On The Capacity of 1-to-K Broadcast Packet Erasure Channels with Channel Output Feedback

Chih-Chun Wang *Member, IEEE,*

***Abstract*—This paper focuses on the 1-to-K broadcast packet erasure channel (PEC), a generalization of the broadcast binary erasure channel from the binary symbol to a finite field $\mathsf{GF}(q)$ with sufficiently large q. We consider the setting in which the source node has instant feedback of the channel outputs of the K receivers after each transmission. The main results of this paper are: (i) The capacity region for general 1-to-3 broadcast PECs, and (ii) The capacity region for two types of 1-to-K broadcast PECs: the symmetric PECs, and the spatially independent PECs with one-sided fairness constraints. This paper also develops (iii) A pair of outer and inner bounds of the capacity region for arbitrary 1-to-K broadcast PECs, which can be easily evaluated by any linear programming solver. The proposed inner bound is proven by a new class of intersession network coding schemes, termed the *packet evolution* schemes, which is based on the concept of *code alignment* in $\mathsf{GF}(q)$ that is in parallel with the interference alignment techniques for the Euclidean space. Extensive numerical experiments show that the outer and inner bounds meet for almost all broadcast PECs encountered in practical scenarios and thus effectively bracket the capacity of general 1-to-K broadcast PECs with COF.**

***Index Terms*—Broadcast capacity, channel output feedback, network coding, network code alignment, packet erasure channels.**

I. Introduction

Broadcast channels have been actively studied since the inception of network information theory. Although the broadcast capacity region remains unknown for general channel models, significant progress has been made in various sub-directions (see [4] for a tutorial paper), including but not limited to the degraded broadcast channel models [1], the 2-user capacity with degraded message sets [13] or with message side information [27]. Motivated by wireless broadcast communications, the Gaussian broadcast channel (GBC) [26] is among the most widely studied broadcast channel models.

In the last decade, the concept of network coding has emerged [16], which focuses on achieving the capacity of a communication network. More explicitly, the network-coding-based approaches generally model each hop of a packet-based communication network by a *packet erasure channel* (PEC) instead of the classic Gaussian channel. Such simple abstraction allows us to explore the information-theoretic capacity of a much larger network with mathematical rigor and also sheds new insights on the network effects of a communication system. One such example is that when all destinations are interested in the same set of packets, the capacity of any arbitrarily large, multi-hop PEC network can be characterized by the corresponding min-cut/max-flow values [5], [16]. Another example is the broadcast channel capacity with message side information. Unlike the existing GBC-based results that are limited to the simplest 2-user scenario [27], the capacity region for 1-to-K broadcast PECs with message side information has been derived for $K = 3$ and tightly bounded for general K values [23], [24].[1] In addition to providing new insights on network communications, this simple PEC-based abstraction of network coding also accelerates the transition from theory to practice. Many capacity-achieving *network codes* [10] have since been implemented for either the wireline [3] or the wireless multi-hop networks [12], [14].

Motivated by the state-of-the-art wireless network coding protocols and the corresponding applications, this paper studies the memoryless 1-to-K broadcast PEC with Channel Output Feedback (COF). Namely, a single source node sends out a stream of packets wirelessly, which carries information of K independent downlink data sessions, one for each receiver d_k, $k = 1, \cdots, K$, respectively. Due to the randomness of the underlying wireless channel, each transmitted packet may or may not be heard by a receiver d_k. After packet transmission, each d_k then informs the source its own channel output by sending back the ACKnowledgement (ACK) packets periodically (batch feedback) or after each time slot (per-packet instant feedback) [28]. [9] derives the capacity region of the memoryless 1-to-2 broadcast PEC with COF. The results show that COF strictly improves the capacity of the memoryless 1-to-2 broadcast PEC, which is in sharp contrast with the classic result that feedback does not increase the capacity for any memoryless 1-to-1 channel. [9] can also be viewed as a mirroring result to the GBCs with COF [19]. It is worth noting that other than increasing the achievable throughput, COF can also be used for queue and delay management [17], [22], for rate-control in a wireless network coded system [14], and for complexity reduction of network code design [11], [21]. For example, it is proven in [21] that with feedback, random *non-coding* transmission can achieve the capacity when there is only one source and destination pair in the network. On the other hand, in spite of the closed-form capacity expression for the case of one source/destinatin pair, the capacity characterization problem becomes much more

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

Part of this paper was presented in the 48th Annual Allerton Conference on Communication, Control, and Computing, 2010.

