defines a tunnel with strictly positive volume and then allows the trajectory to zigzag within the tunnel and finally reach the end point $\vec{\ddot{x}}$.*

We conclude this subsection by discussing some properties of the 18-dimensional polytope.

Lemma 7: The above 18-dimensional polytope, denoted by Γ, satisfies the following properties:

- Property 1: Both the point $\vec{\ddot{x}}$ and the origin belong to Γ.
- Property 2: If (R_1, R_2) is in the interior of the capacity region in Proposition 1 and the 2-receiver broadcast PEC is not physically degraded, then the interior of Γ is non-empty.

The proof of Lemma 7 is relegated to Appendix E.

[8]One can see that Γ depends on the given target rates (R_1, R_2), which are assumed to be constant in our discussion.

E. *The "If" Analysis of Proposition 1 For the Case of $M = 1$ — Part II: The Tunneling Approach*

We now define the tunnel in a given polytope.

Definition 5: Consider two points $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ in a polytope Γ such that each coordinate of $\vec{x}_{\text{end}}$ is strictly larger than each coordinate of $\vec{x}_{\text{start}}$. The line connecting $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ contains all the convex combinations of the terminal points $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$. For any $\delta > 0$, a size-δ tunnel T connecting $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ is the maximum (closed) set of points satisfying the following conditions: (i) $\vec{x} \in \mathsf{T}$ implies that $\vec{x}$ satisfies the coordinate-wise inequality $\vec{x}_{\text{start}} \leq \vec{x} \leq \vec{x}_{\text{end}}$; and (ii) $\vec{x} \in \mathsf{T}$ implies that there exists a point $\vec{y}$ in the line connecting $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ such that the Euclidean distance $d(\vec{x}, \vec{y}) \leq \delta$.

The intuition behind the above definition is straightforward. Condition (ii) allows each tunnel to have a positive volume; and Condition (i) ensures that a tunnel is confined between the two terminal points $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$.

Definition 6: A tunnel T of a polytope Γ is *proper* if T is contained in the interior of Γ.

As discussed in Section V-D, the goal is to design an LNC scheme with the trajectory of $\{X^{[1]}_{\mathbf{b}}(t) : \forall\mathbf{b}\}$ confined in a proper tunnel T connecting $\mathbf{0}$ and $\vec{\ddot{x}}$. Unfortunately, such a proper tunnel T does not exist in general. As a result, we slightly revise our approach in the following way: For any $\epsilon > 0$ we find a pair of starting and ending nodes $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ in the interior of Γ that satisfy

$$\vec{x}_{\text{start}} < \vec{x}_{\text{end}} \text{ in a coordinate-wise sense;} \tag{44}$$

$$d(\vec{\mathbf{0}}, \vec{x}_{\text{start}}) < \epsilon \text{ and } d(\vec{x}_{\text{end}}, \vec{\ddot{x}}) < \epsilon. \tag{45}$$

We can then divide the trajectory construction into 3 phases: Phase 1: The first mile from $\vec{\mathbf{0}}$ to $\vec{x}_{\text{start}}$; Phase 2: From $\vec{x}_{\text{start}}$ to $\vec{x}_{\text{end}}$ along a proper tunnel T; and Phase 3: The last mile from $\vec{x}_{\text{end}}$ to $\vec{\ddot{x}}$.

For the following discussion, we assume that the 2-receiver broadcast PEC is not physically degraded. The physically-degraded setting can be viewed as a degenerate case and the corresponding achievability scheme can be derived in a similar way.

Property 2 of Lemma 7 ensures that we can always[9] find in the interior of Γ the starting and ending node pair $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ that satisfy (44) and (45). Since both $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$ are in the interior of Γ, there is a proper tunnel T connecting $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$. Since (45) is satisfied, it is intuitive that the amount of time necessary for executing Phases 1 and 3 is roughly $\mathcal{O}(\epsilon n)$ time slots. The time duration of executing Phases 1 and 3 is thus negligible[10] when we use a sufficiently small

[9]Lemma 7 ensures that $\mathbf{0}$ and $\vec{\ddot{x}}$ are in (the boundary of) the polytope Γ. Lemma 7 also ensures that the interior of Γ is not empty. Therefore, we can always find an $\vec{x}_{\text{start}}$ in the interior that is arbitrarily close to $\mathbf{0}$ and find another $\vec{x}_{\text{end}}$ in the interior that is arbitrarily close to the final destination $\vec{\ddot{x}}$, which satisfy the distance requirement (45). Moreover, since $\vec{x}_{\text{end}}$ is in the interior of Γ, all its coordinates are strictly positive. Since $\vec{x}_{\text{start}}$ only needs to be chosen arbitrarily close to $\mathbf{0}$, we can thus choose $\vec{x}_{\text{start}}$ to also satisfy (44).

[10]More explicitly, in Appendix F we have shown that the amount of time to execute Phase 1 is $\mathcal{O}(\epsilon n)$ time slots and the information-loss for *skipping Phase 3* is also $\mathcal{O}(\epsilon n)$ packets when compared to the nR_1 and nR_2 information symbols. As a result, the throughput loss from the normalized rate region's perspective can be made arbitrarily small when choosing a sufficiently small ϵ in (45).

ϵ and a sufficiently large n. In the following, we thus only explain how to implement Phase 2 while relegating the detailed implementation of Phases 1 and 3 to Appendix F.

We first describe how to implement *Phase 2: From $\vec{x}_{start}$ to $\vec{x}_{end}$ along a proper tunnel* T. This phase can be divided into finitely many sub-phases. For the beginning of each sub-phase, say time t, if the cumulative frequency $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\}$ is not the same as the end point $\vec{x}_{\text{end}}$, then by the definition of the tunnel T, there must exist at least one $\tilde{\mathbf{b}} \in \mathsf{FTs}$ and a sufficiently small $\delta_{\tilde{\mathbf{b}}} > 0$ such that the 18-dimension point $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b} \neq \tilde{\mathbf{b}}\} \cup \{X_{\tilde{\mathbf{b}}}^{[1]}(t) + \delta_{\tilde{\mathbf{b}}}\}$ is still inside the tunnel T. Choose arbitrarily one such $\tilde{\mathbf{b}}$ and fix it. Then choose the largest possible $\delta_{\tilde{\mathbf{b}}}$ such that $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b} \neq \tilde{\mathbf{b}}\} \cup \{X_{\tilde{\mathbf{b}}}^{[1]}(t) + \delta_{\tilde{\mathbf{b}}}\}$ is still inside the tunnel T.

After selecting the $\tilde{\mathbf{b}}$ and $\delta_{\tilde{\mathbf{b}}}$ values, we will keep sending coding vectors of type-$\tilde{\mathbf{b}}$ for the current sub-phase, and this sub-phase will last for $n \cdot \delta_{\tilde{\mathbf{b}}}$ time slots. After the current sub-phase ends, i.e., in the end of time $t' \triangleq t + n\delta_{\tilde{\mathbf{b}}}$, we will restart the process and choose a new set of $\tilde{\mathbf{b}}$ and $\delta_{\tilde{\mathbf{b}}}$ based on $\{X_{\mathbf{b}}^{[1]}(t') : \forall \mathbf{b}\}$. Repeat this process until the cumulative frequency becomes $\vec{x}_{\text{end}}$. One can easily see that this iterative construction ends after finitely[11] many sub-phases. Moreover, since $\vec{x}_{\text{end}}$ is in the interior of Γ, it implies that $\vec{x}_{\text{end}}$ satisfies (15) with strict inequality. By the definition of $X_{\mathbf{b}}^{[1]}(t)$ in (43), when $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\} = \vec{x}_{\text{end}}$, we must have $t < n$. That is, Phase 2 will finish within the time budget n.

To prove the correctness of the above implementation, we need to show that for each sub-phase the chosen coding type $\mathsf{TYPE}_{\tilde{\mathbf{b}}}$ must remain non-empty for the allocated time duration of $n\delta_{\tilde{\mathbf{b}}}$ time slots with close-to-one probability so that with close-to-one probability we can choose a coding vector from $\mathsf{TYPE}_{\tilde{\mathbf{b}}}$ for every time slot of the sub-phase of interest. Instead of proving a particular $\mathsf{TYPE}_{\tilde{\mathbf{b}}}$ being non-empty, we prove the following stronger statement instead: Consider any fixed $\mathsf{GF}(q)$ with $q \geq 3$. *If the cumulative frequency $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\}$ is within the proper tunnel T in time t, then with close-to-one probability $\mathsf{TYPE}_{\mathbf{b}} \neq \emptyset$ for all $\mathbf{b} \in \mathsf{FTs}$.* This implies that we can freely choose (from the 18 FTs) at any time t as long as we let $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\}$ stay within the proper tunnel T. Such a fact/observation is the building foundation of the tunneling approach.

Again we rely on the law of large numbers and the first-order expectation-based analysis to prove the above central statement. For any $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\} \in \mathsf{T}$ in time t, since T is a proper tunnel, it must satisfy (15) and (23)–(27) with strict inequality. Since (15) is satisfied with strict inequality, we have not used up the total time budget of n time slots and we can still choose a coding vector for the current time slot. Since (23) to (27) are satisfied with strict inequality in the beginning of the current time slot t, by the rank conversion arguments in Section V-C, the set/rank inequalities in (37)–(41) for time t are strict inequalities with close-to-one probability. What remains to be proven is that when (37)–(41) are strict inequalities, $\mathsf{TYPE}_{\mathbf{b}} \neq \emptyset$ for all $\mathbf{b} \in \mathsf{FTs}$.

TABLE I
THE CODING TYPES AND THE ASSOCIATED RANK COMPARISON CONDITIONS THAT GUARANTEE NON-EMPTINESS.

$\mathsf{TYPE}_0 \Leftrightarrow A_6 \subsetneq \Omega, A_7 \subsetneq \Omega$;	$\mathsf{TYPE}_1 \Leftrightarrow A_6 \subsetneq \Omega, A_4 \subsetneq A_7$;
$\mathsf{TYPE}_2 \Leftrightarrow A_7 \subsetneq \Omega, A_3 \subsetneq A_6$;	$\mathsf{TYPE}_3 \Leftrightarrow^{\dagger}$ strict (41), $A_3 \subsetneq A_6$, $A_4 \subsetneq A_7$;
$\mathsf{TYPE}_7 \Leftrightarrow A_3 \subsetneq A_6, A_4 \subsetneq A_7$;	$\mathsf{TYPE}_{11} \Leftrightarrow$ strict (41), $A_3 \subsetneq A_6$;
$\mathsf{TYPE}_9 \Leftrightarrow A_6 \subsetneq \Omega$;	$\mathsf{TYPE}_{18} \Leftrightarrow A_7 \subsetneq \Omega$;
$\mathsf{TYPE}_{15} \Leftrightarrow$ strict (40), $A_3 \subsetneq A_6$;	$\mathsf{TYPE}_{23} \Leftrightarrow$ strict (39), $A_4 \subsetneq A_7$;
$\mathsf{TYPE}_{19} \Leftrightarrow$ strict (41), $A_4 \subsetneq A_7$;	$\mathsf{TYPE}_{31} \Leftrightarrow$ strict (39) and (40);
$\mathsf{TYPE}_{27} \Leftrightarrow$ strict (41);	$\mathsf{TYPE}_{63} \Leftrightarrow$ strict (39);
$\mathsf{TYPE}_{47} \Leftrightarrow A_3 \subsetneq A_6$;	$\mathsf{TYPE}_{95} \Leftrightarrow$ strict (40);
$\mathsf{TYPE}_{87} \Leftrightarrow A_4 \subsetneq A_7$;	
TYPE_{127} is always non-empty.	

† For coding type 3, we further assume the underlying $\mathsf{GF}(q)$ satisfies $q \geq 3$.

We first prove the case in which $\mathbf{b} = 23$. Note that TYPE_{23} is defined in (35) and whether $\mathsf{TYPE}_{23} \neq \emptyset$ holds can be decided by comparing the sizes of $(A_3 \cap A_5)$ and $((A_3 \cap A_5) \cap (A_1 \cup A_4))$. Also note that $|A_3 \cap A_5| = q^{\mathrm{Rank}(A_3 \cap A_5)}$ and

$$\begin{aligned} & q^{\max(\mathrm{Rank}(A_3 \cap A_5 \cap A_1), \mathrm{Rank}(A_3 \cap A_5 \cap A_4))} && (46) \\ & = \max(|A_3 \cap A_5 \cap A_1|, |A_3 \cap A_5 \cap A_4|) \\ & \leq |(A_3 \cap A_5) \cap (A_1 \cup A_4)| && (47) \\ & \leq |A_3 \cap A_5 \cap A_1| + |A_3 \cap A_5 \cap A_4| - 1 && (48) \\ & = q^{\mathrm{Rank}(A_3 \cap A_5 \cap A_1)} + q^{\mathrm{Rank}(A_3 \cap A_5 \cap A_4)} - 1, && (49) \end{aligned}$$

where (47) follows from simple set operations and (48) follows from that the origin $\mathbf{0}$ is always in the intersection of two subspaces $A_3 \cap A_5 \cap A_1$ and $A_3 \cap A_5 \cap A_4$. Since $q \geq 2$, we thus have $\mathsf{TYPE}_{23} \neq \emptyset$ if and only if

$$\begin{aligned} & \mathrm{Rank}(A_3 \cap A_5) \\ & > \max(\mathrm{Rank}(A_3 \cap A_5 \cap A_1), \mathrm{Rank}(A_3 \cap A_5 \cap A_4)). && (50) \end{aligned}$$

In Appendix G we have proven that

$$\begin{aligned} & \mathrm{Rank}(A_3 \cap A_5) - \mathrm{Rank}(A_3 \cap A_5 \cap A_1) \\ & = \mathrm{Rank}(S_1 \oplus \Omega_1) + \mathrm{Rank}(S_1 \oplus S_2) \\ & \quad - \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_1) - \mathrm{Rank}(S_1) && (51) \end{aligned}$$

and

$$\begin{aligned} & \mathrm{Rank}(A_3 \cap A_5) - \mathrm{Rank}(A_3 \cap A_5 \cap A_4) \\ & = \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2) - \mathrm{Rank}(S_2 \oplus \Omega_2) \\ & = \mathrm{Rank}(A_7) - \mathrm{Rank}(A_4). \end{aligned}$$

The above analysis shows that when both (39) and $A_4 \subsetneq A_7$ are strict inequalities, $\mathsf{TYPE}_{23} \neq \emptyset$ and choosing a $\mathbf{c}_t$ from TYPE_{23} is possible. By similar arguments, it can be proven that (i) Each FT is associated with a subset of inequalities of (37)–(41); and (ii) A FT is non-empty if and only if the inequalities in the corresponding subset are all strict. Table I summarizes the mapping from "the FT of interest being non-empty" to "the corresponding subset of inequalities being strict." The detailed derivation of Table I is relegated to Appendix G.

Therefore, when all 7 inequalities in (37)–(41) are strict, all 18 $\mathsf{TYPE}_{\mathbf{b}}$ are non-empty. As a result, we have proven that if the cumulative frequency $\{x_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\}$ is within the proper

[11] The number of sub-phases depends only on the size of the tunnel T and does not depend on n, the block length of the network code. This observation is critical when applying the arguments of the law of large numbers. Also see the discussion in Appendix A-C.

tunnel T, then with close-to-one probability $\mathsf{TYPE}_{\mathbf{b}} \neq \emptyset$ for all $\mathbf{b} \in \mathsf{FTs}$. The "If" analysis of Proposition 1 is complete.

VI. Proof of Proposition 1 for Arbitrary M

The proof for the case of $M = 1$ can be readily generalized for the case of arbitrary M.

A. The "Only If" Analysis of Proposition 1 For The Case of Arbitrary M

We follow the knowledge-space-based approach as in the case of $M = 1$. Specifically, recall that the $(nR_1 + nR_2) \times M$ coding matrix $\mathbf{C}_t$ for time t can be described as $\mathbf{C}_t = [\mathbf{c}_{t,1}^{\mathrm{T}}, \mathbf{c}_{t,2}^{\mathrm{T}}, \cdots, \mathbf{c}_{t,M}^{\mathrm{T}}]$ where the $n(R_1 + R_2)$-dimensional row vector $\mathbf{c}_{t,m}$ describes the coding vector used to generate the m-th input $W^{[m]}$ at time t for all $m \in \{1, \cdots, M\}$. For any $i \in \{1, 2\}$, $t \in \{1, \cdots, n\}$, and $m \in \{1, \cdots, M\}$, we now define the knowledge space $S_i(t.m)$ by

$$\begin{aligned} S_i(t.m) \triangleq \mathsf{span}\{\mathbf{c}_{\tau,h} : \forall \tau \text{ and } h \text{ such that either} \\ \text{(i) } \tau < t \text{ and } h \in \mathsf{rx}_i \text{ at time } \tau; \\ \text{or (ii) } \tau = t,\ h \leq m, \text{ and } h \in \mathsf{rx}_i \text{ at time } t\}. \end{aligned} \tag{52}$$

That is, $S_i(t.m)$ is the linear span of the vectors of those packets that have successfully arrived at d_i either before time instant t or at time instant t but only counting the vectors from the first m input packets $W^{[1]}(t)$ to $W^{[m]}(t)$. In the broadest sense, it is equivalent to splitting each time index t into M *sub time instants* denoted by $t.1$ to $t.M$ and assuming that each of the M inputs $W^{[1]}(t)$ to $W^{[M]}(t)$ is sent sequentially at the split time instant $t.m$, $m = 1$ to M.

In the beginning of any split time instant $t.m$, we define A_1 to A_7 in the same way as in (31) to (33) based on the knowledge spaces $S_1(t.(m-1))$ and $S_2(t.(m-1))$ in the end of the split time instant $t.(m-1)$. We can also define 18 feasible coding types $\mathsf{TYPE}_{\mathbf{b}}$ in the same way as described in Section V-B.

Fix any given linear network code such that d_i can decode all $X_{i,1}$ to X_{i,nR_i} in the end of time n for all $i = 1, 2$ with close-to-one probability. Since the 18 disjoint FTs fully cover Ω, for each split time instant $t.m$ we can always label the coding choice $\mathbf{c}_{t,m}$ of the given LNC scheme as one of the 18 FTs. Define

$$x_{\mathbf{b}}^{[m]} \triangleq \frac{1}{n} \mathsf{E}\left\{ \sum_{t=1}^{n} 1_{\{\mathbf{c}_{t,m} \in \mathsf{TYPE}_{\mathbf{b}} \text{ in the beginning of time } t.m\}} \right\}$$

as the normalized expected number of $\mathbf{c}_{t,m}$ of type $\mathbf{b}$ where $\mathsf{TYPE}_{\mathbf{b}}$ is evaluated in the beginning of the split time instant $t.m$. Since the total number of time slots is n, the above construction of $x_{\mathbf{b}}^{[m]}$ must satisfy (15) for the case of arbitrary M values.

We now establish the rank conversion inequalities (16) to (22). Consider the linear spaces A_k, $k = 1$ to 7, in the beginning of the split time instant 1.1 and in the end of the split time instant time $n.M$, and denote them by $A_k(0)$ and $A_k(n.M)$, respectively. Consider $A_6 = S_1 \oplus S_2 \oplus \Omega_1$ for example. By (52) and the same reasoning as in Section V-C, we have

$$\begin{aligned} &\mathrm{Rank}(A_6(0)) \\ &\quad + \sum_{m=1}^{M} \sum_{\forall \mathbf{b} \text{ w. } b_6 = 0} \left(\sum_{t=1}^{n} 1_{\left\{ \substack{\mathbf{c}_{t,m} \in \mathsf{TYPE}_{\mathbf{b}} \text{ in the} \\ \text{beginning of time } t.m\text{; and} \\ \text{one of } \{d_1, d_2\} \text{ receives it}} \right\}} \right) \\ &= \mathrm{Rank}(A_6(n.M)). \end{aligned} \tag{53}$$

Define $y_k \triangleq \frac{1}{n}\mathsf{E}\{\mathrm{Rank}(A_k(n.M))\}$ as the normalized expected rank of $A_k(n.M)$. Taking the normalized expectation of (53), counting only the FTs, noting that the channel is cross-input independent, and by the linearity of expectation and the stationarity and memorylessness of the channel, we have proven that (21) must hold for arbitrary M values. By similar *rank-conversion* arguments, we can also prove (16) to (22) for arbitrary M values.

Note that the argument for the rank comparison inequalities and the decodability equalities in Section V-C does not depend on the value of M. Therefore the rank comparison inequalities and the decodability equalities still hold. The "Only If" direction of Proposition 1 has thus been proven for arbitrary M values.

B. The "If" Analysis of Proposition 1 For the Case of Arbitrary M

Benefiting from the constructive nature of our approach, we can easily prove the following statement by the same arguments as used in Sections V-D and V-E.

> Suppose that the source node s designs the coding vectors $\mathbf{c}_{t,m}$ sequentially in the order of $(t, m) = (1,1), (1,2), \cdots, (1, M), (2,1), \cdots, (2, M), (3,1), \cdots, (n, M)$. Also suppose when s designs $\mathbf{c}_{t,m}$, there is a genie informing s the reception status of all $\mathbf{c}_{\tilde{t},\tilde{m}}$ for all $(\tilde{t}, \tilde{m})$ before (t, m). Then we can achieve any (R_1, R_2) that is in the interior of Proposition 1.

That is, we simply need to use the tunneling approach to achieve the desired long-term relative frequency $\{\ddot{x}_{\mathbf{b}}^{[m]} : \forall \mathbf{b}, m\}$ in the same way as described in Sections V-D and V-E. However, the above statement alone is not sufficient to prove the "If" direction of Proposition 1 due to the fact that there exists no such genie. More specifically, *each destination d_i can only report its reception status rx_i after the time instant t rather than reporting its reception after every split time instant $t.m$.* As a result, in the beginning of time t, source s needs to design the coding vectors $\mathbf{c}_{t,m}$ for all $m \in \{1, \cdots, M\}$ simultaneously, and cannot design $\mathbf{c}_{t,m}$ sequentially since the latter requires the information of the knowledge spaces $S_i(t.(m-1))$ for every split time instant $t.m$. In the following, we argue that even without the help from the genie, source s can still achieve the outer bound established in Section VI-A.

The main difference between a genie-aided solution and a practical solution is the following. For any $t \in \{1, \cdots, n\}$ and $m \in \{1, \cdots, M\}$, when designing the coding vector $\mathbf{c}_{t,m}$ for the m-th input at time t, a genie-aided solution has the information of the knowledge spaces $S_1(t.(m-1))$ and $S_2(t.(m-1))$ in the end of the last split time instant $t.(m-1)$ while a practical solution can only access the information of the knowledge spaces $S_1((t-1).M)$ and $S_2((t-1).M)$ in the end of the last time slot $(t-1)$. With the correctly updated information, the coding type $\mathsf{TYPE}_{\mathbf{b}}$ chosen by a genie-aided solution reflects correctly the relative contribution of the m-th input $W^{[m]}$ at time t with respect to the previous split time instants $(\tilde{t},\tilde{m})$ before (t,m). On the other hand, when a practical scheme chooses to send a coding type $\mathsf{TYPE}_{\mathbf{b}}$ over the m-th input $W^{[m]}$ at time t, the decision is based on the outdated information $S_i((t-1).M)$, which does not capture any changes due to the random reception status of the first $(m-1)$ inputs $W^{[1]}$ to $W^{[m-1]}$. Therefore, the practical scheme cannot predict the *actual contribution* of $W^{[m]}$ when it is received by the destination d_i. As a result, even though s intends to send a $\mathbf{c}_{t,m}$ of type-$\mathbf{b}$, when the destination d_i actually receives it, the $\mathbf{c}_{t,m}$ may have the same effect as sending a different coding type $\mathbf{b}'$.

Consider the following example with $M=2$, $nR_1=2$, and $nR_2=1$. That is, we have two symbols X_1 and X_2 for d_1 and one symbol Y for d_2. The overall message space is $\Omega=(\mathsf{GF}(q))^3$ and the individual message spaces are $\Omega_1=\mathsf{span}\{(1,0,0),(0,1,0)\}$ and $\Omega_2=\mathsf{span}\{(0,0,1)\}$. Suppose in the very beginning of the transmission, source s would like to send a type-18 (type-0010010) coding vector via the first input $W^{[1]}$ and send a type-0 vector via the second input $W^{[2]}$. Since neither d_1 nor d_2 has received anything in the past, we have $S_1=\{(0,0,0)\}=S_2$. As a result, one can verify that the coding vector $\mathbf{c}_1=(1,0,0)$ is a type-18 coding vector and $\mathbf{c}_2=(1,0,1)$ is a type-0 coding vector.

Suppose s sends out $W^{[1]}=\overline{\mathbf{X}}\mathbf{c}_1^{\mathrm{T}}=X_1$ and $W^{[2]}=\overline{\mathbf{X}}\mathbf{c}_2^{\mathrm{T}}=X_1+Y$, and also suppose that $W^{[1]}$ is received successfully by d_2. Then after d_2 receiving $W^{[1]}$, the knowledge space of d_2 changes to $S_2=\mathsf{span}\{(1,0,0)\}$. One can now verify that with the new S_2, the coding vector $\mathbf{c}_2=(1,0,1)$ becomes a type-9 (type-0001001) vector instead of type-0. As a result, the the actual/effective coding types become $(\mathsf{TYPE}_{18},\mathsf{TYPE}_9)$, which deviate from the intended $(\mathsf{TYPE}_{18},\mathsf{TYPE}_0)$ combination.

To circumvent this problem, we can either rely on the ideas of random linear network coding (RLNC) [8], which requires a sufficiently large q, or design a scheme for small $q\geq 2$ at the cost of higher complexity. In the following, we demonstrate the RLNC-based solution. The code design for small $q\geq 2$ is relegated in Section VI-C.

An RLNC-based solution: Continue from our previous example. When s is interested in sending out a type-0 vector over $W^{[2]}$, we let s choose $\mathbf{c}_2$ uniformly randomly from the corresponding set

$$\Omega\backslash(A_1\cup A_2\cup\cdots\cup A_7)=\Omega\backslash(\Omega_1\cup\Omega_2) \tag{54}$$

where the linear subspaces A_1 to A_7 are evaluated based on the knowledge spaces $S_1=S_2=\{(0,0,0)\}$ in the very beginning of the transmission. There are (q^3-q^2-q+1) number of choices of $\mathbf{c}_2=(\alpha_1,\alpha_2,\alpha_3)$ that are in (54) and we pick the to-be-transmitted $\mathbf{c}_2$ uniformly randomly from all those choices. Let us now consider the same scenario in which d_2 has successfully received $W^{[1]}=X_1$ and the new knowledge space of d_2 becomes $S_2=\mathsf{span}\{(1,0,0)\}$. One can easily verify that out of the (q^3-q^2-q+1) choices of $\mathbf{c}_2=(\alpha_1,\alpha_2,\alpha_3)$ vectors, (q^3-2q^2+q) of them are still of type-0 even when evaluated using the latest knowledge space $S_2=\mathsf{span}\{(1,0,0)\}$. Only (q^2-2q+1) of the choices belong to some other type-$\mathbf{b}$, $\mathbf{b}\neq 0$, when evaluated with the latest $S_2=\mathsf{span}\{(1,0,0)\}$. When a sufficiently large q is used, with close-to-one probability the intended type-0 vector $\mathbf{c}_2$ (evaluated based on the outdated S_2) remains a type-0 vector regardless of the reception status of the first input $W^{[1]}$.

It turns out that for any time t we can use RLNC (i.e., choosing the coding vectors $\mathbf{c}_{t,m}$ uniformly randomly from the intended $\mathsf{TYPE}_{\mathbf{b}}$) to circumvent the potential knowledge-space-mismatch problem as long as the following *strengthened* rank comparison inequalities hold in the beginning of time t (or equivalently in the end of time $(t-1).M$):

$$\mathsf{gap}\triangleq 7(M-1)+1 \tag{55}$$
$$\mathrm{Rank}(A_6)-\mathrm{Rank}(A_3)>\mathsf{gap} \tag{56}$$
$$\mathrm{Rank}(A_7)-\mathrm{Rank}(A_4)>\mathsf{gap} \tag{57}$$
$$\mathrm{Rank}(\Omega)-\mathrm{Rank}(A_6)>\mathsf{gap} \tag{58}$$
$$\mathrm{Rank}(\Omega)-\mathrm{Rank}(A_7)>\mathsf{gap} \tag{59}$$
$$\mathrm{Rank}(A_3)+\mathrm{Rank}(A_5)-\mathrm{Rank}(A_1)-\mathrm{Rank}(A_6)>\mathsf{gap} \tag{60}$$
$$\mathrm{Rank}(A_4)+\mathrm{Rank}(A_5)-\mathrm{Rank}(A_2)-\mathrm{Rank}(A_7)>\mathsf{gap} \tag{61}$$
$$\mathrm{Rank}(A_6)+\mathrm{Rank}(A_7)-\mathrm{Rank}(A_5)-\mathrm{Rank}(\Omega)>\mathsf{gap}, \tag{62}$$

where (56) to (62) are strengthened versions of the original inequalities in (37) to (41). The only difference is that (56) to (62) require[12] some rank gap, denoted by gap, between the left-hand side and the right-hand side of (37) to (41).

The reason why RLNC works whenever (56) to (62) hold is as follows. Take coding type-23 for example. From the definition of TYPE_{23} in (35) and from (49), the number of coding choices of TYPE_{23} in the beginning of time t satisfies

$$(q^{\mathsf{term1}}-q^{\mathsf{term2}}-q^{\mathsf{term3}})\leq|\mathsf{TYPE}_{23}|\leq q^{\mathsf{term1}} \tag{63}$$

where all three terms $\mathsf{term1}\triangleq\mathrm{Rank}(A_3\cap A_5)$, $\mathsf{term2}\triangleq\mathrm{Rank}(A_3\cap A_5\cap A_1)$, and $\mathsf{term3}\triangleq\mathrm{Rank}(A_3\cap A_5\cap A_4)$ are evaluated in the beginning of time t. Suppose we choose a coding vector $\mathbf{c}_{t,m}$ for the m-th input $W^{[m]}(t)$ in the beginning of time t. In the beginning of the split instant $t.m$, the linear sub-spaces A_1 to A_7 may have already evolved due to the reception status of the previous $(m-1)$ inputs $W^{[1]}$ to $W^{[m-1]}$. Therefore, some of the $|\mathsf{TYPE}_{23}|$ coding choices are no longer of type-23. However, one can lower bound the number of coding choices that remain of type-23 by

$$|\mathsf{TYPE}_{23}^{\ddagger}|\geq q^{\mathsf{term1}}-q^{\mathsf{term2}^{\dagger}}-q^{\mathsf{term3}^{\dagger}}$$

[12]One can further reduce the rank gap requirement by analyzing the detailed evolution of the linear sub-spaces. However, for the purpose of proving the achievability results, the (relatively loose) rank gap requirements specified in (55) to (62) are sufficient.

where $|\mathsf{TYPE}_{23}^{\ddagger}|$ is the number of original type-23 coding vectors (when evaluated in the beginning of time t) that are still of type-23 in the beginning of split time instant $t.m$; $\mathsf{term1}$ is identical to the one defined in (63); $\mathsf{term2}^{\dagger} \triangleq \mathrm{Rank}(A_3 \cap A_5 \cap A_1)$ and $\mathsf{term3}^{\dagger} \triangleq \mathrm{Rank}(A_3 \cap A_5 \cap A_4)$ are now evaluated in the beginning of the split time instant $t.m$. As a result, the probability that a randomly chosen type-23 coding vector $\mathbf{c}_{t,m}$ in the beginning of time t remains of type-23 for the split time instant $t.m$ is lower bounded by

$$\frac{|\mathsf{TYPE}_{23}^{\ddagger}|}{|\mathsf{TYPE}_{23}|} \geq \frac{q^{\mathsf{term1}} - q^{\mathsf{term2}^{\dagger}} - q^{\mathsf{term3}^{\dagger}}}{q^{\mathsf{term1}}}.$$

Hence, as long as $\mathsf{term1} > \max(\mathsf{term2}^{\dagger}, \mathsf{term3}^{\dagger})$, then for sufficiently large q, it is guaranteed that with close-to-one probability the coding type chosen in the beginning of time t is indeed the actual coding type for the split time instant $t.m$.

In the following, we prove that $\mathsf{term1} > \mathsf{term2}^{\dagger}$ if (55) to (62) are satisfied in the beginning of time t. The proof of $\mathsf{term1} > \mathsf{term3}^{\dagger}$ follows similarly.

By (51), we have

$$\begin{aligned}&\mathsf{term1} - \mathsf{term2}\\ &= \mathrm{Rank}(A_3) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_1) - \mathrm{Rank}(A_6)\end{aligned}$$

where A_3, A_5, A_1, and A_6 are evaluated in the beginning of time t. By (60), we thus have $(\mathsf{term1} - \mathsf{term2}) > 7(M-1)+1$. By Lemma 10 in Appendix H, we have $0 \leq \mathsf{term2}^{\dagger} - \mathsf{term2} \leq 3(M-1)$. Jointly, we thus have $(\mathsf{term1} - \mathsf{term2}^{\dagger}) > 4(M-1)+1$. The proof of $\mathsf{term1} > \mathsf{term2}^{\dagger}$ is complete.

We can repeat the above proof of the case of TYPE_{23} for all other 17 different $\mathsf{TYPE}_{\mathbf{b}}$. As a result, we have proven that RLNC circumvents the potential knowledge-space-mismatch problem as long as (55) to (62) hold in the beginning of the given time slot t.

Note that the above statement is a sufficient condition at time t, under which we can use RLNC to circumvent the need of having a genie updating the knowledge spaces for each split time instant $t.m$. The final step of completing the proof is to see whether we can satisfy (55) to (62) for all $t = 1$ to n. Unfortunately, even though it is possible to satisfy (55) to (62) for some $t \in \{1, \cdots, n\}$, it is impossible to satisfy (55) to (62) for all $t = 1$ to n. On the other hand, if we revisit the tunneling scheme described in Sections V-D and V-E, we can see that the key to the success of the tunneling approach is to ensure that for a majority of the overall duration $n - \mathcal{O}(\epsilon n)$, the trajectory of the normalized relative frequency $\{X_{\mathbf{b}}^{[m]}(t) : \forall \mathbf{b}, m\}$ is kept within the proper tunnel connecting $\vec{x}_{\text{start}}$ and $\vec{x}_{\text{end}}$. Since a proper tunnel is within the interior of the polytope Γ, it means that the nature of the tunneling approach automatically ensures that the gaps between the left-hand side and right-hand side of the rank comparison equalities (23) to (27) are strictly positive for the trajectory of the *normalized relative frequency*. This in turn implies that the gaps for the raw, unnormalized rank inequalities (37) to (41) are linearly proportional to n. Since the minimum gap requirement $\mathsf{gap} = 7(M-1)+1$ for (56) to (62) is independent of the n value, the nature of the tunneling approach ensures that we can use RLNC to circumvent the knowledge-space-mismatch problem during the entire tunneling phase (Phase 2). We also note that it is not possible to satisfy (56) to (62) for every time instant of Phases 1 and 3. Therefore, the throughput during Phases 1 and 3 still suffers from the knowledge-space mismatch problem. On the other hand, any performance degradation during Phases 1 and 3 is negligible from a long-term throughput perspective. We have thus proven that the combination of RLNC and the tunneling approach asymptotically achieves any (R_1, R_2) in the interior of the capacity region described in Proposition 1. The achievability proof for the case of general M values is complete when assuming sufficiently large $\mathsf{GF}(q)$.

C. *Code Design for* $\mathsf{GF}(q)$ *with Small* q

Even for small $\mathsf{GF}(q)$, we can still achieve the same effects of RLNC with proper network code design. To that end, we first observe that the main feature of the RLNC-based solution is that we first upper bound the largest possible "rank mismatch" due to the outdated feedback. Then as long as the original "rank discrepancy" in the beginning of time $t.1$, see (56) to (62), is larger than the potential rank mismatch through $t.1$ to $t.M$, we can blindly (randomly) choose the coding vector from a given type $\mathbf{b}$. The use of large $\mathsf{GF}(q)$ will ensure that the coding vector will remain in the same $\mathsf{TYPE}_{\mathbf{b}}$ with high probability even with the knowledge space mismatch. The main idea when encoding over small $\mathsf{GF}(q)$ is that the encoder can no longer choose the coding vectors randomly. Instead, the encoder has to *anticipate* the possible changes of the knowledge spaces and perform encoding based on the *worst case scenario*. We again use type-23 to demonstrate this difference.

Suppose we would like to send a type-23 coding vector $\mathbf{c}_{t,m}$ over the m-th input at time t. For $k = 1$ to 7, we use A_k to denote the linear subspace evaluated by the outdated knowledge spaces at time $(t-1).M$ and use $A_k^{\dagger}$ to denote the linear subspace evaluated by the latest knowledge spaces at time $t.(m-1)$. By (35), our goal is to choose a $\mathbf{c}_{t,m}$ in the beginning of time $t.1$ from the following set

$$\begin{aligned}\mathsf{TYPE}_{23}^{\dagger} &= (A_3^{\dagger} \cap A_5^{\dagger}) \backslash (A_1^{\dagger} \cup A_4^{\dagger})\\ &= (A_3^{\dagger} \cap A_5^{\dagger}) \backslash ((A_1^{\dagger} \cap A_3^{\dagger} \cap A_5^{\dagger}) \cup (A_3^{\dagger} \cap A_4^{\dagger} \cap A_5^{\dagger})).\end{aligned}$$

However, $\mathsf{TYPE}_{23}^{\dagger}$ is unknown in the beginning of time $t.1$. To circumvent this challenge, we construct a $\mathsf{TYPE}_{23}^{\ddagger}$, which is guaranteed to be a subset of $\mathsf{TYPE}_{23}^{\dagger}$ regardless of the reception status of the split time instants $t.1$ to $t.(m-1)$. As long as $\mathsf{TYPE}_{23}^{\ddagger}$ can be explicitly computed in the beginning of time $t.1$ and is non-empty, we can choose $\mathbf{c}_{t,m}$ from $\mathsf{TYPE}_{23}^{\ddagger}$ and $\mathbf{c}_{t,m}$ will always be of type-23 regardless of the reception status of the split time instants $t.1$ to $t.(m-1)$.

The set $\mathsf{TYPE}_{23}^{\ddagger}$ is constructed as follows.

$$\mathsf{TYPE}_{23}^{\ddagger} \triangleq (A_3 \cap A_5) \backslash ((A_1^{\ddagger} \cap A_3^{\ddagger} \cap A_5^{\ddagger}) \cup (A_3^{\ddagger} \cap A_4^{\ddagger} \cap A_5^{\ddagger})), \tag{64}$$

where A_k, by definition, is based on the outdated information and is thus guaranteed to a subspace of $A_k^{\dagger}$ for all k. We construct $A_k^{\ddagger}$ by (i) assuming *all* $\mathbf{c}_{t,1}$ to $\mathbf{c}_{t,m-1}$ have been received successfully by *both* d_1 and d_2; (ii) Computing the *anticipated* knowledge spaces $S_1^\ddagger$ and $S_2^\ddagger$ based on this overly optimistic assumption; and (iii) Computing $A_k^\ddagger$ based on $S_1^\ddagger$ and $S_2^\ddagger$. Since we optimistically assume that all $\mathbf{c}_{t,1}$ to $\mathbf{c}_{t,m-1}$ have been received successfully by both d_1 and d_2, we must have $S_i^\ddagger \supseteq S_i^\dagger$, where $S_i^\dagger$ is the actual knowledge space in the beginning of time $t.m$. By (31) to (33), we have $A_k^\ddagger \supseteq A_k^\dagger$ for $k = 1$ to 7. By the above arguments, $\mathsf{TYPE}_{23}^\ddagger$ in (64) can be computed in the beginning of time $t.1$ and is guaranteed to be a subset of $\mathsf{TYPE}_{23}^\dagger$.

To complete the proof, we need to prove that $\mathsf{TYPE}_{23}^\ddagger$ is non-empty. This can be done by assuming (56) to (62) are satisfied and by quantifying the maximum rank increase in a similar way as in the RLNC-based analysis, also see Lemma 10 in Appendix H. The above construction shows that we can send a type-23 coding vector $\mathbf{c}_{t,m}$ over the m-th input at time t even when we do not know the knowledge spaces in the beginning of the split time instant $t.m$. The achievability proof for the case of general M values is thus complete for any $\mathsf{GF}(q)$ including the binary field $\mathsf{GF}(2)$.

An illustrative example of how the tunneling approach works for the setting of general M values is provided in the next section.

Astute readers may notice that the previous $M = 1$ construction in Section V-D requires $\mathsf{GF}(q)$ with $q \geq 3$ but the proof for the general M in this subsection holds for any $\mathsf{GF}(q)$ with $q \geq 2$. The reason is as follows. If we plug in $M = 1$ to the strengthened rank comparison inequalities (55) to (62), we will not only have the inequalities (37) to (41) being strict, but the "rank discrepancy" is at least 2. Compared to the discussion in Section V-D that only assumes that the rank discrepancy is strictly positive (at least 1), this additional rank discrepancy requirement allows us to further sharpen the connection between the "rank-based comparison" and the "linear space size comparison" for binary field $\mathsf{GF}(2)$.