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

Manuscript received May 17, 2010; revised July ??, 2011.

[1]The results of 1-to-K broadcast PECs with message side information [23], [24] is related to the capacity of the "XOR-in-the-air" scheme [12] in a wireless network.

challenging when there are multiple coexisting data sessions in the network, also known as the inter-session coding problem [25]. The broadcast capacity problem studied in this paper is one instance of the inter-session coding problem.

The main contribution of this work includes: (i) The capacity region for general 1-to-3 broadcast PECs with COF; (ii) The capacity region for two types of 1-to-K broadcast PECs with COF: the *symmetric* PECs, and the *spatially independent* PECs with *one-sided fairness constraints*; and (iii) A pair of outer and inner bounds of the capacity region for general 1-to-K broadcast PECs with COF, which can be easily evaluated by any linear programming solver. Extensive numerical experiments show that the outer and inner bounds meet for almost all broadcast PECs encountered in practical scenarios and thus effectively bracket the capacity region.

The capacity outer bound in this paper is derived by generalizing the degraded channel argument first proposed in [19], and later used by [8], [9]. For the achievability part of (i), (ii), and (iii), we devise a new class of inter-session network coding schemes, termed the *packet evolution method*. The packet evolution method is based on a novel concept of *network code alignment*, which is the PEC-counterpart of the interference alignment method originally proposed for Gaussian interference channels [2], [6] and Gaussian MIMO channels with transmitter channel state information [18]. In parallel with the packet evolution scheme proposed in this work, [8] independently designs different queue-based network control/coding schemes that are also capacity-achieving for arbitrary 1-to-3 PECs and for symmetric 1-to-K PECs.[2]

The rest of this paper is organized as follows. Section II contains the basic setting as well as the detailed comparison to the existing results in [9], [15], [20] via an illustrating example. Section III describes the main theorems of this paper and the proof of the converse theorem. In particular, Section III-A focuses on the capacity results for arbitrary broadcast PEC parameters while Section III-B considers two special types of broadcast PECs: the symmetric and the spatially independent PECs, respectively. Section IV introduces a new class of network coding schemes, termed the packet evolution (PE) method. Based on the PE method, Section V proves the achievability results in Section III. Some theoretic implications and discussions are included in Section VI. Section VII concludes this paper.

II. Problem Setting & Existing Results

A. The Memoryless 1-to-K Broadcast Packet Erasure Channel

For any positive integer K, we use $[K] \triangleq \{1, 2, \cdots, K\}$ to denote the set of integers from 1 to K, and use $2^{[K]}$ to denote the collection of all subsets of $[K]$. For any $S_1, S_2 \in 2^{[K]}$, we use the notation $S_1 \not\subseteq S_2$ to denote $S_1 \neq (S_1 \cap S_2)$, i.e., S_1 is not a subset of S_2.

Consider a 1-to-K broadcast PEC from a single source s to K destinations d_k, $k \in [K]$. For each channel usage, the 1-to-K broadcast PEC takes an input symbol $Y \in \mathsf{GF}(q)$ from s and outputs a K-dimensional vector $\mathbf{Z} \triangleq (Z_1, \cdots, Z_K) \in (\{Y\} \cup \{*\})^K$, where the k-th coordinate Z_k being "$*$" denotes that the transmitted symbol Y does not reach the k-th receiver d_k (thus being erased). We also assume that there is no other type of noise, i.e., the individual output is either equal to the input Y or an erasure "$*$." The *success probabilities* of a 1-to-K PEC are described by 2^K non-negative parameters: $p_{S\overline{[K]\backslash S}}$ for all $S \in 2^{[K]}$ such that $\sum_{S \in 2^{[K]}} p_{S\overline{[K]\backslash S}} = 1$ and for all $y \in \mathsf{GF}(q)$,

$$\mathsf{Prob}\left(\{k \in [K] : Z_k = y\} = S | \, Y = y\right) = p_{S\overline{[K]\backslash S}}.$$