Take TYPE_3 for example. In Appendix G, it has been shown that $\mathsf{TYPE}_3 \neq \emptyset$ if

$$q^{\mathrm{Rank}(A_6 \cap A_7)} > q^{\mathrm{Rank}(A_3 \cap A_7)} + q^{\mathrm{Rank}(A_4 \cap A_6)} + q^{\mathrm{Rank}(A_5)} - 2, \quad (65)$$

and the following equalities are also proven:

$$\mathrm{Rank}(A_6 \cap A_7) - \mathrm{Rank}(A_3 \cap A_7) = \mathrm{Rank}(A_6) - \mathrm{Rank}(A_3) \quad (66)$$

$$\mathrm{Rank}(A_6 \cap A_7) - \mathrm{Rank}(A_4 \cap A_6) = \mathrm{Rank}(A_7) - \mathrm{Rank}(A_4) \quad (67)$$

$$\mathrm{Rank}(A_6 \cap A_7) - \mathrm{Rank}(A_5) = \mathrm{Rank}(A_6) + \mathrm{Rank}(A_7) - \mathrm{Rank}(\Omega) - \mathrm{Rank}(A_5). \quad (68)$$

Therefore, if $q \geq 3$ and all the above rate discrepancy (66)–(68) are strictly positive (at least 1), then (65) holds. Here we observe that if $q \geq 2$ and all the above rate discrepancy (66)–(68) are at least 2, then (65) holds again. The use of a larger gap in (56)–(62) allows us to generalize the previous tunneling-based construction to $\mathsf{GF}(2)$.

D. Reducing From 18M Types to 7M Types

Lemma 4 implies that when computing for the capacity-achieving target frequency $\{\ddot{x}_{\mathbf{b}}^{[1]} : \forall \mathbf{b}\}$, $11M$ out of the $18M$ coordinates can be hardwired to 0 and only $7M$ of them needs to be carefully chosen so that they satisfy the LP problem in Proposition 1. This observation not only reduces the complexity of computing the terminal point $\vec{\ddot{x}}$ of the tunneling approach, it also means that to achieve the capacity we only need to use $7M$ coding types. Lemma 4 can thus be interpreted as statement that *coding types* $\{0, 9, 18, 27, 31, 63, 95\}$ *jointly dominate all feasible coding types* $\forall \mathbf{b} \in \mathsf{FTs}$ *for any given underlying channel parameters.*

VII. Examples and Other Implications of Proposition 1

We first provide an illustrative example about the tunneling approach for general M-input 2-receiver broadcast PECs. We later discuss the implications of Proposition 1 on broadcast PEC with observable/controllable channel states.

A. An Illustrative Example — A Macroscopic View

Consider the same 2-input, 2-receiver, cross-input independent, broadcast PEC setting as was first discussed in Section III-C. That is, the per-input reception probabilities satisfy (13). One can verify that for this setting the rate vector $(R_1, R_2) = (0.875, 0.96)$ is strictly outside the improved simple inner bound in Lemma 3. At the same time, one can also verify that $(R_1, R_2) = (0.875, 0.96)$ is within the capacity region described in Proposition 1 by choosing

$$x_{31}^{[1]} = 0.125,\ x_{27}^{[1]} = 0.21,\ x_{18}^{[1]} = 0.665,\ x_9^{[2]} = 0.78125,\ x_0^{[2]} = 0.21875, \text{ and all other } x_{\mathbf{b}}^{[m]} \text{ values being } 0. \quad (69)$$

The above assignment of $\{x_{\mathbf{b}}^{[m]} : \forall \mathbf{b}, m\}$ shows that to achieve the rates $(R_1, R_2) = (0.875, 0.96)$, we only need to choose coding vectors of 5 different types $\mathbf{b} = 31, 27, 18, 9$, and 0. Furthermore, coding types $\mathbf{b} = 31, 27$, and 18 should be sent exclusively along the first input $W^{[1]}$ and coding types $\mathbf{b} = 9$ and 0 should be sent exclusively along the second input $W^{[2]}$. For illustration purposes, let us ignore the initial and the final phases (Phases 1 and 3) of the tunneling approach and focus on Phase 2, the main body of our network code construction.

When implementing the tunneling approach, in the beginning of every time t we first update the knowledge spaces S_1 and S_2 based on what d_1 and d_2 have heard in the past. Then we select one coding type $\mathbf{b}_1$ from $\{31, 27, 18\}$ and one coding type $\mathbf{b}_2$ from $\{9, 0\}$; choose a coding vector $\mathbf{c}_1$ uniformly randomly from $\mathsf{TYPE}_{\mathbf{b}_1}$ and choose a coding vector $\mathbf{c}_2$ uniformly randomly from $\mathsf{TYPE}_{\mathbf{b}_2}$. Then two symbols are generated by $\mathbf{W}^{[1]} = \overline{X} \cdot \mathbf{c}_1^{\mathrm{T}}$ and $\mathbf{W}^{[2]} = \overline{X} \cdot \mathbf{c}_2^{\mathrm{T}}$ and sent out through the two inputs of the broadcast PEC, respectively. The selection of $\mathbf{b}_1$ and $\mathbf{b}_2$ can be fully correlated or fully uncorrelated. Any coding type selection mechanism can achieve the same target rate $(R_1, R_2) = (0.875, 0.96)$ as long as the relative frequency of the coding types stays closely to the predetermined assignment in (69), i.e., $\{X_{\mathbf{b}}^{[m]}(t) : \forall \mathbf{b}, m\}$ stays within the tunnel for all t. In Section V-E, a zig-zag way of selecting the coding types are introduced, which facilitates analysis. For practical applications, for any $m \in \{1, \cdots, M\}$ one can simply choose $\mathbf{b}_m$ randomly and independently using the predetermined variable values $x_{\mathbf{b}}^{[m]}$ as the probability mass function. The law of large numbers will ensure that $\{X_{\mathbf{b}}^{[m]}(t) : \forall \mathbf{b}, m\}$ stays within the tunnel and the long-term frequency in the end matches the predetermined variable values $x_{\mathbf{b}}^{[m]}$.

B. An Illustrative Example — A Microscopic View

The previous subsection focuses on the high-level macroscopic perspective[13] of the new solution. For comparison, this subsection focuses on the per-packet behavior of the new framework.

We continue the example in Section VII-A. In addition to $M = 2$ and the channel statistics specified previously, we further assume that $nR_1 = 2$ and $nR_2 = 2$. That is, we have two symbols X_1 and X_2 for d_1 and two symbols Y_1 and Y_2 for d_2. The overall message space is $\Omega = (\mathsf{GF}(q))^4$ and the individual message spaces are $\Omega_1 = \mathsf{span}\{(1,0,0,0),(0,1,0,0)\}$ and $\Omega_2 = \mathsf{span}\{(0,0,1,0),(0,0,0,1)\}$.

Consider the beginning of time 1. Since neither d_1 nor d_2 has received anything in the past, we have $S_1 = \{(0,0,0,0)\} = S_2$. From the assignment $x_{\mathbf{b}}^{[m]}$ in (69), we know that we can send a coding vector of type-18 (type-0010010) for $W^{[1]}$. One can verify that $\mathbf{c}_1 = (1,0,0,0)$ is of type-18 and thus can be used for $W^{[1]}$. Since $\overline{\mathbf{X}}\mathbf{c}_1^{\mathrm{T}} = X_1$, sending this type-18 coding vector $\mathbf{c}_1$ is equivalent to sending the session-1 packet X_1 uncodedly. Suppose $W^{[1]} = X_1$ is received by d_2 but not by d_1. In the end of the split time instant[14] 1.1, the knowledge space of d_2 becomes $S_2 = \mathsf{span}\{(1,0,0,0)\}$. From the assignment $x_{\mathbf{b}}^{[m]}$ in (69), we know that we can send a coding vector of type-9 (type-0001001) for $W^{[2]}$. One can verify that $\mathbf{c}_2 = (0,0,1,0)$ is of type-9 and thus can be used for $W^{[2]}$. Since $\overline{\mathbf{X}}\mathbf{c}_2^{\mathrm{T}} = Y_1$, sending this type-9 coding vector $\mathbf{c}_2$ is equivalent to sending the session-2 packet Y_1 uncodedly. Suppose $W^{[2]} = Y_1$ is received by d_1 but not by d_2. In the end of the split time instant 1.2, the knowledge space of d_1 becomes $S_1 = \mathsf{span}\{(0,0,1,0)\}$.

We now consider the beginning of time 2. From the assignment $x_{\mathbf{b}}^{[m]}$ in (69), we know that we can send a coding vector of type-31 (type-0011111) for $W^{[1]}$. Given $S_1 = \mathsf{span}\{(0,0,1,0)\}$ and $S_2 = \mathsf{span}\{(1,0,0,0)\}$, one such choice of type-31 coding vectors is $\mathbf{c}_1 = (1,0,1,0)$. Since $\overline{\mathbf{X}}\mathbf{c}_1^{\mathrm{T}} = X_1 + Y_1$, sending this type-31 coding vector $\mathbf{c}_1$ is equivalent to sending the linear sum of X_1 and Y_1.

From the above observation, one can see that the existing LNC solution [7], [10] that sends (i) uncoded session-1 packets; (ii) uncoded session-2 packets; and (iii) linear combination of the overheard packets is equivalent to sending (i) type-18 coding vectors; (ii) type-9 coding vectors; and (iii) type-31 coding vectors, respectively.

[13]This is sometimes termed the flow perspective of network coding.

[14]We use the genie-aided transmission model herein so that we do not need to deal with the coding-type-mismatch problem elaborated in Section VI-B when discussing this illustrative example.

The reason why the achievability scheme in Proposition 1 strictly outperforms the achievability scheme in Lemma 3 in this example is due to the fact that the optimal solution in (69) is allowed to use two other coding types TYPE_{27} and TYPE_0, which are previously not considered in the existing results [6], [7], [10], [15].

To see why coding types TYPE_{27} and TYPE_0 can increase the throughput, we continue our discussion about this illustrative example. Suppose that for $t = 2$, we choose a type-31 coding vector $\mathbf{c}_1 = (1,0,1,0)$ for $W^{[1]}$. This vector serves both destinations d_1 and d_2 simultaneously. Also suppose that $\mathbf{c}_1 = (1,0,1,0)$ is received by d_1 but not by d_2. d_1 can decode the desired X_1 packet by subtracting the overheard Y_1 from the linear sum $[X_1 + Y_1]$. On the other hand, it also means that after the split time instant 2.1, there is no other way of sending a linear combination of packets that can benefit both d_1 and d_2 simultaneously since the "coding opportunity X_1" has been used up after d_1 hears $[X_1 + Y_1]$ and decodes X_1. In the following, we will see how the situation can change if we allow the use of type-27 and type-0 coding vectors in addition to the existing types 18, 9, and 31.

Let us term the above network code construction code1 and we will design a new network code code2 from scratch. The difference between them is that code1 uses only types 18, 9, and 31, while code2 will use other coding types TYPE_{27} and TYPE_0 as well. For illustration, In time $t = 1$, code2 chooses to send $\mathbf{c}_1 = (1,0,0,0)$, a type-18 coding vector for $W^{[1]}$. Also we assume $W^{[1]} = X_1$ is received by d_2 but not by d_1. In the end of the split time instant 1.1, the knowledge space of d_2 becomes $S_2 = \mathsf{span}\{(1,0,0,0)\}$. Thus far the coding choice of code2 is identical to that of code1. We use Table II to summarize the comparison between code1 and code2.

Suppose for the split time instant 1.2, code2 chooses to send a coding vector $\mathbf{c}_2$ of type-0 (type-0000000) for $W^{[2]}$. One can verify that $\mathbf{c}_2 = (1,1,1,1)$ is of type-0 and the second input is thus $W^{[2]} = X_1 + X_2 + Y_1 + Y_2$. Similar to the discussion of code1, we assume that $W^{[2]} = X_1 + X_2 + Y_1 + Y_2$ is received by d_1 but not by d_2. In the end of the split time instant 1.2, the knowledge space of d_1 becomes $S_1 = \mathsf{span}\{(1,1,1,1)\}$. Note that thus far there is no significant difference between code1 and code2 even though code2 substitutes the type-9 coding vector by a type-0 coding vector for $W^{[2]}$. Specifically, in the end of time 1, neither d_1 nor d_2 has received any desired packets, and both d_1 and d_2 have overheard one packet that could potentially benefit the future coding operations. See Table II for the summary of the split time instants 1.1 and 1.2.

We now consider the beginning of time 2. Suppose that code2 chooses to send a coding vector of type-27 (type-0011011) for $W^{[1]}$. Given $S_1 = \mathsf{span}\{(1,1,1,1)\}$ and $S_2 = \mathsf{span}\{(1,0,0,0)\}$, one such choice of type-27 coding vectors is $\mathbf{c}_1 = (1,0,1,1)$. We now argue that code2 sending $\overline{\mathbf{X}}\mathbf{c}_1^{\mathrm{T}} = X_1 + Y_1 + Y_2$ can also benefit two destinations simultaneously in a similar way as code1 sending $[X_1 + Y_1]$. Specifically, since d_1 has heard $[X_1 + X_2 + Y_1 + Y_2]$ in the split time instant 1.2, upon the reception of $[X_1 + Y_1 + Y_2]$, d_1 can decode the desired packet X_2. Since d_2 has received X_1 in the split time instant 1.1, upon the reception of $[X_1 + Y_1 + Y_2]$, d_2 can decode $[Y_1 + Y_2]$, which is a linear combination of the desired

TABLE II
COMPARISON OF THE PER-PACKET BEHAVIOR OF code1 AND code2.

code1	Split time instant 1.1		Split time instant 2.1	
Input 1	Choose type-18 (type-0010010); $\mathbf{c}_1 = (1,0,0,0)$; $W^{[1]} = X_1$.	Received by d_2 only; $S_1 = \{(0,0,0,0)\}$; $S_2 = \mathsf{span}\{(1,0,0,0)\}$; Decoded pkts: $d_1 : \emptyset$, $d_2 : \emptyset$.	Choose type-31 (type-0011111); $\mathbf{c}_1 = (1,0,1,0)$; $W^{[1]} = X_1 + Y_1$; (Benefit both d_1 and d_2).	Received by d_1 only; $S_1 = \mathsf{span}\{(0,0,1,0),(1,0,1,0)\}$; $S_2 = \mathsf{span}\{(1,0,0,0)\}$; Decoded pkts: $d_1 : \{X_1\}$, $d_2 : \emptyset$.
	Split time instant 1.2		Split time instant 2.2	
Input 2	Choose type-9 (type-0001001); $\mathbf{c}_2 = (0,0,1,0)$; $W^{[2]} = Y_1$.	Received by d_1 only; $S_1 = \mathsf{span}\{(0,0,1,0)\}$; $S_2 = \mathsf{span}\{(1,0,0,0)\}$; Decoded pkts: $d_1 : \emptyset$, $d_2 : \emptyset$.	No choice of $\mathbf{c}_2$ can benefit both d_1 and d_2 simultaneously.	

code2	Split time instant 1.1		Split time instant 2.1	
Input 1	Choose type-18 (type-0010010); $\mathbf{c}_1 = (1,0,0,0)$; $W^{[1]} = X_1$.	Received by d_2 only; $S_1 = \{(0,0,0,0)\}$; $S_2 = \mathsf{span}\{(1,0,0,0)\}$; Decoded pkts: $d_1 : \emptyset$, $d_2 : \emptyset$.	***Choose type-27 (type-0011011);*** $\mathbf{c}_1 = (1,0,1,1)$; $W^{[1]} = X_1 + Y_1 + Y_2$; (Benefit both d_1 and d_2).	Received by d_1 only; $S_1 = \mathsf{span}\{(1,1,1,1),(1,0,1,1)\}$; $S_2 = \mathsf{span}\{(1,0,0,0)\}$; Decoded pkts: $d_1 : \{X_2\}$, $d_2 : \emptyset$.
	Split time instant 1.2		Split time instant 2.2	
Input 2	***Choose type-0 (type-0000000);*** $\mathbf{c}_2 = (1,1,1,1)$; $W^{[2]} = X_1 + X_2 + Y_1 + Y_2$.	Received by d_1 only; $S_1 = \mathsf{span}\{(1,1,1,1)\}$; $S_2 = \mathsf{span}\{(1,0,0,0)\}$; Decoded pkts: $d_1 : \emptyset$, $d_2 : \emptyset$.	***Sending $\mathbf{c} = (0,0,1,1)$ can benefit both d_1 and d_2 simultaneously.***	

packets Y_1 and Y_2. Sending $[X_1+Y_1+Y_2]$ thus benefits both d_1 and d_2 simultaneously.

Assume that after sending $\mathbf{c}_1 = (1,0,1,1)$, d_1 successfully receives $\mathbf{c}_1$ but d_2 receives erasure. As discussed, d_1 will decode X_2 upon the reception of $\mathbf{c}_1$. From the above discussion, one can see that with the same channel realization for the the first three split time instants 1.1, 1.2, and 2.1, both code1 and code2 behave identically in the following sense. Two packets are overheard after the split time instants 1.1 and 1.2, respectively; a linear combination that can serve both destinations is sent out in the split time instant 2.1; and only d_1 receives the linear combination in time 2.1 and uses it to decode one of the desired X packets. See Table II for the summary of the split time instants 1.1, 1.2, and 2.1.

The difference between code1 and code2 becomes apparent in the beginning of the next split time instant. In particular, after the split time instant 2.1, code2 can find yet another linear combination that benefits both destinations simultaneously. Namely, if code2 chooses to send $W = Y_1 + Y_2$, then upon the reception of $W = Y_1 + Y_2$, d_1 can decode X_1 based on the overheard packets $[X_1 + X_2 + Y_1 + Y_2]$ and $[X_1+Y_1+Y_2]$ and d_2 can receive a linear combination of the desired packets Y_1 and Y_2. Sending $W = Y_1+Y_2$ thus benefits both d_1 and d_2 simultaneously. Note that this behavior of code2 is in sharp contrast with code1 for which no such linear combination exists after the split time instant 2.1, see Table II. This example of code2 thus demonstrates the potential benefits of incorporating types 0 and 27 in the network code design.

Astute readers may notice that when code2 sends the linear combination $\mathbf{c}_2 = (1,1,1,1)$ in time 1.2, it actually loses some (average) throughput when compared to code1 since unlike the uncoded $W^{[2]} = Y_1$ sent by code1, which could directly benefit d_2 if received by d_2, the coded packet $W^{[2]} = X_1 + X_2 + Y_1 + Y_2$ sent by code2 cannot directly benefit d_2. However, if the probability that d_2 receives $W^{[2]}$ is very small, then sending $W^{[2]} = X_1 + X_2 + Y_1 + Y_2$ has almost zero throughput loss for time 1.2 when compared to sending $W^{[2]} = Y_1$ since the scenario that $W^{[2]} = Y_1$ directly benefits d_2 (i.e., heard by d_2) rarely happens. Recall from our observation that after the split time instant 2.1, code2 can still send a linear combination that benefits both d_1 and d_2 while code1 cannot. Therefore, sending $W^{[2]} = X_1 + X_2 + Y_1 + Y_2$ at time 1.2 creates additional future coding opportunity that can strictly enhance the throughput.

The key implication is that when compared to sending $W^{[2]} = Y_1$ in time 1.2, the immediate throughput loss (at the current split time instant) and the future throughput gains (after time 2.1) of sending $W^{[2]} = X_1+X_2+Y_1+Y_2$ in time 1.2 are decided by the given channel characteristics, particularly the reception probability of $W^{[2]}$. Therefore, for some channel statistics, sending a type-0 vector $\mathbf{c}_2 = (1,1,1,1)$ can be more beneficial than sending a type-9 vector $\mathbf{c}_2 = (0,0,1,0)$ (sending the uncoded Y_1 packet) while for some other channel statistics, the preference is reversed. For the $M = 1$ case, one can prove that the throughput benefits of coding types 18, 9 and 31 dominate that of any other coding type (e.g., TYPE_0). As a result, one can always design a capacity achieving scheme without using any type-0 coding vectors [7]. However, for the case of $M > 1$, the additional diversity of the M different inputs breaks the dominance relationship and the benefits of using coding vectors of other types (e.g., TYPE_0) start to emerge under some channel characteristics.

It is worth noting that the central message of Proposition 1 is not that some coding types are better than the others. Instead, one main contribution of this work is to show that the benefits and limitations of each coding type can be decided by the underlying channel statistics and we can use an LP solver to dynamically balance the frequency of sending each coding type so that different coding types can complement each other and achieve holistically the optimal throughput performance. The optimality of our results follows from the fact that the proposed framework has exhaustively considered all possible coding types and all possible frequency assignment of the coding types.

C. LNC Capacity for Broadcast PECs with Partially Markovian and Partially Controllable Channel State and Delayed Channel Output Feedback

The methodology proposed in Proposition 1 can be readily extended to more general settings. In this subsection, we demonstrate how to use the results in this work to characterize the LNC capacity of broadcast PECs with instantly observable/controllable channel state and delayed channel output feedback.

Definition 7: Consider an ergodic finite-state Markov chain $\{S_t : \forall t = 1, 2, \cdots\}$ with the state space being $\mathcal{S}$ and a sequence of user controllable actions $\{\mathsf{Act}_t \in \mathcal{A} : \forall t = 1, 2, \cdots\}$ taken from a finite set $\mathcal{A}$. We say an M-input 2-receiver broadcast PEC is partially Markovian with respect to $\{S_t : \forall t\}$ and partially controllable with respect to $\{\mathsf{Act}_t : \forall t\}$ if for any given time instant τ, the distribution of the random reception status rx_1 and rx_2 depends only on the values of S_τ and Act_τ, but not on the value of τ nor on the values of S_t, Act_t, and the reception status for all other time instants $t \neq \tau$. The above partially Markovian and partially controllable M-input 2-receiver broadcast PEC can be described by the reception probability $\{p_{\mathsf{rx}_1,\mathsf{rx}_2|s,\mathsf{a}} : \forall \mathsf{rx}_1, \mathsf{rx}_2, \forall s \in \mathcal{S}, \forall \mathsf{a} \in \mathcal{A}\}$ conditioning on the channel state being $S_t = s$ and the user controllable action being $\mathsf{Act}_t = \mathsf{a}$. We can also define cross-input and cross-receiver independence in a similar way as in Definitions 1 and 2 except that the independence is now defined over the conditional distribution $p_{\mathsf{rx}_1,\mathsf{rx}_2|s,\mathsf{a}}$.

We are interested in finding the largest (R_1, R_2) such that an (optimal) LNC scheme can send nR_1 and nR_2 independent packets to d_1 and d_2, respectively, within the total time budget n. That is, in the beginning of each time t, source s is allowed to choose a specific action Act_t based on the reception status in the past and on the Markovian channel state in the past and the present. That is,

$$\mathsf{Act}_t = f_{t,\mathsf{Act}}([\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}, [S_\tau]_{\tau=1}^t).$$

Then s sends an M-dimensional row vector $\mathbf{W}(t)$ by

$$\mathbf{W}(t) = \overline{\mathbf{X}} \cdot \mathbf{C}_t$$

where the $(nR_1 + nR_2) \times M$ encoding matrix $\mathbf{C}_t$ is chosen based on all the past reception status $[\mathsf{rx}_1, \mathsf{rx}_2]_1^{t-1}$ and on the Markovian channel state in the past and the present $[S_\tau]_{\tau=1}^t$. Each destination d_i decodes its desired packets $\mathbf{X}_i$ in the end of time n,

$$\hat{\mathbf{X}}_i = g_i([\mathbf{Z}_i]_1^n, [\mathbf{C}_t]_{t=1}^n)$$

based on what it has received in the past, $[\mathbf{Z}_i]_1^n$, and on knowing how each coded packet is generated.

We can then define the LNC capacity region of the M-input 2-receiver broadcast PECs with instantly observable/controllable channel state and delayed channel output feedback in a similar way as in Definitions 3 and 4.

For notational simplicity, we use $p^{[m]}_{a_1a_2|s,\mathsf{a}}$ to denote the reception probabilities for the m-th symbol $W^{[m]}$ given the channel state being s and the user action being a, where each bit a_i being 1 or 0 indicates whether d_i receives $W^{[m]}$ or not, respectively.

Remark: This partially Markovian and partially controllable M-input 2-receiver broadcast PEC is a strict generalization of the setting in Section II and it closely captures many important practical applications. For example, consider a setting of cognitive radio. When an external source is transmitting, the packet erasure probability is higher due to the presence of stronger interference. When the external source is silent, the packet erasure probability is lower due to better channel quality. The activity of the external source can now be modeled by the Markovian state S_t. Another example is the classic Gilbert-Elliott channel model for burst erasures, which is a special example of the above Markovian setting.

The user controllable action Act_t also provides another important degree of freedom when optimizing the system throughput. For example, source s may choose different coding and modulation schemes for each time t. Suppose two choices are possible, i.e., $\mathcal{A} = \{1, 2\}$. The first coding and modulation scheme (when choosing $\mathsf{Act}_t = 1$) has lower transmission rate (bit per second) but higher probability of arriving at the destinations successfully while the second coding and modulation scheme (when choosing $\mathsf{Act}_t = 2$) has higher transmission rate but lower probability of arriving at the destinations successfully. To simplify our discussion of this example, assume that the channel state S_t is always constant. The above scenario can now be modeled by choosing $M = 2$ and setting

$$p^{[1]}_{00|\mathsf{a}=1} = 0.04,\ p^{[1]}_{10|\mathsf{a}=1} = p^{[1]}_{01|\mathsf{a}=1} = 0.16,\ p^{[1]}_{11|\mathsf{a}=1} = 0.64;$$
$$p^{[2]}_{00|\mathsf{a}=1} = 1, \quad p^{[2]}_{01|\mathsf{a}=1} = p^{[2]}_{10|\mathsf{a}=1} = p^{[2]}_{11|\mathsf{a}=1} = 0;$$
$$p^{[1]}_{00|\mathsf{a}=2} = p^{[1]}_{01|\mathsf{a}=2} = p^{[1]}_{10|\mathsf{a}=2} = p^{[1]}_{11|\mathsf{a}=2} = 0.25;$$
$$p^{[2]}_{00|\mathsf{a}=2} = p^{[2]}_{01|\mathsf{a}=2} = p^{[2]}_{10|\mathsf{a}=2} = p^{[2]}_{11|\mathsf{a}=2} = 0.25;$$

where for notational simplicity we omit the subscript that conditions on the constant channel state S_t. The above assignment of $p^{[m]}_{a_1a_2|\mathsf{a}}$ implies that when $\mathsf{Act}_t = 1$, one can transmit only one packet through $W^{[1]}$ since any packet sent through $W^{[2]}$ is always erased $p^{[2]}_{00|\mathsf{a}=1} = 1$. This models the lower transmission rate of $\mathsf{Act}_t = 1$. Conditioning on $\mathsf{Act}_t = 1$, the probability that each destination successfully receives $W^{[1]}$ is 0.8. On the other hand, when $\mathsf{Act}_t = 2$ one can transmit two packets through $W^{[1]}$ and $W^{[2]}$ simultaneously, which models the higher transmission rate of $\mathsf{Act}_t = 2$. On the other hand, the probability of each destination successfully receives $W^{[i]}$ is 0.5 when $\mathsf{Act}_t = 2$, which is lower than the case when $\mathsf{Act}_t = 1$. The above assignment of $p^{[m]}_{a_1a_2|\mathsf{a}}$ thus models the throughput and reliability tradeoff when we are allowed to dynamically change the coding and modulation schemes. The user controllable action Act_t also has the flavor of "network scheduling," which is commonly considered in the networking society.

The LNC capacity of the partially Markovian and partially controllable broadcast PEC is described as follows.

Proposition 3: For any fixed $\mathsf{GF}(q)$, consider any cross-input independent, partially Markovian and partially controllable M-input 2-receiver broadcast PEC. A rate vector

(R_1, R_2) is in the LNC capacity region if and only if there exist $|\mathcal{S}| \cdot |\mathcal{A}|$ non-negative variables $t_{s,\mathsf{a}}$ for all $s \in \mathcal{S}$ and $\mathsf{a} \in \mathcal{A}$; $18M \cdot |\mathcal{S}| \cdot |\mathcal{A}|$ non-negative variables $x_{\mathbf{b}}^{[m,s,\mathsf{a}]}$ for all $\mathbf{b} \in \mathsf{FTs}$, $m \in \{1, \cdots, M\}$, $s \in \mathcal{S}$, and $\mathsf{a} \in \mathcal{A}$; and 7 non-negative variables y_1 to y_7 such that jointly they satisfy the following 4 groups of linear conditions:

• Group 1, termed the *time-sharing conditions*, has $|\mathcal{S}| + M \cdot |\mathcal{S}| \cdot |\mathcal{A}|$ equalities:

$$\forall s \in \mathcal{S}, \quad \sum_{\forall \mathsf{a} \in \mathcal{A}} t_{s,\mathsf{a}} = \pi_s$$

$$\forall m, \forall s \in \mathcal{S}, \forall \mathsf{a} \in \mathcal{A}, \quad \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs}} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \leq t_{s,\mathsf{a}}$$

where $\{\pi_s : \forall s \in \mathcal{S}\}$ is the steady-state distribution of the Markov chain S_t.

• Group 2, termed the *rank-conversion conditions*, has 7 equalities:

$$y_1 = \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_1 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \cdot \left(p_{10|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

$$y_2 = \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_2 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \cdot \left(p_{01|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

$$y_3 = R_1 + \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_3 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \left(p_{10|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

$$y_4 = R_2 + \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_4 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \left(p_{01|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

$$y_5 = \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_5 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \cdot \left(p_{10|s,\mathsf{a}}^{[m]} + p_{01|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

$$y_6 = R_1 + \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_6 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \cdot \left(p_{10|s,\mathsf{a}}^{[m]} + p_{01|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

$$y_7 = R_2 + \sum_{m,s,\mathsf{a}} \left(\sum_{\forall \mathbf{b} \in \mathsf{FTs} \text{ and } b_7 = 0} x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \right) \cdot \left(p_{10|s,\mathsf{a}}^{[m]} + p_{01|s,\mathsf{a}}^{[m]} + p_{11|s,\mathsf{a}}^{[m]} \right)$$

where the summation $\sum_{m,s,\mathsf{a}}$ is shorthand for $\sum_{\forall m, \forall s \in \mathcal{S}, \forall \mathsf{a} \in \mathcal{A}}$

• Group 3, termed the *rank-comparison conditions*, has 7 inequalities that are identical to (23) to (27).

• Group 4, termed the *decodability conditions*, has 2 equalities that are identical to (28).

Proof of Proposition 3: Most definitions of this proof have been defined in Sections V and VI.

The "Only If" direction: Fix any given linear network code such that d_i can decode all $X_{i,1}$ to X_{i,nR_i} in the end of time n for all $i = 1, 2$ with close-to-one probability. We can construct the $t_{s,\mathsf{a}}$, $x_{\mathbf{b}}^{[m,s,\mathsf{a}]}$, y_1 to y_7 variables as follows.

$$t_{s,\mathsf{a}} \triangleq \frac{1}{n} \mathsf{E} \left\{ \sum_{t=1}^{n} 1_{\{S_t = s, \mathsf{Act}_t = a\}} \right\} \tag{70}$$

$$x_{\mathbf{b}}^{[m,s,\mathsf{a}]} \triangleq \frac{1}{n} \mathsf{E} \left\{ \sum_{t=1}^{n} 1_{\{S_t = s, \mathsf{Act}_t = a, \mathbf{c}_{t,m} \text{ for } W^{[m]} \text{ is of type } \mathbf{b}.\}} \right\} \tag{71}$$

$$y_k \triangleq \frac{1}{n} \mathsf{E} \left\{ \text{Rank}(A_k(n)) \right\}. \tag{72}$$

For (71), the coding type $\mathbf{b}$ is evaluated in the beginning of the split time instant $t.m$.

We can then follow the same analysis as in Sections V and VI and prove that the variables assignment in (70) to (72) satisfy all four groups of conditions in Proposition 3.

The "If" direction: For any (R_1, R_2) in the interior of the LNC capacity region described in Proposition 3, consider the corresponding $t_{s,\mathsf{a}}$, $x_{\mathbf{b}}^{[m,s,\mathsf{a}]}$, and y_k values as fixed constants in the subsequent discussion. An achievability scheme can then be constructed based on the tunneling approach discussed in Sections V-D, V-E, and VI. That is, Phase 1 lasts for a negligible $\mathcal{O}(\epsilon n)$ number of time slots, which focuses on attaining the relative frequency $\vec{x}_{\text{start}}$ that is in the interior of the $(18M \cdot |\mathcal{S}| \cdot |\mathcal{A}|)$-dimensional polytope Γ and satisfies $d(\vec{\mathbf{0}}, \vec{x}_{\text{start}}) < \epsilon$. For Phase 2, the main body of the scheme, we will use the tunneling approach to ensure that the trajectory goes from $\vec{x}_{\text{start}}$ to $\vec{x}_{\text{end}}$ where $\vec{x}_{\text{end}}$ is in the interior of Γ and is very close to the target relative frequency $\{x_{\mathbf{b}}^{[m,s,\mathsf{a}]}\}$, i.e., $d(\vec{x}_{\text{end}}, \{x_{\mathbf{b}}^{[m,s,\mathsf{a}]}\}) < \epsilon$. Phase 3 lasts for a negligible $\mathcal{O}(\epsilon n)$ number of time slots, which focuses on attaining the target relative frequency $\{x_{\mathbf{b}}^{[m,s,\mathsf{a}]}\}$. The detailed implementation of Phases 1 and 3 follows almost identically to the discussion in Appendix F. We now focus on Phase 2, the main body of the tunneling approach.

For any time t in Phase 2, if the channel state $S_t = s$, then we choose the action Act_t independently randomly accordingly to the probability mass function $\left\{ \frac{t_{s,\mathsf{a}}}{\pi_s} : \forall \mathsf{a} \in \mathcal{A} \right\}$. Then for the m-th input, we choose a coding type $\mathbf{b}$ independently randomly accordingly to the probability mass function $\left\{ \frac{x_{\mathbf{b}}^{[m,s,\mathsf{a}]}}{t_{s,\mathsf{a}}} : \forall \mathbf{b} \in \mathsf{FTs} \right\}$. Once a coding type $\mathbf{b}$ is selected, we choose independently and uniformly randomly a coding vector $\mathbf{c}$ from $\mathsf{TYPE}_{\mathbf{b}}$ and use such $\mathbf{c}$ to encode the m-th input $W^{[m]}$. Such an approach guarantees that the long-term frequency of choosing coding type $\mathbf{b}$ for the m-th input when counting only the time instants with $S_t = s$ and $\mathsf{Act}_t = \mathsf{a}$ converges to the given assignment $x_{\mathbf{b}}^{[m,s,\mathsf{a}]}$. By the tunneling approach and the corresponding analysis in Sections V and VI, it is guaranteed that the proposed scheme achieves the target rates (R_1, R_2). ■

VIII. CONCLUSION

This work has characterized the full LNC capacity of the cross-input independent M-input 2-receiver broadcast PECs with channel output feedback. A new linear-space-based constructive approach has been proposed, which uses an LP solver to exhaustively search for the LNC scheme(s) with best possible throughput. The resulting LNC scheme is thus guaranteed to attain the LNC capacity. We have then used pure algebraic arguments to show that the LNC capacity matches a simple capacity outer bound and is thus the true capacity region. A byproduct of the above results is a complete LNC capacity region characterization for 2-receiver partially Markovian and partially controllable broadcast PECs.

One future direction is to generalize the results for an arbitrary number of receivers. The main challenge for the generalization is that the achievability results no longer hold since for the case of ≥ 3 receivers, whether a feasible coding type is empty or not cannot be determined solely by checking the rank inequalities as was the case of 2 receivers, see Table I. We believe an even deeper understanding of the linear network code structure is necessary for the case of ≥ 3 receivers.

APPENDIX A
PROOF OF LEMMA 3

We prove this lemma by explicit construction similar to the ideas in [7], [10].

The main idea of the optimal LNC schemes in [7], [10] is the following. Source s first sends each flow-1 packet X_k, $k = 1$ to nR_1, uncodedly until it is received by at least one of the users d_1 and d_2. Then source s sends each flow-2 packet Y_l, $l = 1$ to nR_2, uncodedly until it is received by at least one of d_1 and d_2. After finishing sending all X_k and Y_l uncodedly, we examine the reception status of each X_k and Y_l. If a X_k is received by the intended destination d_1, then no further transmission is necessary for such X_k. If not, it means that such X_k is received by d_2 but not by d_1, which is a good candidate for future LNC operations. Symmetrically, there is no need to send those Y_l that have been received by its intended destination d_2. All the other Y_l are received by d_1 but not by d_2, which are good candidates for future LNC operations. As a result, after sending X_k and Y_l uncodedly, source s simply sends linear sums of X_k and Y_l that combine those "overheard-by-the-other-destination" X_k and Y_l. The following scheme is based on the above simple LNC construction.

In the following, we first provide the so-called *first-order analysis* for the achievability of a linear network coding solution. After presenting the first-order analysis, we then provide details how to use the law of large numbers to derive a rigorous proof from the first-order analysis. One can see that although the application of the law of large numbers is straightforward, the corresponding (ϵ, δ)-language substantially lengthens the overall proof and makes the overall logic flow less intuitive. Since the achievability proofs in Sections V-D, V-E, and VI-B are quite long and involve many new concepts even when using the first-order analysis, for those proofs we omit the detailed application of the law of large numbers. The interested readers should be able to follow the spirit of the second half of this appendix section and derive rigorous proofs by themselves. Also see the discussion in the subsection Appendix A-C.

A. The First-Order Analysis

Given any $R_1, R_2, R_1^{[m]}, R_2^{[m]}$ variables satisfying (9) to (12), we first divide the nR_i packets among the M inputs by $nR_i^{[1]}$ to $nR_i^{[M]}$ for $i = 1, 2$, and send the $nR_i^{[m]}$ packets uncodedly through the m-th input until each one is received by at least one of d_1 and d_2. These two steps are always feasible since (9) and (10) are satisfied.

After sending packets uncodedly, we examine the reception status. Totally, the following number of flow-i packets have been received correctly by d_i

$$\sum_{m=1}^{M} nR_i^{[m]} \left(\frac{p_{11}^{[m]} + 1_{\{i=1\}} p_{10}^{[m]} + 1_{\{i=2\}} p_{01}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right)$$

and the remaining

$$\sum_{m=1}^{M} nR_i^{[m]} \left(\frac{1_{\{i=1\}} p_{01}^{[m]} + 1_{\{i=2\}} p_{10}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right). \tag{73}$$

packets have been heard by the other destination d_j, $j \neq i$, but not by d_i. (73) corresponds to the number of all the newly created *network coding opportunities* [7], [10], [12].

The next step is to redistribute the coding opportunities among all M inputs, and use the remaining time slots (of each of the M sub-channels) to send the packets that have not arrived at their intended destinations by LNC. To analyze this step, we notice that the m-th sub-channel has

$$n \left(1 - \frac{R_1^{[m]} + R_2^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right) \tag{74}$$

number of time slots left. As a result, we can finish transmission of all the remaining coding opportunities (73) if the following conditions are satisfied for all $i = 1, 2$.

$$(73) < \sum_{m=1}^{M} \left((74) \cdot \left(p_{11}^{[m]} + 1_{\{i=1\}} p_{10}^{[m]} + 1_{\{i=2\}} p_{01}^{[m]} \right) \right). \tag{75}$$

We then notice that (11) implies (75) for the case of $i = 1$ and (12) implies (75) for the case of $i = 2$. As a result, we can finish the remaining coding opportunities in (73) using the remaining time slots of each of the M sub-channels. The above scheme can thus send nR_i packets to d_i within the allotted n time slots. The proof of Lemma 3 is complete.

Note that all the above statements are made in the sense of expected values. That is why it is termed the first-order analysis.

B. Complete Analysis of The Law of Large Numbers

Given any $R_1, R_2, R_1^{[m]}, R_2^{[m]}$ variables satisfying (9) to (12), we first define the following $3M$ different constants $\bar{t}_{i,m}$ for $i = 1, 2$ and $\bar{t}_{\mathrm{C},m}$ for $m = 1$ to M, respectively.

$$\forall i = 1, 2, \quad \bar{t}_{i,m} \triangleq \frac{R_i^{[m]}}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}} (1 + \delta) \tag{76}$$

$$\text{and} \quad \bar{t}_{\mathrm{C},m} \triangleq 1 - \bar{t}_{1,m} - \bar{t}_{2,m} \tag{77}$$

where $\delta > 0$ is some strictly positive constant. We choose the value of $\delta > 0$ to be sufficiently small such that the following

$M+2$ inequalities are satisfied simultaneously.

$$\forall m = 1, \cdots, M, \quad \bar{t}_{\mathrm{C},m} > 0, \tag{78}$$

$$\sum_{m=1}^{M} R_1^{[m]} \left(\frac{p_{01}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right) (1+\delta) \leq \sum_{m=1}^{M} \bar{t}_{\mathrm{C},m} (p_{10}^{[m]} + p_{11}^{[m]}), \tag{79}$$

$$\text{and} \sum_{m=1}^{M} R_2^{[m]} \left(\frac{p_{10}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right) (1+\delta) \leq \sum_{m=1}^{M} \bar{t}_{\mathrm{C},m} (p_{01}^{[m]} + p_{11}^{[m]}). \tag{80}$$

We first claim that such a $\delta > 0$ can always be found. The reason is that (10) implies that we can find a small $\delta > 0$ such that (78) is satisfied. Further, (11) (resp. (12)) implies that we can find a small $\delta > 0$ such that (79) (resp. (80)) is satisfied.

Once the $\bar{t}_{i,m}$, $\bar{t}_{\mathrm{C},m}$, and δ values are decided, we are ready to describe the network code construction. For any given network code block length n, the network code contains three stages. Stage 1: Divide the nR_1 packets $X_{1,1}$ to X_{1,nR_1} among the M inputs by $nR_1^{[1]}$ to $nR_1^{[M]}$, and send the $nR_1^{[m]}$ packets uncodedly through the m-th input until each one is received by at least one of d_1 and d_2. We run Stage 1 on the m-th input for $n \cdot \bar{t}_{1,m}$ time slots. For any m, if all the $nR_1^{[m]}$ packets allocated to the m-th input are heard by at least one of $\{d_1, d_2\}$ before using up the time budget $n\bar{t}_{1,m}$, then we simply let the m-th input remain idle until we use up the allocated $n\bar{t}_{1,m}$ time slots. After using up the time budget $n\bar{t}_{1,m}$, we proceed to Stage 2. Before describing Stage 2, we define the following two types of *encoding errors*. Note that regardless whether there is any encoding error, we proceed to Stage 2 anyhow.

Encoding error type 1 for Stage 1: For at least one m value, there is still one packet, out of the $nR_1^{[m]}$ packets assigned to the m-th input, that has not been heard by any of $\{d_1, d_2\}$ in the end of Stage 1.

Encoding error type 2 for Stage 1: For at least one m value, out of the total number of $nR_1^{[m]}$ packets assigned to the m-th input, the number of packets that have been heard by the other destination d_2 but not by d_1 is larger than

$$nR_1^{[m]} \left(\frac{p_{01}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right) \left(1 + \frac{\delta}{2} \right). \tag{81}$$

Stage 2 is symmetric to Stage 1. That is, we divide the nR_2 packets $X_{2,1}$ to X_{2,nR_2} among the M inputs and send the $nR_2^{[m]}$ packets uncodedly through the m-th input until each one is received by at least one of d_1 and d_2. The allocated time budget for the m-th input is $n\bar{t}_{2,m}$. We also define the encoding error types 1 and 2 for Stage 2:

Encoding error type 1 for Stage 2: For at least one m value, there is still one packet, out of the $nR_2^{[m]}$ packets assigned to the m-th input, that has not been heard by any of $\{d_1, d_2\}$ in the end of Stage 2.

Encoding error type 2 for Stage 2: For at least one m value, out of the total number of $nR_2^{[m]}$ packets assigned to the m-th input, the number of packets that have been heard by the other destination d_1 but not by d_2 is larger than

$$nR_2^{[m]} \left(\frac{p_{10}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}} \right) \left(1 + \frac{\delta}{2} \right). \tag{82}$$

Regardless whether there is any encoding error, we proceed to Stage 3 anyhow.

Stage 3: We first notice that those $\mathbf{X}_i$ packets that have been heard by the other destination d_j, $j \neq i$, but not by the intended d_i correspond to the newly created *network coding opportunities* [7], [10], [12]. We relabel those packets as $\tilde{\mathbf{X}}_i \triangleq \{\tilde{X}_{i,1}, \tilde{X}_{i,2}, \cdots, \tilde{X}_{i,L_i}\}$ where L_i represents the total number of those coding opportunities. The next step is to redistribute these L_i coding opportunities among all M inputs in a way that is proportional to $\bar{t}_{\mathrm{C},m}(1_{\{i=1\}} p_{10}^{[m]} + 1_{\{i=2\}} p_{01}^{[m]} + p_{11}^{[m]})$. Namely, we find $L_i^{[1]}$ to $L_i^{[M]}$ such that

$$L_i = \sum_{m=1}^{M} L_i^{[m]} \tag{83}$$

$$L_i^{[m]} \propto \bar{t}_{\mathrm{C},m}(1_{\{i=1\}} p_{10}^{[m]} + 1_{\{i=2\}} p_{01}^{[m]} + p_{11}^{[m]}). \tag{84}$$

For each input m, we now have $L_1^{[m]}$ number of $\tilde{\mathbf{X}}_1$ packets that have been heard by d_2 but not d_1, and $L_2^{[m]}$ number of $\tilde{\mathbf{X}}_2$ packets that have been heard by d_1 but not d_2. We denote those packets by $\tilde{\mathbf{X}}_1^{[m]}$ and $\tilde{\mathbf{X}}_2^{[m]}$, respectively. In Stage 3, for the m-th input, we choose one such $\tilde{\mathbf{X}}_1^{[m]}$ packet, denoted by $\tilde{X}_1^{[m]}$, and one such $\tilde{\mathbf{X}}_2^{[m]}$ packet, denoted by $\tilde{X}_2^{[m]}$; and send the linear sum $[\tilde{X}_1^{[m]} + \tilde{X}_2^{[m]}]$.

In the end of the current time t, perform the following procedure for $i = 1, 2$, respectively. If the linear sum is received by d_i, then we remove such $\tilde{X}_i^{[m]}$ from the pool of $\tilde{\mathbf{X}}_i^{[m]}$ packets and choose arbitrarily another $\tilde{X}_i^{[m]}$ from that pool for the next time slot. If the pool is empty, then we simply choose $\tilde{X}_i^{[m]} = 0$ instead. As can be seen, Stage 3 can be executed indefinitely if there is no time budget constraint, since Stage 3 allows us to send a linear sum with $\tilde{X}_1^{[m]} = \tilde{X}_2^{[m]} = 0$. To incorporate the time budget constraint, for each of the m-th input, we only run Stage 3 for $n\bar{t}_{\mathrm{C},m}$ number of time slots. We now define the following encoding error for Stage 3.

Encoding error for Stage 3: There exists an m value such that at least one of the two pools: the $\tilde{\mathbf{X}}_1^{[m]}$ and $\tilde{\mathbf{X}}_2^{[m]}$ packets, is not empty after running stage 3 on the m-th input for $n \cdot \bar{t}_{\mathrm{C},m}$ time slots.

We now prove the following claims for the above network code construction.

Claim 1: We can always finish the above network coding scheme within the time budget n time slots. Proof: This is a direct result of (77).

Claim 2: If there is no encoding error in Stages 1 to 3, then both d_i, $i = 1, 2$, can successfully receive/decode all the desired packets $\mathbf{X}_i$ successfully. Proof: Per our construction, at the end of Stage 1, d_1 has received all the $\mathbf{X}_1$ packets except for those in $\tilde{\mathbf{X}}_1$ since there is no type-1 encoding error for Stage 1. Since there is no encoding error in Stage 3, d_1 can *decode* all the $\tilde{\mathbf{X}}_1^{[m]}$ packets through the m-th input for all $m = 1$ to M by subtracting the $\tilde{\mathbf{X}}_2$ packets overheard in

Stage 2. As a result, d_1 can receive/decode all $\mathbf{X}_1$ packets by the end of time n. The case for d_2 can be proven by symmetry.

Claim 2 implies that the decoding error probability is the probability that there is an encoding error in one of the three stages. Note that the above scheme is well defined for arbitrary n values. To complete the achievability proof, we thus need to show that for any $\epsilon > 0$, we can find a sufficiently large n such that the probability of encoding error is less than ϵ.

Given any $\epsilon > 0$, we first notice that the expected number of time slots to finish sending $nR_i^{[m]}$ packets over the m-th input in Stage 1 is $nR_i^{[m]}/(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]})$. By (76) and the law of large numbers (LLN) there exists an $n_{1,m,\text{type-1}}$ such that when $n \geq n_{1,m,\text{type-1}}$, the probability that there is a type-1 encoding error for Stage 1 over the m-th input is $\leq \frac{\epsilon}{6M}$. We then notice that conditioning on there is no type-1 encoding error in Stage 1, the expected number of packets that are heard by d_2 but not by d_1 is $nR_1^{[m]}\left(\frac{p_{01}^{[m]}}{p_{01}^{[m]}+p_{10}^{[m]}+p_{11}^{[m]}}\right)$. By (81) and the LLN, there exists an $n_{1,m,\text{type-2}}$ such that when $n \geq n_{1,m,\text{type-2}}$, the conditional probability that there is a type-2 encoding error for Stage 1 over the m-th input, conditioning on the event that there is no type-1 encoding error in Stage 1, is $\leq \frac{\epsilon}{6M}$. Symmetrically, there exists an $n_{2,m,\text{type-1}}$ such that when $n \geq n_{2,m,\text{type-1}}$ the probability that there is a type-1 encoding error for Stage 2 over the m-th input is $\leq \frac{\epsilon}{6M}$. Similarly, there exists an $n_{2,m,\text{type-2}}$ such that when $n \geq n_{2,m,\text{type-2}}$ the conditional probability that there is a type-2 encoding error for Stage 2 over the m-th input, conditioning on the event that there is no type-1 encoding error in Stage 2, is $\leq \frac{\epsilon}{6M}$.

We now consider the encoding error in Stage 3. We first quantify the probability that the $\tilde{\mathbf{X}}_1^{[m]}$ pool is not empty after running stage 3 on the m-th input for $n \cdot \bar{t}_{\text{C},m}$ time slots. We notice that conditioning on the event that there is no type-2 encoding error in Stage 1, by (81) we have

$$L_1 \leq \sum_{m=1}^{M} nR_1^{[m]} \left(\frac{p_{01}^{[m]}}{p_{01}^{[m]} + p_{10}^{[m]} + p_{11}^{[m]}}\right)\left(1 + \frac{\delta}{2}\right).$$

By (79), we have

$$L_1\left(\frac{1+\delta}{1+\delta/2}\right) \leq \sum_{m=1}^{M} n\bar{t}_{\text{C},m}(p_{10}^{[m]} + p_{11}^{[m]}).$$

By (83) and (84), we then have

$$\left(\frac{L_1^{[m]}}{p_{10}^{[m]} + p_{11}^{[m]}}\right)\left(\frac{1+\delta}{1+\delta/2}\right) \leq n\bar{t}_{\text{C},m}. \tag{85}$$

We then notice that the expected number of time slots to finish sending the $L_1^{[m]}$ packets over the m-th input in Stage 3 is $L_1^{[m]}/(p_{10}^{[m]} + p_{11}^{[m]})$. By (85) and the LLN, there exists an $n_{1,m,\text{stage-3}}$ such that when $n \geq n_{1,m,\text{stage-3}}$, the conditional probability that we are not able to finish sending the $L_1^{[m]}$ packets over the m-th input within the allocated $n\bar{t}_{\text{C},m}$ time slots, conditioning on the event of no type-2 encoding error in Stage 1, is $\leq \frac{\epsilon}{6M}$. Symmetrically, there exists an $n_{2,m,\text{stage-3}}$ such that when $n \geq n_{2,m,\text{stage-3}}$, the conditional probability that we are not able to finish sending the $L_2^{[m]}$ packets over the m-th input within the allocated $n\bar{t}_{\text{C},m}$ time slots, conditioning on the event of no type-2 encoding error in Stage 2, is $\leq \frac{\epsilon}{6M}$.

From the above reasoning, for any

$$n \geq \max\{n_{i,m,\text{type-1}}, n_{i,m,\text{type-2}}, n_{i,m,\text{stage-3}} : \forall i \in \{1,2\}, m \in \{1,\cdots,M\}\}$$

the probability of having at least one encoding error in Stages 1 to 3 can be upper bounded by the union bound

$$2M\frac{\epsilon}{6M} + 2M\frac{\epsilon}{6M} + 2M\frac{\epsilon}{6M}$$

where the first term corresponds to the type-1, type-2 errors in Stage 1; the second term corresponds to the type-1, type-2 errors in Stage 2; and the last term corresponds to the encoding error in Stage 3.

We have thus proven that for any $\epsilon > 0$, the decoding error probability for the above scheme is no larger than ϵ when n is sufficiently large. The proof of Lemma 3 is complete.[15]

C. *Remarks on Deriving Complete Proofs Based on The First-Order Analysis*

One can trace the complete proof and see that it follows the logic flow of the first-order analysis very closely. In general, the first order analysis can be made rigorous by the LLN as long as the following is true.

- The normalized time budget constraint, e.g., (10) to (12), is a strict inequality so that there we can add the δ term, e.g., (76) and (79)–(82), which is critical for the LLN analysis.
- The number of "stages" in the design of the network code must be a finite number not depending on n, the network code length. For example, there are 3 stages in the aforementioned network code. In this way, when we invoke the LLN, the error probability will approaches zero for a sufficiently large n when we apply the union bound analysis.

APPENDIX B
PROOFS OF PROPOSITION 2 AND LEMMA 4

Proof of Proposition 2: We prove the following equivalent statement. For any given rate vector (R_1, R_2) satisfying

[15] Astute readers may notice that there is one subtlety omitted in the above argument: It does not consider causality constraints. That is, when we redistribute the coding opportunities, we implicitly assume that all M inputs have finished executing Stages 1 and 2 at the same time so that we can collect all the L_i coding opportunities, denoted by $\tilde{\mathbf{X}}_i$, and redistribute them to all M inputs simultaneously. This implicit assumption is in general not true. However, we can circumvent this difficulty by some pipelining techniques. That is, we consider the $(F+1)$ number of *frames*, where F is some large number. Then we can use the aforementioned network code for F times but each network code usage is spread over two consecutive frames. Namely, we always collect the coding opportunities in the first "half" of each frame and always perform redistribution and linear-sum encoding in the second "half" of each frame. However, we impose the requirement that the coding opportunities collected in (the first half of) one frame is always redistributed and encoded in (the second half of) the next frame. Since we spread the collection and the consumption of the coding opportunities into two frames, we circumvent the causality constraint. In the achievability proof of our main result in Proposition 1, we do not use the concept of *redistribution* and thus we do not need any pipelining technique when describing the proposed network coding solution.

the outer bound of Lemma 2, there exist $18M$ non-negative values $x_{\mathbf{b}}^{[m]}$ for all $\mathbf{b} \in \mathsf{FTs}$ and $m = 1, \cdots, M$, and 7 non-negative values y_1 to y_7 such that jointly they satisfy (15) to (28) in Proposition 1.

The proof is done by explicit construction that consists of five main steps.

Step 1: Given the the values of R_1, R_2, and the $R_i^{[m,k]}$ variables satisfying Lemma 2, we will first construct a set of $9M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $m \in \{1, \cdots, M\}$ and $\mathbf{b} \in \{0, 1, 2, 9, 18, 27, 31, 63, 95\}$ satisfying the following $4 + 2M$ (in)equalities:

$$R_1 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{18,27,31,63\}} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]} \right), \quad (86)$$

$$R_1 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,2,18\}} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right), \quad (87)$$

$$R_2 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,1,9\}} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right), \quad (88)$$

$$R_2 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{9,27,31,95\}} x_{\mathbf{b}}^{[m]} \right) \cdot \left(p_{01}^{[m]} + p_{11}^{[m]} \right), \quad (89)$$

and for all $m \in \{1, \cdots, M\}$

$$x_0^{[m]} + x_1^{[m]} + x_9^{[m]} + x_{18}^{[m]} + x_{27}^{[m]} + x_{31}^{[m]} + x_{63}^{[m]} \le 1, \quad (90)$$

$$x_0^{[m]} + x_2^{[m]} + x_9^{[m]} + x_{18}^{[m]} + x_{27}^{[m]} + x_{31}^{[m]} + x_{95}^{[m]} \le 1. \quad (91)$$

Our construction is as follows. For any $m \in \{1, \cdots, M\}$, we set

$$x_{63}^{[m]} = \frac{R_1^{[m,1]}}{p_{10}^{[m]} + p_{11}^{[m]}}, \qquad x_2^{[m]} = \frac{R_1^{[m,2]}}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}},$$
$$x_1^{[m]} = \frac{R_2^{[m,1]}}{p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}}, \qquad x_{95}^{[m]} = \frac{R_2^{[m,2]}}{p_{01}^{[m]} + p_{11}^{[m]}},$$

and $x_{\mathbf{b}}^{[m]} = 0$ for all other $\mathbf{b} \in \{0, 9, 18, 27, 31\}$. Obviously, the above construction satisfies

$$R_1^{[m,1]} = (x_{18}^{[m]} + x_{27}^{[m]} + x_{31}^{[m]} + x_{63}^{[m]}) \cdot (p_{10}^{[m]} + p_{11}^{[m]}), \quad (92)$$

$$R_1^{[m,2]} = (x_0^{[m]} + x_2^{[m]} + x_{18}^{[m]}) \cdot (p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}), \quad (93)$$

$$R_2^{[m,1]} = (x_0^{[m]} + x_1^{[m]} + x_9^{[m]}) \cdot (p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}), \quad (94)$$

$$R_2^{[m,2]} = (x_9^{[m]} + x_{27}^{[m]} + x_{31}^{[m]} + x_{95}^{[m]}) \cdot (p_{01}^{[m]} + p_{11}^{[m]}). \quad (95)$$

Then (7), (92), and (94) jointly imply (90). Symmetrically, (8), (93), and (95) jointly imply (91). The inequalities (86)–(89) are direct results of (6) and (92)–(95).

Step 2: We claim that not only can we find the $9M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $m \in \{1, \cdots, M\}$ and $\mathbf{b} \in \{0, 1, 2, 9, 18, 27, 31, 63, 95\}$ satisfying (86) to (91), but the construction of $x_{\mathbf{b}}^{[m]}$ can also be done in a way that additionally it satisfies the following $4M$ conditions as well.

$$\forall m, \quad x_1^{[m]} \cdot x_2^{[m]} = 0, \qquad x_{63}^{[m]} \cdot x_{95}^{[m]} = 0,$$
$$x_1^{[m]} \cdot x_{95}^{[m]} = 0, \quad \text{and} \quad x_2^{[m]} \cdot x_{63}^{[m]} = 0. \quad (96)$$

The reason is as follows. Suppose that for some $m_0 \in \{1, \cdots, M\}$, we have both $x_1^{[m_0]}$ and $x_2^{[m_0]}$ being strictly positive. Define $\Delta \triangleq \min(x_1^{[m_0]}, x_2^{[m_0]}) > 0$. We then notice that when setting $x_0^{[m_0]} \leftarrow x_0^{[m_0]} + \Delta$, $x_1^{[m_0]} \leftarrow x_1^{[m_0]} - \Delta$, $x_2^{[m_0]} \leftarrow x_2^{[m_0]} - \Delta$, and keeping the other $x_{\mathbf{b}}^{[m]}$ intact, all the newly constructed $x_{\mathbf{b}}^{[m]}$ are still non-negative. The new construction obviously satisfies $x_1^{[m_0]} \cdot x_2^{[m_0]} = 0$. Moreover, one can easily verify that (86) to (91) still hold after the above modification that "moves the weight of $x_1^{[m_0]}$ (and $x_2^{[m_0]}$) to $x_0^{[m_0]}$."

To ensure that at least one of $x_{63}^{[m]}$ and $x_{95}^{[m]}$ being zero for all m, we can perform a similar modification that "moves the weight of $x_{63}^{[m]}$ (and $x_{95}^{[m]}$) to $x_{27}^{[m]}$." Similarly, to ensure that at least one of $x_1^{[m]}$ and $x_{95}^{[m]}$ being zero, we can perform a modification that moves the weight of $x_1^{[m]}$ (and $x_{95}^{[m]}$) to $x_9^{[m]}$. To ensure that at least one of $x_2^{[m]}$ and $x_{63}^{[m]}$ being zero, we can move the weight of $x_2^{[m]}$ (and $x_{63}^{[m]}$) to $x_{18}^{[m]}$.

We also observe that jointly (96), (90), and (91) imply that for all $m \in \{1, \cdots, M\}$,

$$x_0^{[m]} + x_1^{[m]} + x_2^{[m]} + x_9^{[m]} + x_{18}^{[m]} + x_{27}^{[m]} + x_{31}^{[m]} + x_{63}^{[m]} + x_{95}^{[m]} \le 1. \quad (97)$$

The reason is that if $x_1^{[m]} > 0$, then by (96) we must have $x_2^{[m]} = x_{95}^{[m]} = 0$. Therefore, (90) implies (97). Symmetrically, if $x_2^{[m]} > 0$, then by (96) we must have $x_1^{[m]} = x_{63}^{[m]} = 0$. Therefore, (91) implies (97). If both $x_1^{[m]} = x_2^{[m]} = 0$, then by (96), at least one of $x_{63}^{[m]}$ and $x_{95}^{[m]}$ must be zero. If $x_{95}^{[m]} = 0$, then (90) implies (97). If $x_{63}^{[m]} = 0$, then (91) implies (97). From the above discussion, we can see that (97) always holds for our construction.

In sum, after Step 2 we have found a set of $9M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $m \in \{1, \cdots, M\}$ and $\mathbf{b} \in \{0, 1, 2, 9, 18, 27, 31, 63, 95\}$ satisfying (86)–(89) and (97).

Step 3: We claim that not only can we find the $9M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $m \in \{1, \cdots, M\}$ and $\mathbf{b} \in \{0, 1, 2, 9, 18, 27, 31, 63, 95\}$ satisfying simultaneously (86)–(89) and (97), but the construction of $x_{\mathbf{b}}^{[m]}$ can also be done such that $x_1^{[m]} = x_2^{[m]} = 0$ for all $m \in \{1, \cdots, M\}$. (Note that in the end result of this step we no longer require the $x_{\mathbf{b}}^{[m]}$ variables to satisfy (96).)

The reason is as follows. Suppose that for some $m_0 \in \{1, \cdots, M\}$, we have one of $x_1^{[m_0]}$ and $x_2^{[m_0]}$ being strictly positive. Without loss of generality, we assume $x_1^{[m_0]} = 0$ and $x_2^{[m_0]} > 0$. We now describe our construction. In our construction, we decrease $x_2^{[m_0]}$ by Δ_2; increase $x_{18}^{[m_0]}$ by Δ_{18}; decrease each of $x_{27}^{[m]}$, $x_{31}^{[m]}$, and $x_{63}^{[m]}$ by $\Delta_{27}^{[m]}$, $\Delta_{31}^{[m]}$, and $\Delta_{63}^{[m]}$ for all $m \in \{1, \cdots, M\}$, respectively; increase $x_{95}^{[m]}$ by $\Delta_{95}^{[m]}$ for all $m \in \{1, \cdots, M\}$, respectively; and keep all the remaining $x_{\mathbf{b}}^{[m]}$ variables unchanged. We choose the Δ values sequentially as follows.

Choose $\Delta_2 = x_2^{[m_0]}$ first. Then choose $\Delta_{18} = \Delta_2$. After that, choose $\{\Delta_{27}^{[m]}, \Delta_{31}^{[m]}, \Delta_{63}^{[m]} : \forall m\}$ that satisfy simultaneously the following two conditions: (i) $0 \le \Delta_{\mathbf{b}}^{[m]} \le x_{\mathbf{b}}^{[m]}$ for all $\mathbf{b} \in \{27, 31, 63\}$; and (ii) the following equality holds

$$x_2^{[m_0]} \cdot (p_{10}^{[m_0]} + p_{11}^{[m_0]}) = \sum_{m=1}^{M} \left(\Delta_{27}^{[m]} + \Delta_{31}^{[m]} + \Delta_{63}^{[m]}\right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right). \quad (98)$$

Note that while the choices of Δ_2 and Δ_{18} are strictly positive, $\{\Delta_{27}^{[m]}, \Delta_{31}^{[m]}, \Delta_{63}^{[m]} : \forall m\}$ do not need to be strictly positive and some of them can be zero. Moreover, the above choice of $\{\Delta_{27}^{[m]}, \Delta_{31}^{[m]}, \Delta_{63}^{[m]} : \forall m\}$ is always feasible although the choice may not be unique. The reason is as follows. We notice that since both the left-hand sides of (86) and (87) are R_1, we thus have (87) = (86), which implies

$$\begin{aligned} &x_2^{[m_0]} \cdot (p_{10}^{[m_0]} + p_{11}^{[m_0]}) \\ &\quad + x_2^{[m_0]} \cdot p_{01}^{[m_0]} + \sum_{m \neq m_0} x_2^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right) \\ &\quad + \sum_{m=1}^{M} x_0^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right) + \sum_{m=1}^{M} x_{18}^{[m]} \cdot p_{01}^{[m]} \\ &= \sum_{m=1}^{M} \left(x_{27}^{[m]} + x_{31}^{[m]} + x_{63}^{[m]}\right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right). \end{aligned}$$

Therefore, we have

$$x_2^{[m_0]} \cdot (p_{10}^{[m_0]} + p_{11}^{[m_0]}) \leq \sum_{m=1}^{M} \left(x_{27}^{[m]} + x_{31}^{[m]} + x_{63}^{[m]}\right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right). \quad (99)$$

Comparing (98) and (99), it is clear that we can choose $\Delta_{\mathbf{b}}^{[m]}$ in the interval $[0, x_{\mathbf{b}}^{[m]}]$ for all $\mathbf{b} \in \{27, 31, 63\}$ such that jointly they satisfy (98).

Finally choose $\Delta_{95}^{[m]} = \Delta_{27}^{[m]} + \Delta_{31}^{[m]}$ for all $m \in \{1, \cdots, M\}$.

We can now verify that (86) still holds due to $\Delta_{18} = \Delta_2 = x_2^{[m_0]}$ and (98). (87) still holds due to $\Delta_{18} = \Delta_2$. (88) still holds since we did not change the corresponding $x_{\mathbf{b}}^{[m]}$ variables. (89) still holds since $\Delta_{95}^{[m]} = \Delta_{27}^{[m]} + \Delta_{31}^{[m]}$. (97) holds since any potential increment is offset by an identical decrement. More specifically, the increase of $x_{18}^{[m_0]}$ is offset by the decrease of $x_2^{[m_0]}$, and the increase of $x_{95}^{[m]}$ is offset by the decrease of $x_{27}^{[m]}$ and $x_{31}^{[m]}$. After the above modification, we now have $x_1^{[m_0]} = x_2^{[m_0]} = 0$. We can repeat the process until $x_1^{[m]} = x_2^{[m]} = 0$ for all m.

Step 4: We claim that not only can we find the $9M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $m \in \{1, \cdots, M\}$ and $\mathbf{b} \in \{0, 1, 2, 9, 18, 27, 31, 63, 95\}$ satisfying simultaneously (i) $x_1^{[m]} = x_2^{[m]} = 0$ for all m; (ii) (86) to (89); and (iii) (97), but the construction of $x_{\mathbf{b}}^{[m]}$ can also be done such that the following equality holds.

$$\sum_{m=1}^{M} x_0^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right) = \sum_{m=1}^{M} x_{27}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right). \quad (100)$$

The proof is as follows. We start from the $x_{\mathbf{b}}^{[m]}$ variables satisfying the above Conditions (i) to (iii). Define α and β to be

$$\alpha \triangleq \sum_{m=1}^{M} x_0^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right), \quad (101)$$

$$\beta \triangleq \sum_{m=1}^{M} x_{27}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right), \quad (102)$$

where both α and β are non-negative by definition. Consider the following three cases depending on the order of α and β.

Case 1: $\alpha = \beta$. In this case, (100) holds already. No further modification is necessary.

Case 2: $\alpha < \beta$. In this case, for each m with $x_{27}^{[m]} > 0$, we can continuously decrease $x_{27}^{[m]}$ and increase $x_{31}^{[m]}$ by the same amount, i.e., "move the weight from $x_{27}^{[m]}$ to $x_{31}^{[m]}$." When we continuously move the weight from $x_{27}^{[m]}$ to $x_{31}^{[m]}$, one can easily see that (86)–(89) and (97) still hold; the α value remains intact; and the β value decreases continuously. If we move all the weight from $x_{27}^{[m]}$ to $x_{31}^{[m]}$ for all $m \in \{1, \cdots M\}$, then the β value decreases to 0 continuously. Therefore, somewhere along the process, we will have $\alpha = \beta$. Whenever $\alpha = \beta$, we stop moving any weight from $x_{27}^{[m]}$ to $x_{31}^{[m]}$. The final $x_{\mathbf{b}}^{[m]}$ values satisfy (100).

Case 3: $\alpha > \beta$. In this case, for each m when $x_{31}^{[m]} > 0$, we can continuously decrease $x_{31}^{[m]}$ and increase $x_{27}^{[m]}$ by the same amount, i.e., "move the weight from $x_{31}^{[m]}$ to $x_{27}^{[m]}$." Note that moving the weight from $x_{31}^{[m]}$ to $x_{27}^{[m]}$ increase β while all the equalities (86)–(89) and (97) still hold. If along the process we have $\alpha = \beta$, then no further modification is necessary. Suppose after moving all the weight of $x_{31}^{[m]}$ to $x_{27}^{[m]}$ we still have $\alpha > \beta$. (That is, now $x_{31}^{[m]} = 0$ for all m.) In this case, we need to perform the following further modification on $x_{\mathbf{b}}^{[m]}$.

To that end, we define 7 different summations as follows.

$$\begin{aligned} \mathsf{sum}_1 &\triangleq \sum_{m=1}^{M} \left(x_{18}^{[m]} + x_{27}^{[m]}\right) \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right), \\ \mathsf{sum}_2 &\triangleq \sum_{m=1}^{M} x_{63}^{[m]} \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right), \\ \mathsf{sum}_3 &\triangleq \sum_{m=1}^{M} x_0^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right), \\ \mathsf{sum}_4 &\triangleq \sum_{m=1}^{M} x_{18}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right), \\ \mathsf{sum}_5 &\triangleq \sum_{m=1}^{M} x_9^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right), \\ \mathsf{sum}_6 &\triangleq \sum_{m=1}^{M} \left(x_9^{[m]} + x_{27}^{[m]}\right) \cdot \left(p_{01}^{[m]} + p_{11}^{[m]}\right), \\ \mathsf{sum}_7 &\triangleq \sum_{m=1}^{M} x_{95}^{[m]} \cdot \left(p_{01}^{[m]} + p_{11}^{[m]}\right). \end{aligned} \quad (103)$$

Since by our construction we have $x_{\mathbf{b}}^{[m]} = 0$ for all $\mathbf{b} \in$

$\{1, 2, 31\}$, then (86) to (89) can be rewritten as

$$R_1 = \mathsf{sum}_1 + \mathsf{sum}_2, \quad (104)$$
$$R_1 = \mathsf{sum}_3 + \mathsf{sum}_4, \quad (105)$$
$$R_2 = \mathsf{sum}_3 + \mathsf{sum}_5,$$
$$R_2 = \mathsf{sum}_6 + \mathsf{sum}_7.$$

Recall that we are focusing on the case $\alpha > \beta$. We claim that both sum_2 and sum_7 are strictly positive. The reason is as follows. We first notice that both the left-hand side of (104) and (105) are R_1. The equality $(105) = (104)$ implies that

$$\mathsf{sum}_3 + \sum_{m=1}^{M} x_{18}^{[m]} \cdot p_{01}^{[m]} = \sum_{m=1}^{M} x_{27}^{[m]} \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right) + \mathsf{sum}_2. \quad (106)$$

By definitions (101) and (103), we have $\mathsf{sum}_3 = \alpha$. Therefore, (106) and the assumption $\beta < \alpha$ imply

$$\beta + \sum_{m=1}^{M} x_{18}^{[m]} \cdot p_{01}^{[m]} < \sum_{m=1}^{M} x_{27}^{[m]} \cdot \left(p_{10}^{[m]} + p_{11}^{[m]}\right) + \mathsf{sum}_2,$$
$$\Leftrightarrow \sum_{m=1}^{M} \left(x_{18}^{[m]} + x_{27}^{[m]}\right) \cdot p_{01}^{[m]} < \mathsf{sum}_2, \quad (107)$$

where "$\Leftrightarrow$" follows from the definition of β in (102). The strict inequality (107) shows that sum_2 is strictly positive. By symmetric arguments, one can show that sum_7 is strictly positive provided $\alpha > \beta$.

In the following modification process, we will continuously decrease the value of α while keeping the value of β intact until we finally have $\alpha = \beta$. After $\alpha = \beta$, no further modification is necessary since we have attained (100).

From our previous analysis, we know that whenever $\alpha > \beta$, both sum_2 and sum_7 are strictly positive. Therefore, we can find at least one pair of indices $\{m_1, m_2\} \subset \{1, \cdots, M\}$ such that $x_{63}^{[m_1]} > 0$ and $x_{95}^{[m_2]} > 0$. In our modification, we decrease $x_{63}^{[m_1]}$ by Δ_{63}; increase $x_{18}^{[m_1]}$ by Δ_{18}; decrease $x_{95}^{[m_2]}$ by Δ_{95}; increase $x_9^{[m_2]}$ by Δ_9; decrease $x_0^{[m]}$ by $\Delta_0^{[m]}$ for all $m \in \{1, \cdots, M\}$, respectively; and keep all other $x_{\mathbf{b}}^{[m]}$ intact. We choose the Δ values as follows.

We choose two strictly positive but sufficiently small delta terms $\Delta_{63} \leq x_{63}^{[m_1]}$ and $\Delta_{95} \leq x_{95}^{[m_2]}$ that satisfy the following equality:

$$\Delta_{63} \cdot (p_{10}^{[m_1]} + p_{01}^{[m_1]} + p_{11}^{[m_1]}) = \Delta_{95} \cdot (p_{10}^{[m_2]} + p_{01}^{[m_2]} + p_{11}^{[m_2]}). \quad (108)$$

What do we mean by *sufficiently small* Δ_{63} and Δ_{95} will be clear in the later discussion.

Then choose $\Delta_{18} = \Delta_{63}$ and $\Delta_9 = \Delta_{95}$. After that, we choose $\{\Delta_0^{[m]} : \forall m\}$ that satisfy simultaneously the following two conditions: (i) $0 \leq \Delta_0^{[m]} \leq x_0^{[m]}$; and (ii) the following (in)equalities hold

$$\Delta_{18} \cdot (p_{10}^{[m_1]} + p_{01}^{[m_1]} + p_{11}^{[m_1]}) = \sum_{m=1}^{M} \Delta_0^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right) \leq \alpha - \beta. \quad (109)$$

Note that while the choices of Δ_{63}, Δ_{95}, Δ_{18}, and Δ_9 are strictly positive, $\{\Delta_0^{[m]} : \forall m\}$ do not need to be strictly positive and some of them can be zero. Also, the above choice of $\{\Delta_0^{[m]} : \forall m\}$ is always feasible provided the strictly positive term Δ_{18} is sufficiently small (recalling that we choose sufficiently small Δ_{63} to begin with). The reason is as follows. Firstly, we can always choose $\Delta_0^{[m]} = \frac{\Delta_{18} \cdot (p_{10}^{[m_1]} + p_{01}^{[m_1]} + p_{11}^{[m_1]})}{\alpha} x_0^{[m]}$ for all $m \in \{1, \cdots, M\}$. By the definition of α in (101), such choices of $\{\Delta_0^{[m]} : \forall m\}$ satisfy the equality in (109). Secondly, observe that $\alpha > \beta$ in the case we are considering. A sufficiently small $\Delta_{18} > 0$ can thus satisfy the inequality in (109).

We will now verify that with the above choices of the Δ terms, the new modified $\{x_{\mathbf{b}}^{[m]} : \forall m\}$ always satisfy (86)–(89), (97), and $\alpha \geq \beta$. Specifically, (86) still holds due to $\Delta_{18} = \Delta_{63}$. (87) still holds due to (109). (88) still holds due to $\Delta_9 = \Delta_{95}$, (108), $\Delta_{63} = \Delta_{18}$, and (109). (89) still holds since $\Delta_9 = \Delta_{95}$. (97) holds since any potential increment is offset by an identical decrement. More specifically, the increase of $x_{18}^{[m_1]}$ is offset by the decrease of $x_{63}^{[m_1]}$, and the increase of $x_9^{[m_2]}$ is offset by the decrease of $x_{95}^{[m_2]}$. Also, we still have $\alpha \geq \beta$ since β remains unchanged (we did not modify $x_{27}^{[m]}$) and the decrease of α is $\sum_{m=1}^{M} \Delta_0^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]}\right)$, which is upper bounded by $\alpha - \beta$ as specified in (109).

Note that whenever $\alpha > \beta$, we have sum_2 and sum_7 being strictly positive, which in turn implies that we can successfully perform the above modification process. Therefore, we can repeat this process until $\alpha = \beta$. The modification of Step 4 is now complete.

Step 5: After Step 4, only $7M$ out of the $9M$ variables $x_{\mathbf{b}}^{[m]}$ can be non-zero (those with $\mathbf{b} \in \{0, 9, 18, 27, 31, 63, 95\}$) and $2M$ of them are strictly 0 (those with $\mathbf{b} \in \{1, 2\}$). The $9M$ variables satisfy (86)–(89), (97), and (100). Given such $9M$ variables $x_{\mathbf{b}}^{[m]}$, we can extend them to $18M$ variables by setting $x_{\mathbf{b}}^{[m]} = 0$ for all $\mathbf{b} \in \{3, 7, 11, 15, 19, 23, 47, 87, 127\}$. We now claim that the final $x_{\mathbf{b}}^{[m]}$ satisfy all the (in)equalities (15) to (28) in Proposition 1.

The proof is by direct substitution. The time-sharing inequality (15) is a direct result of (97). With our final $x_{\mathbf{b}}^{[m]}$, the

y variables in (16) to (22) become

$$y_1 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9,18,27,31,63\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{11}^{[m]} \right), \tag{110}$$

$$y_2 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9,18,27,31,95\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{01}^{[m]} + p_{11}^{[m]} \right), \tag{111}$$

$$y_3 = R_1 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{11}^{[m]} \right), \tag{112}$$

$$y_4 = R_2 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,18\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{01}^{[m]} + p_{11}^{[m]} \right), \tag{113}$$

$$y_5 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9,18,27\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right), \tag{114}$$

$$y_6 = R_1 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right), \tag{115}$$

$$y_7 = R_2 + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,18\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right). \tag{116}$$

By the above construction of the y variables, one can easily verify that $y_3 \leq y_6$ and $y_4 \leq y_7$, which leads to (23). In the following, we show that $y_3 = y_1$ and $y_4 = y_2$, which leads to (28). We observe that by substituting the R_1 term in (112) by (86), we have

$$y_3 = \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9,18,27,31,63\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{11}^{[m]} \right) = y_1,$$

where the last inequality follows from (110). Symmetrically, we can verify that $y_4 = y_2$.

What remains to be verified is that (24) to (27) also hold. To that end, we will show that $y_5 = y_6 = y_7 = R_1 + R_2$, which jointly with $y_3 = y_1$ and $y_4 = y_2$ imply (24) to (27). We first show that $y_6 = R_1 + R_2$. By (115) and (88), we quickly have $y_6 = R_1 + R_2$ (recalling that $x_1^{[m]} = 0$ by our construction). Symmetrically, we can verify that $y_7 = R_1 + R_2$ as well. By (114) and (100) we have

$$\begin{aligned} y_5 &= \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,18\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right) \\ &\quad + \sum_{m=1}^{M} \left(\sum_{\forall \mathbf{b} \in \{0,9\}} \right) x_{\mathbf{b}}^{[m]} \cdot \left(p_{10}^{[m]} + p_{01}^{[m]} + p_{11}^{[m]} \right) \\ &= R_1 + R_2, \end{aligned}$$

where the last equality follows from (87) and (88) and $x_2^{[m]} = x_1^{[m]} = 0$ in our construction. From the above discussion, we have proven that the $\{x_{\mathbf{b}}^{[m]}\}$ in our construction satisfy all the linear conditions of Proposition 1. The proof is thus complete. ■

We now present the following alternative statement of the capacity region, which is a byproduct of the above proof of Proposition 2.

Lemma 8: For any fixed $\mathsf{GF}(q)$, consider any cross-input independent M-input 2-receiver broadcast PEC. A rate vector (R_1, R_2) is in the capacity region if and only if there exist $7M$ non-negative variables $x_{\mathbf{b}}^{[m]}$ for all $\mathbf{b} \in \{0, 9, 18, 27, 31, 63, 95\}$, $m \in \{1, \cdots, M\}$, and 7 non-negative variables y_1 to y_7 such that jointly they satisfy

$$\forall m \in \{1, \cdots, M\}, \qquad \sum_{\forall \mathbf{b} \in \{0,9,18,27,31,63,95\}} x_{\mathbf{b}}^{[m]} \leq 1,$$

eqs. (110) to (116), $y_1 = y_3$, $y_2 = y_4$, and $y_5 = y_6 = y_7 = R_1 + R_2$.

Proof: If (R_1, R_2) is in the capacity region, then the LP problem in Lemma 2 is feasible. By the proof of Proposition 2, we can compute a set of $x_{\mathbf{b}}^{[m]}$ and y_i values following the 5-step approach detailed in the proof of Proposition 2. And in the end of that proof, we have already shown that the resulting $x_{\mathbf{b}}^{[m]}$ and y_i values satisfy the description of Lemma 8.

Conversely, if there exists a set of $x_{\mathbf{b}}^{[m]}$ and y_i values satisfying Lemma 8, then (R_1, R_2) is in the capacity region since those $x_{\mathbf{b}}^{[m]}$ and y_i values will also satisfy the LP problem in Proposition 1. The proof is thus complete. ■

Lemma 4 can now be proven as follows. Consider any given (R_1, R_2). If the LP problem in Proposition 1 is feasible, then (R_1, R_2) must be in the capacity region. Therefore, there exist $7M$ $x_{\mathbf{b}}^{[m]}$ values and 7 y_i values satisfying the conditions in Lemma 8. By simple algebraic verification, those $7M$ $x_{\mathbf{b}}^{[m]}$ values and 7 y_i values also satisfy the conditions in Lemma 4. The proof of Lemma 4 is thus complete.

APPENDIX C
PROOF OF LEMMA 5

Lemma 5 can be proven by interleaving several CH_1-based codes over the time axis and the M input symbols, and then apply the interleaved super code to CH_2.

Specifically, we consider M super time slots. Each super time slot contains n time slots. If we use a CH_1-based code for each super time slot and the underlying channel is CH_1, then overall we can send $M \cdot nR_i$ symbols from the source s to destination d_i, $i = 1, 2$, within $M \cdot n$ time slots. For any $l \in \{1, \cdots, n\}$ and $h \in \{0, \cdots, (M-1)\}$, we use "the l-th (time) instant of the h-th original code" to refer to the M-dimensional symbol vector $\mathbf{W}$ that is sent by the h-th code[16] during the l-th time instant of the corresponding super time slot.

Now we interleave the M original CH_1-based codes in the following way and apply the end results to CH_2. That is, for any time slot $t = 1, \cdots, (Mn + M - 1)$ and any $m \in \{1, \cdots, M\}$, we set the m-th input of the new interleaved code

[16] We label the M codes of interest as the 0-th code to the $(M-1)$-th code for simpler notation in the subsequent discussion.

to 0 if $\lceil \frac{t-m+1}{M} \rceil \notin \{1, \cdots, n\}$. If $\lceil \frac{t-m+1}{M} \rceil \in \{1, \cdots, n\}$, we set the m-th input of the new interleaved code to the m-th input of the $\lceil \frac{t-m+1}{M} \rceil$-th instant of the $((t-m) \bmod M)$-th original code.

One can quickly see that interleaving breaks the cross-input dependence. Therefore when applying the above interleaved scheme to CH_2, we can transmit MnR_1 and MnR_2 information packets to d_1 and d_2, respectively, within $(Mn+M-1)$ time slots. The effective rates become $\frac{Mn}{Mn+M-1}(R_1, R_2)$. Using a sufficiently large n, the throughput loss (due to interleaving) is negligible. The proof of Lemma 5 is thus complete.

Appendix D
Derivation of The Rank Conversion Equalities

Eq. (21) has been proven in Section V-C. The definitions in this appendix follow the ones used in Section V-C.

Consider $A_1 = S_1$. By (29) we have $\text{Rank}(A_1(0)) = 0$. We then note that when s sends a $\mathbf{c}_t \in \mathsf{TYPE}_\mathbf{b}$ for some $\mathbf{b}$ with b_1 being 0, then that $\mathbf{c}_t$ is not in $A_1 = S_1$. Therefore, whenever d_1 receives $W(t) = \overline{\mathbf{X}} \cdot \mathbf{c}_t^{\mathrm{T}}$ successfully, the rank of A_1 will increase by one. We thus have

$$\text{Rank}(A_1(0)) + \sum_{\forall \mathbf{b} \text{ w. } b_1=0} \left(\sum_{t=1}^{n} 1_{\left\{ \substack{\mathbf{c}_t \in \mathsf{TYPE}_\mathbf{b}, \text{ and} \\ d_1 \text{ receives it}} \right\}} \right) = \text{Rank}(A_1(n)). \quad (117)$$

Taking the normalized expectation of (117), counting only the FTs, and by the linearity of expectation and the stationarity and memorylessness of the channel, we have proven that (16) must hold for $M = 1$.

Consider $A_3 = S_1 \oplus \Omega_1$ and we have $\text{Rank}(A_3(0)) = nR_1$. When s sends a $\mathbf{c}_t \in \mathsf{TYPE}_\mathbf{b}$ for some $\mathbf{b}$ with b_3 being 0, then that $\mathbf{c}_t$ is not in $A_3 = S_1 \oplus \Omega_1$. Therefore, whenever d_1 receives $W(t) = \overline{\mathbf{X}} \cdot \mathbf{c}_t^{\mathrm{T}}$ successfully, the rank of A_3 will increase by one. We thus have

$$\text{Rank}(A_3(0)) + \sum_{\forall \mathbf{b} \text{ w. } b_3=0} \left(\sum_{t=1}^{n} 1_{\left\{ \substack{\mathbf{c}_t \in \mathsf{TYPE}_\mathbf{b}, \text{ and} \\ d_1 \text{ receives it}} \right\}} \right) = \text{Rank}(A_3(n)). \quad (118)$$

Taking the normalized expectation of (118), we have proven (18) for $M = 1$.

Consider $A_5 = S_1 \oplus S_2$ and we have $\text{Rank}(A_5(0)) = 0$. When s sends a $\mathbf{c}_t \in \mathsf{TYPE}_\mathbf{b}$ for some $\mathbf{b}$ with b_5 being 0, then that $\mathbf{c}_t$ is not in $A_5 = S_1 \oplus S_2$. Therefore, whenever one of $\{d_1, d_2\}$ receives $W(t) = \overline{\mathbf{X}} \cdot \mathbf{c}_t^{\mathrm{T}}$ successfully, the rank of A_5 will increase by one. We thus have

$$\text{Rank}(A_5(0)) + \sum_{\forall \mathbf{b} \text{ w. } b_5=0} \left(\sum_{t=1}^{n} 1_{\left\{ \substack{\mathbf{c}_t \in \mathsf{TYPE}_\mathbf{b}, \text{ and} \\ \text{one of } \{d_1, d_2\} \text{ receives it}} \right\}} \right) = \text{Rank}(A_5(n)). \quad (119)$$

Taking the normalized expectation of (20), we have proven (20) for $M = 1$.

Since (17), (19), and (22) are symmetric versions of (16), (18), and (21), the proof of (16) to (22) for the case of $M = 1$ is complete.

Appendix E
Proof of Lemma 7

We first prove Property 1 of Lemma 7. When we substitute $\vec{\mathbf{x}}$ by $\vec{\mathbf{0}}$ into the first three groups of conditions in Proposition 1, it is clear that it satisfies Group 1, the time sharing condition; $y_1 = y_2 = y_5 = 0$, $y_3 = y_6 = R_1$, and $y_4 = y_7 = R_2$; and all 7 rank comparison conditions in Group 3 are satisfied. As a result, $\vec{\mathbf{0}}$ is in Γ. By definition, the given 18-dimensional point $\vec{\tilde{x}}$ satisfies the conditions in Groups 1 to 3. As a result, $\vec{\tilde{x}}$ is also in Γ. Property 1 is thus proven.

We now prove Property 2. Since (R_1, R_2) is in the interior of the capacity region, we have $R_1 > 0$ and $R_2 > 0$. Also since the 2-receiver broadcast PEC is not physically degraded and since[17] $M = 1$, we have $p_{01}^{[1]} > 0$ and $p_{10}^{[1]} > 0$. In the following, we will construct step-by-step a point $\vec{x}$ that is in the interior of Γ.

An interior point of Γ is a point such that all 8 inequalities in (15) and (23)–(27) are satisfied with strict inequalities and each of the 18 coordinates is strictly positive. We first consider the all-zero point $\vec{\mathbf{0}}$ and set the variable values to $\vec{x} = \vec{\mathbf{0}}$. Then we have $y_1 = y_2 = y_5 = 0$, $y_3 = y_6 = R_1$, and $y_4 = y_7 = R_2$. (15) and (24) are thus satisfied with strict inequality while all the other 5 rank-comparison inequalities (23) and (25)–(27) are satisfied with equality.

Since $x_0^{[1]}$ participates in all 7 rank conversion equalities (16) to (22), if we increase the value of $x_0^{[1]}$ (currently being 0) by a small constant $\delta_0 > 0$, then the values of y_1 and y_3 will increase by $\delta_0(p_{10}^{[1]} + p_{11}^{[1]})$; the values of y_2 and y_4 will increase by $\delta_0(p_{01}^{[1]} + p_{11}^{[1]})$; and the values of y_5 to y_7 will increase by $\delta_0(p_{10}^{[1]} + p_{01}^{[1]} + p_{11}^{[1]})$. One can quickly see that (25) and (26) are still satisfied with equality; both conditions in (23) and the condition in (27) are now satisfied with strict inequality; and (15) and (24) may or may not be satisfied anymore. However, since the starting point $\vec{x} = \vec{\mathbf{0}}$ satisfies both (15) and (24) with strict inequality, we can choose a sufficiently small δ_0 such that (15) and (24) are still satisfied with strict inequality after increasing the $x_0^{[1]}$ value by δ_0. This new $\vec{x}$ will be used as a new starting point for the subsequent construction.

Since $x_3^{[1]}$ participates in all 5 rank conversion inequalities (16) to (20), if we increase the value of $x_3^{[1]}$ (currently being 0) by a small constant $\delta_3 > 0$, then the values of y_1 and y_3 will increase by $\delta_3(p_{10}^{[1]} + p_{11}^{[1]})$; the values of y_2 and y_4 will increase by $\delta_3(p_{01}^{[1]} + p_{11}^{[1]})$; the value of y_5 will increase by $\delta_3(p_{10}^{[1]} + p_{01}^{[1]} + p_{11}^{[1]})$; and the values of y_6 and y_7 remain unchanged. One can quickly see that the conditions of (24) are still satisfied with strict inequality since y_6 and y_7 do not change and the starting point satisfies (24) with strict inequality; both (25) and (26) are now satisfied with strict inequality; and the conditions in (15), (23), and (27) may or may not be satisfied anymore. However, since the starting point $\vec{x}$ satisfies (15), (23), and (27) with strict inequality, we can choose a sufficiently small δ_3 such that (15), (23), and (27) are still satisfied with strict inequality after increasing its $x_3^{[1]}$

[17] Lemma 7 can be generalized for the case of $M > 1$, for which physically degradedness implies $\sum_m p_{01}^{[m]} > 0$ and $\sum_m p_{10}^{[m]} > 0$.

value by δ_3. This new $\vec{x}$ will be used as the new starting point for the subsequent construction.

Since $x_7^{[1]}$ participates in all 4 rank conversion inequalities (16) to (19), if we increase the value of $x_7^{[1]}$ (currently being 0) by a small constant $\delta_7 > 0$, then the values of y_1 and y_3 will increase by $\delta_7(p_{10}^{[1]}+p_{11}^{[1]})$; the values of y_2 and y_4 will increase by $\delta_7(p_{01}^{[1]}+p_{11}^{[1]})$; and the values of y_5 to y_7 remain unchanged. By similar reasonings as in the previous discussion, one can quickly see that the conditions of (24) to (27) are now satisfied with strict inequality; and the conditions in (15) and (23) may or may not be satisfied anymore. However, since the starting point $\vec{x}$ satisfies both (15) and (23) with strict inequality, we can choose a sufficiently small δ_7 such that (15) and (23) are still satisfied with strict inequality after increasing its $x_7^{[1]}$ value by δ_7. This new $\vec{x}$ satisfies the time-sharing inequality (15) and all rank comparison inequalities (23) to (27) with strict inequality.

The last step of the proof is to note that an interior point must have strictly positive values for all 18 coordinates. Thus far, only three coordinates $x_0^{[1]}$, $x_3^{[1]}$, and $x_7^{[1]}$ are strictly positive. Following similar arguments as in the previous discussion, we can sequentially increase the value of $x_{\mathbf{b}}^{[1]}$ for all $\mathbf{b} \in \mathsf{FTs}\backslash\{0,3,7\}$ by a sufficiently small amount $\delta_{\mathbf{b}} > 0$ while still satisfying (15) and (23)–(27) with strict inequality. The proof of Lemma 7 is thus complete.

Appendix F
Detailed Implementation of Phases 1 and 3 of The Tunneling Approach

There are multiple ways of completing the trajectory from $\mathbf{0}$ to $\vec{x}_{\text{start}}$ in Phase 1. For simplicity, we assume that $\vec{x}_{\text{start}}$ is constructed in the same way as when we construct the interior point in Appendix E.

Specifically, in Table I of Section V-E we have established that whether type-0 is non-empty depends on whether both $A_6 \subsetneq \Omega$ and $A_7 \subsetneq \Omega$ hold. Since the conditions in (24) are satisfied with strict inequality when $\vec{x} = \mathbf{0}$, we can always choose a coding vector $\mathbf{c}_t \in \mathsf{TYPE}_0$ in the beginning of transmission. Since the conditions in (24) are satisfied with strict inequality even after increasing the value of $x_0^{[1]}$ by δ_0 where δ_0 is defined in Appendix E, with close-to-one probability we can continuously choose coding vector $\mathbf{c}_t \in \mathsf{TYPE}_0$ for $n \cdot \delta_0$ number of time slots.

After choosing type-0 coding vectors, we consider type-3 coding vectors. In Table I of Section V-E, we have established that whether type-3 is non-empty depends on whether all three conditions $A_3 \subsetneq A_6$, $A_4 \subsetneq A_7$, and (41) hold with strict inequality. Since the conditions in (23) and (27) are satisfied with strict inequality before and after we increase the value of $x_3^{[1]}$ by δ_3 in Appendix E, with close-to-one probability we can continuously choose coding vector $\mathbf{c}_t \in \mathsf{TYPE}_3$ for $n \cdot \delta_3$ number of time slots. After choosing type-3 coding vectors, we consider type-7 coding vectors. In Table I of Section V-E, we have established that whether type-7 is non-empty depends on whether both $A_3 \subsetneq A_6$ and $A_4 \subsetneq A_7$ hold simultaneously. Since the conditions in (23) are satisfied with strict inequality before and after we increase the value of $x_7^{[1]}$ by δ_7 in Appendix E, with close-to-one probability we can continuously choose coding vector $\mathbf{c}_t \in \mathsf{TYPE}_7$ for $n \cdot \delta_7$ number of time slots.

After choosing type-7 coding vectors, we choose coding-type $\mathbf{b}$ sequentially for all $\mathbf{b} \in \mathsf{FTs}\backslash\{0,3,7\}$, respectively. For each $\mathbf{b}$, we will continuously choose coding vector $\mathbf{c}_t \in \mathsf{TYPE}_{\mathbf{b}}$ for $n \cdot \delta_{\mathbf{b}}$ number of time slots where the $\delta_{\mathbf{b}}$ values are discussed in Appendix E. Since during this process all 7 rank comparison inequalities (23)–(27) are satisfied with strict inequality, by the same reason as in the previous steps, continuously sending coding vector $\mathbf{c}_t \in \mathsf{TYPE}_{\mathbf{b}}$ for $n \cdot \delta_{\mathbf{b}}$ number of time slots is feasible with close-to-one probability. The above procedure completes Phase 1 of the tunneling approach.

There are multiple ways of completing the trajectory from $\vec{x}_{\text{end}}$ to $\vec{\bar{x}}$ in Phase 3. For simplicity, we quantify the penalty of *not performing Phase 3* and show that the penalty can be made negligible. We first use the $\vec{x}_{\text{end}}$ value in (45) to compute the corresponding y_1 to y_7 values using (16)–(22), and denote the computed values by $y_{i,\text{end}}$, $i = 1, \cdots, 7$. By (45) and by the fact that $\vec{\bar{x}}$ satisfies (28), we must have

$$y_{1,\text{end}} + \epsilon' > y_{3,\text{end}} \text{ and } y_{2,\text{end}} + \epsilon' > y_{4,\text{end}} \tag{120}$$

for some small $\epsilon' > 0$, which satisfies $\epsilon' \to 0$ when the ϵ in (45) approaches 0. Given the ϵ value used by the tunneling approach to compute $\vec{x}_{\text{end}}$, we can compute the corresponding ϵ'. Then we perform the encoding steps proposed in Phases 1 and 2. After finishing Phase 2, the empirical frequency $\{X_{\mathbf{b}}^{[1]}(t) : \forall \mathbf{b}\}$ equals to $\vec{x}_{\text{end}}$. By (120), we must have $\text{Rank}(S_i) + n\epsilon' > \text{Rank}(S_i \oplus \Omega_i)$ in the end of Phase 2.

By Lemma 6, it is equivalent to $\text{Rank}(S_i \cap \Omega_i) > n(R_i - \epsilon')$ for all $i \in \{1, 2\}$. That is, in the end of Phase 2, destination d_i can obtain $n(R_i - \epsilon')$ independent linear combinations of the message symbols $X_{i,1}$ to X_{i,nR_i}. Namely, each d_i knows a large percentage of the independent linear combinations of its desired symbols $\mathbf{X}_i$. However, each d_i needs to know *all* nR_i independent linear combinations before decoding. As a result, source s simply needs to send the *missing* linear combinations to each d_i, respectively. We observe that even when not performing any inter-flow coding, the duration of sending the missing linear combinations is $\mathcal{O}(\epsilon' n)$. As a result, the penalty of not performing any inter-flow coding in Phase 3 is at most $\mathcal{O}(\epsilon')$ from the long-term average throughput perspective.

Appendix G
Detailed Derivation of Table I

Before proving the statements in Table I we first introduce the following lemma

Lemma 9: Consider any three linear subspaces A, B, and C in the global space Ω. If $B \supseteq C$, then $(A \cap B) \oplus C = (A \oplus C) \cap B$.

Proof: By simple set operations, we quickly have $(A \cap B) \oplus C \subseteq (A \oplus C) \cap (B \oplus C) = (A \oplus C) \cap B$.

As a result, we only need to prove that any vector $\mathbf{v} \in (A \oplus C) \cap B$ must also be in $(A \cap B) \oplus C$. Consider any

$\mathbf{v} \in (A \oplus C) \cap B$. By definition, we can write

$$\mathbf{v} = \mathbf{v}_A + \mathbf{v}_C = \mathbf{v}_B$$

for some $\mathbf{v}_A \in A$, $\mathbf{v}_B \in B$, and $\mathbf{v}_C \in C$. As a result, $\mathbf{v}_A = \mathbf{v}_B - \mathbf{v}_C$ must also belong to B since $C \subseteq B$. As a result, $\mathbf{v}_A \in A \cap B$. Since $\mathbf{v} = \mathbf{v}_A + \mathbf{v}_C$, we must have $\mathbf{v} \in (A \cap B) \oplus C$. The proof is complete. ■

In the following, we consider all 18 different $\mathbf{b} \in \mathsf{FTs}$ one by one.

• Case 1: $\mathbf{b} = 0$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_0 &= \mathsf{TYPE}_{0000000}\\ &\triangleq \Omega \backslash (A_1 \cup A_2 \cup \cdots \cup A_7)\\ &= \Omega \backslash ((S_1 \oplus S_2 \oplus \Omega_1) \cup (S_1 \oplus S_2 \oplus \Omega_2)).\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_0 \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(\Omega) - \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_1) > 0\\ \text{and } &\mathrm{Rank}(\Omega) - \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2) > 0.\end{aligned}$$

Also see our discussion of $\mathbf{b} = 23$ in Section V-E (around eqs. (46) to (50)). The above conditions are equivalent to $A_6 \subsetneq \Omega$ and $A_7 \subsetneq \Omega$, respectively. The entry for $\mathbf{b} = 0$ in Table I is thus proven.

• Case 2: $\mathbf{b} = 1$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_1 &= \mathsf{TYPE}_{0000001}\\ &\triangleq A_7 \backslash (A_1 \cup A_2 \cup \cdots \cup A_6)\\ &= (S_1 \oplus S_2 \oplus \Omega_2) \backslash ((S_2 \oplus \Omega_2) \cup (S_1 \oplus S_2 \oplus \Omega_1)).\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_1 \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2)\\ &\quad - \mathrm{Rank}((S_2 \oplus \Omega_2) \cap (S_1 \oplus S_2 \oplus \Omega_2))\\ &= \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2) - \mathrm{Rank}(S_2 \oplus \Omega_2) \qquad (121)\\ &= \mathrm{Rank}(A_7) - \mathrm{Rank}(A_4) > 0\end{aligned}$$

and

$$\begin{aligned}&\mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_2)\\ &\quad - \mathrm{Rank}((S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2))\\ &= \mathrm{Rank}(\Omega) - \mathrm{Rank}(S_1 \oplus S_2 \oplus \Omega_1) \qquad (122)\\ &= \mathrm{Rank}(\Omega) - \mathrm{Rank}(A_6) > 0\end{aligned}$$

where (121) follows from $(A_4 \cap A_7) = A_4$; and (122) follows from Lemma 6. The above conditions are equivalent to $A_4 \subsetneq A_7$ and $A_6 \subsetneq \Omega$, respectively. The entry for $\mathbf{b} = 1$ in Table I is thus proven.

• Case 3: $\mathbf{b} = 2$. This is a symmetric case of $\mathbf{b} = 1$.

• Case 4: $\mathbf{b} = 3$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_3 &= \mathsf{TYPE}_{0000011}\\ &\triangleq (A_6 \cap A_7) \backslash (A_1 \cup A_2 \cup \cdots \cup A_5)\\ &= ((S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2)) \backslash\\ &\qquad ((S_1 \oplus \Omega_1) \cup (S_2 \oplus \Omega_2) \cup (S_1 \oplus S_2)).\end{aligned}$$

We notice that the expression of TYPE_3 has the form of $B_1 \backslash (B_2 \cup B_3 \cup B_4)$ where B_1 to B_4 are linear subspaces. If we define $B'_i = B_1 \cap B_i$ where $i = 2$ to 4, we can see that TYPE_3 can be expressed as $B_1 \backslash (B'_2 \cup B'_3 \cup B'_4)$ where B_1 and B'_2 to B'_4 are linear subspaces. Note that the size of B_1 is $q^{\mathrm{Rank}(B_1)}$. By the union bound and by the observation that each linear subspace always contains the $\mathbf{0}$ vector, the size of $(B'_2 \cup B'_3 \cup B'_4)$ is no larger than $q^{\mathrm{Rank}(B'_2)} + q^{\mathrm{Rank}(B'_3)} + q^{\mathrm{Rank}(B'_4)} - 2$.

As a result, assuming $q \geq 3$ the statement $\mathsf{TYPE}_3 \neq \emptyset$ holds if and only if $\mathrm{Rank}(B_1) > \mathrm{Rank}(B'_i)$ for $i = 2$ to 4. Plug in the expressions of B_1 and B'_2 to B'_4, the statement $\mathsf{TYPE}_3 \neq \emptyset$ holds if and only if the following inequality are true.

$$\begin{aligned}&\mathrm{Rank}((S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2))\\ &\quad - \mathrm{Rank}((S_1 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2)) \qquad (123)\\ &= (\mathrm{Rank}(A_6) + \mathrm{Rank}(A_7) - \mathrm{Rank}(\Omega))\\ &\quad - (\mathrm{Rank}(A_3) + \mathrm{Rank}(A_7) - \mathrm{Rank}(\Omega)) \qquad (124)\\ &= \mathrm{Rank}(A_6) - \mathrm{Rank}(A_3) > 0,\end{aligned}$$

$$\begin{aligned}&\mathrm{Rank}((S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2))\\ &\quad - \mathrm{Rank}((S_2 \oplus \Omega_2) \cap (S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2))\\ &= \mathrm{Rank}(A_7) - \mathrm{Rank}(A_4) > 0, \qquad (125)\end{aligned}$$

and

$$\begin{aligned}&\mathrm{Rank}((S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2)) -\\ &\quad \mathrm{Rank}((S_1 \oplus S_2) \cap (S_1 \oplus S_2 \oplus \Omega_1) \cap (S_1 \oplus S_2 \oplus \Omega_2))\\ &= (\mathrm{Rank}(A_6) + \mathrm{Rank}(A_7) - \mathrm{Rank}(\Omega)) - \mathrm{Rank}(A_5) > 0, \qquad (126)\end{aligned}$$

where (124) follows from $(A_3 \cap A_6 \cap A_7) = (A_3 \cap A_7)$ and from Lemma 6; (125) is symmetric to (123); and (126) follows from $(A_5 \cap A_6 \cap A_7) = A_5$ and from Lemma 6. The above conditions are equivalent to $A_3 \subsetneq A_6$, $A_4 \subsetneq A_7$, and strict (41), respectively. The entry for $\mathbf{b} = 3$ in Table I is thus proven.

• Case 5: $\mathbf{b} = 7$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_7 &= \mathsf{TYPE}_{0000111}\\ &\triangleq (A_5 \cap A_6 \cap A_7) \backslash (A_1 \cup A_2 \cup A_3 \cup A_4)\\ &= (S_1 \oplus S_2) \backslash ((S_1 \oplus \Omega_1) \cup (S_2 \oplus \Omega_2)).\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_7 \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(A_5) - \mathrm{Rank}(A_3 \cap A_5)\\ &= \mathrm{Rank}(A_5) - (\mathrm{Rank}(A_3) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_6)) \qquad (127)\\ &= \mathrm{Rank}(A_6) - \mathrm{Rank}(A_3) > 0,\end{aligned}$$

and

$$\begin{aligned}&\mathrm{Rank}(A_5) - \mathrm{Rank}(A_4 \cap A_5)\\ &= \mathrm{Rank}(A_5) - (\mathrm{Rank}(A_4) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_7)) \qquad (128)\\ &= \mathrm{Rank}(A_7) - \mathrm{Rank}(A_4) > 0,\end{aligned}$$

where (127) and (128) follow from Lemma 6. The above conditions are equivalent to $A_3 \subsetneq A_6$ and $A_4 \subsetneq A_7$, respectively. The entry for $\mathbf{b} = 7$ in Table I is thus proven.

• Case 6: $\mathbf{b} = 9$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_9 &= \mathsf{TYPE}_{0001001}\\ &\triangleq (A_4 \cap A_7)\backslash(A_1 \cup A_2 \cup A_3 \cup A_5 \cup A_6)\\ &= A_4\backslash A_6.\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_9 \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(A_4) - \mathrm{Rank}(A_4 \cap A_6)\\ &= \mathrm{Rank}(A_4) - (\mathrm{Rank}(A_4) + \mathrm{Rank}(A_6) - \mathrm{Rank}(\Omega)) \quad (129)\\ &= \mathrm{Rank}(\Omega) - \mathrm{Rank}(A_6) > 0,\end{aligned}$$

where (129) follows from Lemma 6. The above condition is equivalent to $A_6 \subsetneq \Omega$. The entry for $\mathbf{b} = 9$ in Table I is thus proven.

• Case 7: $\mathbf{b} = 11$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_{11} &= \mathsf{TYPE}_{0001011}\\ &\triangleq (A_4 \cap A_6 \cap A_7)\backslash(A_1 \cup A_2 \cup A_3 \cup A_5)\\ &= (A_4 \cap A_6)\backslash(A_3 \cup A_5).\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_{11} \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(A_4 \cap A_6) - \mathrm{Rank}(A_3 \cap A_4 \cap A_6)\\ &= \mathrm{Rank}(A_4 \cap A_6) - \mathrm{Rank}(A_3 \cap A_4)\\ &= (\mathrm{Rank}(A_4) + \mathrm{Rank}(A_6) - \mathrm{Rank}(\Omega)) -\\ &\quad (\mathrm{Rank}(A_3) + \mathrm{Rank}(A_4) - \mathrm{Rank}(\Omega)) \quad (130)\\ &= \mathrm{Rank}(A_6) - \mathrm{Rank}(A_3) > 0,\end{aligned}$$

and

$$\begin{aligned}&\mathrm{Rank}(A_4 \cap A_6) - \mathrm{Rank}(A_5 \cap A_4 \cap A_6)\\ &= \mathrm{Rank}(A_4 \cap A_6) - \mathrm{Rank}(A_5 \cap A_4)\\ &= (\mathrm{Rank}(A_4) + \mathrm{Rank}(A_6) - \mathrm{Rank}(\Omega)) -\\ &\quad (\mathrm{Rank}(A_5) + \mathrm{Rank}(A_4) - \mathrm{Rank}(A_7)) \quad (131)\\ &= \mathrm{Rank}(A_6) + \mathrm{Rank}(A_7) - \mathrm{Rank}(A_5) - \mathrm{Rank}(\Omega) > 0,\end{aligned}$$

where (130) and (131) follow from Lemma 6. The above conditions are equivalent to $A_3 \subsetneq A_6$ and strict (41), respectively. The entry for $\mathbf{b} = 11$ in Table I is thus proven.

• Case 8: $\mathbf{b} = 15$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_{15} &= \mathsf{TYPE}_{0001111}\\ &\triangleq (A_4 \cap A_5 \cap A_6 \cap A_7)\backslash(A_1 \cup A_2 \cup A_3)\\ &= (A_4 \cap A_5)\backslash(A_2 \cup A_3).\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_{15} \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(A_4 \cap A_5) - \mathrm{Rank}(A_2 \cap A_4 \cap A_5)\\ &= \mathrm{Rank}(A_4 \cap A_5) - \mathrm{Rank}(A_2)\\ &= (\mathrm{Rank}(A_4) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_7)) - \mathrm{Rank}(A_2) > 0, \quad (132)\end{aligned}$$

and

$$\begin{aligned}&\mathrm{Rank}(A_4 \cap A_5) - \mathrm{Rank}(A_3 \cap A_4 \cap A_5) \quad (133)\\ &= \mathrm{Rank}(A_3 \oplus (A_4 \cap A_5)) - \mathrm{Rank}(A_3) \quad (134)\\ &= \mathrm{Rank}(\Omega_1 \oplus (S_1 \oplus ((S_2 \oplus \Omega_2) \cap (S_1 \oplus S_2))))\\ &\quad - \mathrm{Rank}(A_3)\\ &= \mathrm{Rank}(\Omega_1 \oplus ((S_1 \oplus S_2 \oplus \Omega_2) \cap (S_1 \oplus S_2)))\\ &\quad - \mathrm{Rank}(A_3) \quad (135)\\ &= \mathrm{Rank}(\Omega_1 \oplus S_1 \oplus S_2) - \mathrm{Rank}(A_3) > 0, \quad (136)\end{aligned}$$

where (132) and (134) follow from Lemma 6; and (135) follows from Lemma 9. The above conditions are equivalent to strict (40) and $A_3 \subsetneq A_6$, respectively. The entry for $\mathbf{b} = 15$ in Table I is thus proven.

• Case 9: $\mathbf{b} = 18$. This is a symmetric case of $\mathbf{b} = 9$.

• Case 10: $\mathbf{b} = 19$. This is a symmetric case of $\mathbf{b} = 11$.

• Case 11: $\mathbf{b} = 23$. This is a symmetric case of $\mathbf{b} = 15$.

• Case 12: $\mathbf{b} = 27$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_{27} &= \mathsf{TYPE}_{0011011}\\ &\triangleq (A_3 \cap A_4 \cap A_6 \cap A_7)\backslash(A_1 \cup A_2 \cup A_5)\\ &= (A_3 \cap A_4)\backslash A_5.\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_{27} \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(A_3 \cap A_4) - \mathrm{Rank}(A_3 \cap A_4 \cap A_5)\\ &= \mathrm{Rank}(A_3 \cap A_4)\\ &\quad - (\mathrm{Rank}(A_4 \cap A_5) + \mathrm{Rank}(A_3) - \mathrm{Rank}(A_6)) \quad (137)\\ &= (\mathrm{Rank}(A_3) + \mathrm{Rank}(A_4) - \mathrm{Rank}(\Omega))\\ &\quad - ((\mathrm{Rank}(A_4) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_7))\\ &\quad + \mathrm{Rank}(A_3) - \mathrm{Rank}(A_6)) \quad (138)\\ &= \mathrm{Rank}(A_6) + \mathrm{Rank}(A_7) - \mathrm{Rank}(\Omega) - \mathrm{Rank}(A_5) > 0\end{aligned}$$

where (137) follows from the equivalence between (133) and (136); and (138) follows from Lemma 6. The above condition is equivalent to strict (41). The entry for $\mathbf{b} = 27$ in Table I is thus proven.

• Case 13: $\mathbf{b} = 31$. By definition, we have

$$\begin{aligned}\mathsf{TYPE}_{31} &= \mathsf{TYPE}_{0011111}\\ &\triangleq (A_3 \cap \cdots \cap A_7)\backslash(A_1 \cup A_2)\\ &= (A_3 \cap A_4 \cap A_5)\backslash(A_1 \cup A_2).\end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_{31} \neq \emptyset$ is equivalent to

$$\begin{aligned}&\mathrm{Rank}(A_3 \cap A_4 \cap A_5) - \mathrm{Rank}(A_1 \cap A_3 \cap A_4 \cap A_5) \quad (139)\\ &= \mathrm{Rank}(A_3 \cap A_4 \cap A_5) - \mathrm{Rank}(A_1 \cap A_4)\\ &= (\mathrm{Rank}(A_3) + \mathrm{Rank}(A_4) + \mathrm{Rank}(A_5)\\ &\quad - \mathrm{Rank}(A_6) - \mathrm{Rank}(A_7))\\ &\quad - (\mathrm{Rank}(A_1) + \mathrm{Rank}(A_4) - \mathrm{Rank}(A_7)) \quad (140)\\ &= \mathrm{Rank}(A_3) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_6) - \mathrm{Rank}(A_1) > 0,\end{aligned}$$

and

$$\begin{aligned}&\mathrm{Rank}(A_3 \cap A_4 \cap A_5) - \mathrm{Rank}(A_2 \cap A_3 \cap A_4 \cap A_5)\\ &= \mathrm{Rank}(A_4) + \mathrm{Rank}(A_5) - \mathrm{Rank}(A_7) - \mathrm{Rank}(A_2) > 0, \quad (141)\end{aligned}$$

where the first term of (140) follows from the same arguments as used in (137) and (138) that quantify $\text{Rank}(A_3 \cap A_4 \cap A_5)$; the second term of (140) follows from Lemma 6; and (141) is a symmetric version of (139). The above conditions are equivalent to strict (39) and strict (40), respectively. The entry for $\mathbf{b} = 31$ in Table I is thus proven.

• Case 14: $\mathbf{b} = 47$. By definition, we have

$$\begin{aligned} \mathsf{TYPE}_{47} &= \mathsf{TYPE}_{0101111} \\ &\triangleq (A_2 \cap A_4 \cap A_5 \cap A_6 \cap A_7) \backslash (A_1 \cup A_3) \\ &= A_2 \backslash A_3. \end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_{47} \neq \emptyset$ is equivalent to

$$\begin{aligned} &\text{Rank}(A_2) - \text{Rank}(A_3 \cap A_2) \\ &= \text{Rank}(A_3 \oplus A_2) - \text{Rank}(A_3) \qquad (142) \\ &= \text{Rank}(A_6) - \text{Rank}(A_3) > 0, \end{aligned}$$

where (142) follows from Lemma 6. The above condition is equivalent to $A_3 \subsetneq A_6$. The entry for $\mathbf{b} = 47$ in Table I is thus proven.

• Case 15: $\mathbf{b} = 63$. By definition, we have

$$\begin{aligned} \mathsf{TYPE}_{63} &= \mathsf{TYPE}_{0111111} \\ &\triangleq (A_2 \cap \cdots \cap A_7) \backslash A_1 \\ &= (A_2 \cap A_3) \backslash A_1. \end{aligned}$$

Since $q \geq 2$, the statement $\mathsf{TYPE}_{63} \neq \emptyset$ is equivalent to

$$\begin{aligned} &\text{Rank}(A_2 \cap A_3) - \text{Rank}(A_1 \cap A_2 \cap A_3) \\ &= \text{Rank}(A_2 \cap A_3) - \text{Rank}(A_1 \cap A_2) \\ &= (\text{Rank}(A_2) + \text{Rank}(A_3) - \text{Rank}(A_6)) \\ &\quad - (\text{Rank}(A_1) + \text{Rank}(A_2) - \text{Rank}(A_5)) \qquad (143) \\ &= \text{Rank}(A_3) + \text{Rank}(A_5) - \text{Rank}(A_6) - \text{Rank}(A_1) > 0, \end{aligned}$$

where (143) follows from Lemma 6. The above condition is equivalent to strict (39). The entry for $\mathbf{b} = 63$ in Table I is thus proven.

• Case 16: $\mathbf{b} = 87$. This is a symmetric case of $\mathbf{b} = 47$.

• Case 17: $\mathbf{b} = 95$. This is a symmetric case of $\mathbf{b} = 63$.

• Case 18: $\mathbf{b} = 127$. By definition, we have

$$\begin{aligned} \mathsf{TYPE}_{127} &= \mathsf{TYPE}_{1111111} \\ &\triangleq (A_1 \cap \cdots \cap A_7) = A_1 \cap A_2. \end{aligned}$$

Since A_1 and A_2 are linear subspaces, so is $(A_1 \cap A_2)$. Since any linear subspace always contains the $\mathbf{0}$ vector, $\mathsf{TYPE}_{127} \neq \emptyset$ is always true. The entry for $\mathbf{b} = 127$ in Table I is thus proven.

APPENDIX H
A LEMMA FOR THE ACHIEVABILITY PROOF FOR THE CASE OF ARBITRARY M

Lemma 10: Consider any set $\mathcal{H} \subseteq \{1, 2, 3, \cdots, 7\}$ and any $m \in \{1, \cdots, M\}$. Define $\mathsf{term} \triangleq \text{Rank}\left(\bigcap_{h \in \mathcal{H}} A_h\right)$ where the linear subspaces A_h, $h \in \mathcal{H}$, are evaluated in the beginning of time t (or equivalently the end of the split time instant $(t-1).M$). Also define $\mathsf{term}^\dagger \triangleq \text{Rank}\left(\bigcap_{h \in \mathcal{H}} A_h^\dagger\right)$ where the linear subspaces $A_h^\dagger$, $h \in \mathcal{H}$, are evaluated in the beginning of the split time instant $t.m$. Then we must have

$$0 \leq \mathsf{term}^\dagger - \mathsf{term} \leq |\mathcal{H}| \cdot (m-1).$$

Proof: To prove $0 \leq (\mathsf{term}^\dagger - \mathsf{term})$, we prove the following stronger statement instead.

$$\left(\bigcap_{h \in \mathcal{H}} A_h\right) \subseteq \left(\bigcap_{h \in \mathcal{H}} A_h^\dagger\right). \qquad (144)$$

Consider first the case of $|\mathcal{H}| = 1$, i.e., $\mathcal{H} = \{h\}$ for some $h \in \{1, \cdots, 7\}$. We first note that the knowledge spaces S_1 and S_2 enlarge monotonically over time. Therefore, the linear subspace A_h, being a sum space of monotonically growing S_i, $i = 1, 2$, and constant linear spaces Ω_j, $j = 1, 2$, also enlarges monotonically. As a result, $A_h \subseteq A_h^\dagger$ and (144) is proven for the case of $|\mathcal{H}| = 1$. Since any intersection of monotonically growing linear subspaces is also monotonically growing, (144) is proven for the case of arbitrary $\mathcal{H}$ as well.

To prove $(\mathsf{term}^\dagger - \mathsf{term}) \leq |\mathcal{H}| \cdot (m-1)$, we again first consider the simplest case of $|\mathcal{H}| = 1$. For any $h \in \{1, \cdots, 7\}$, we would like to prove that $\text{Rank}(A_h^\dagger) - \text{Rank}(A_h) \leq (m-1)$. Consider the case of $A_6 = (S_1 \oplus S_2 \oplus \Omega_1)$ for example. Let S_1 and S_2 denote the knowledge spaces of d_1 and d_2 in the beginning of time t. Let $\mathbf{c}_{t,1}$ to $\mathbf{c}_{t,m-1}$ denote the coding vector of the first $(m-1)$ inputs transmitted before the split time instant $t.m$. As a result,

$$\begin{aligned} A_6^\dagger = &(S_1 \oplus \mathsf{span}\{\text{those } \mathbf{c}_{t,1} \text{ to } \mathbf{c}_{t,m-1} \text{ rec'd by } d_1\}) \\ &\oplus (S_2 \oplus \mathsf{span}\{\text{those } \mathbf{c}_{t,1} \text{ to } \mathbf{c}_{t,m-1} \text{ rec'd by } d_2\}) \\ &\oplus \Omega_1 \\ = &A_6 \oplus \mathsf{span}\{\text{those } \mathbf{c}_{t,1} \text{ to } \mathbf{c}_{t,m-1} \text{ rec'd by } d_1 \text{ or } d_2\}. \end{aligned}$$

As a result, $\text{Rank}(A_6^\dagger) - \text{Rank}(A_6) \leq (m-1)$. The same approach can be used to prove for the cases of A_1 to A_7. We have thus proven $\mathsf{term}^\dagger - \mathsf{term} \leq |\mathcal{H}| \cdot (m-1)$ for the simplest case of $|\mathcal{H}| = 1$.

When proving the case of general $\mathcal{H}$, we note that for any linear subspaces $A \subseteq B$ and $C \subseteq D$, by Lemma 6 we have

$$\begin{aligned} &\text{Rank}(B \cap D) - \text{Rank}(A \cap C) \\ &= \text{Rank}(B) + \text{Rank}(D) - \text{Rank}(B \oplus D) \\ &\qquad - (\text{Rank}(A) + \text{Rank}(C) - \text{Rank}(A \oplus C)) \\ &\leq (\text{Rank}(B) - \text{Rank}(A)) + (\text{Rank}(D) - \text{Rank}(C)) \end{aligned} \qquad (145)$$

where (145) follows from the fact that $(B \oplus D) \supseteq (A \oplus C)$. By choosing $A = \bigcap_{h \in \mathcal{H}_1} A_h$; $B = \bigcap_{h \in \mathcal{H}_1} A_h^\dagger$; $C = \bigcap_{h \in \mathcal{H}_2} A_h$; and $D = \bigcap_{h \in \mathcal{H}_2} A_h^\dagger$ with $\mathcal{H}_1 \cup \mathcal{H}_2 = \mathcal{H}$ and $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$, we can thus prove by induction that $(\mathsf{term}^\dagger - \mathsf{term}) \leq |\mathcal{H}| \cdot (m-1)$ for general $\mathcal{H}$ (with the help of (144) implicitly). The proof of this lemma is complete. ■

REFERENCES

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

[2] T. M. Cover, "Comments on broadcast channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2524–2530, Oct. 1998.

[3] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 789–804, March 2006.
[4] A. Dana and B. Hassibi, "The capacity region of multiple input erasure broadcast channels," in *Proc. IEEE Int'l Symp. Inform. Theory*. Adelaide, Australia, September 2005.
[5] A. El Gamal, "The feedback capacity of degraded broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 379–381, March 1978.
[6] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Multiuser broadcast erasure channel with feedback — capacity and algorithms," in *Proc. NetCoop*, 2010.
[7] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback — capacity and algorithms," in *Proc. 5th Workshop on Network Coding, Theory, & Applications (NetCod)*. Lausanne, Switzerland, June 2009, pp. 54–61.
[8] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.
[9] D. Koutsonikolas, C.-C. Wang, Y. Hu, and N. Shroff, "FEC-based AP downlink transmission schemes for multiple flows: Combining the reliability and throughput enhancement of intra- and inter-flow coding," *Elsevier Performance Evaluation (PEVA)*, vol. 68, no. 11, November 2011.
[10] P. Larsson and N. Johansson, "Multi-user ARQ," in *Proc. IEEE Vehicular Technology Conference*, 2006.
[11] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, February 2003.
[12] X. Li, C.-C. Wang, and X. Lin, "Optimal immediately-decodable inter-session network coding (IDNC) schemes for two unicast sessions with hard deadline constraints," in *Proc. 49th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, Illinois, USA, September 2011, pp. 784–791.
[13] L. Ozarow and S. Leung-Yan-Cheong, "An achievable region and outer bound for the Gaussian broadcast channel with feedback," *IEEE Trans. Inf. Theory*, vol. 30, no. 4, July 1984.
[14] H. Sato, "An outer bound to the capacity region of broadcast channels," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 374–377, May 1978.
[15] C.-C. Wang, "Capacity of 1-to-K broadcast packet erasure channels with channel output feedback," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 957–988, February 2012.