That is, $p_{S\overline{[K]\backslash S}}$ denotes the probability that the transmitted symbol Y is received *by and only by* the receivers $\{d_k : k \in S\}$. In addition, the following notation will be used frequently in this work: For all $S \in 2^{[K]}$, we define

$$p_{\cup S} = \sum_{\forall T \in 2^{[K]} : T \cap S \neq \emptyset} p_{T\overline{[K]\backslash T}}. \tag{1}$$

That is, $p_{\cup S}$ is the probability that *at least one of the receivers d_k in S* successfully receives the transmitted symbol Y. For example, when $K = 2$,

$$p_{\cup\{1,2\}} = p_{\{1\}\overline{\{2\}}} + p_{\{2\}\overline{\{1\}}} + p_{\{1,2\}\overline{\emptyset}}$$

is the probability that at least one of d_1 and d_2 receives the transmitted symbol Y. We sometimes use p_k as shorthand for $p_{\cup\{k\}}$, which is the marginal probability that the k-th receiver d_k receives Y successfully.

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

$$\mathsf{Prob}\left(\forall t \in [n], \{k : Z_k(t) = y(t)\} = S(t) \right. \\ \left. |\forall t \in [n], Y(t) = y(t)\right) = \prod_{t=1}^{n} p_{S(t)\overline{[K]\backslash S(t)}}.$$

The above setting allows the success events among different receivers to be dependent, also defined as *spatial dependence*. For example, when two logical receivers d_{k_1} and d_{k_2} are situated in the same physical node, we simply set the $p_{S\overline{[K]\backslash S}}$ parameters to allow perfect correlation between the success events of d_{k_1} and d_{k_2}. Throughout this paper, we consider memoryless 1-to-K broadcast PECs that may or may not be spatially dependent.

B. Broadcast PEC Capacity with Channel Output Feedback

We consider the following broadcast scenario from s to $\{d_k : k \in [K]\}$. Assume slotted transmission. Source s is allowed to use the 1-to-K PEC exactly n times and would like to carry information for K independent downlink data sessions, one for each d_k, respectively. For each $k \in [K]$, the k-th session (from s to d_k) contains nR_k information symbols $\mathbf{X}_k \triangleq \{X_{k,j} \in \mathsf{GF}(q), j \in [nR_k]\}$, where R_k is the data rate for the (s, d_k) session. All the information symbols $X_{k,j}$ for all $k \in [K]$ and $j \in [nR_k]$ are independently and uniformly distributed in $\mathsf{GF}(q)$.

[2]In contrast with the packet evolution schemes used in the proof of Propositions 3 and 5, it is an open problem how to adopt and analyze the schemes in [8] for non-symmetric 1-to-K PECs with $K \geq 4$.

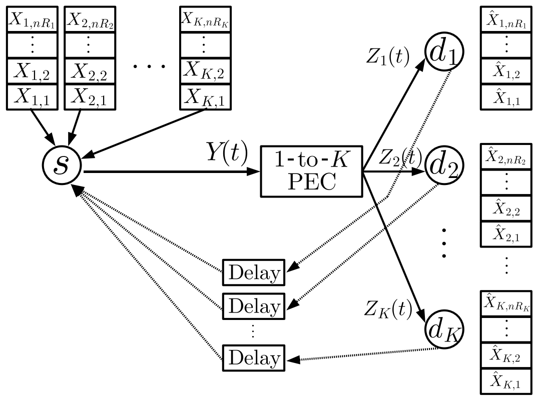
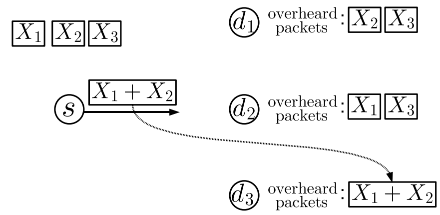
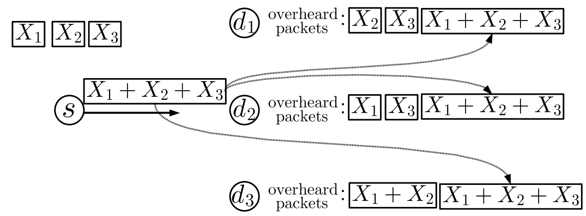


Fig. 1. Illustration of a 1-to-K broadcast PEC with COF.



(a) Sending the first Phase-2 packet $[X_1 + X_2]$.



(b) The optimal coding operation after sending the $[X_1 + X_2]$.

Fig. 2. Example of the suboptimality of the 2-phase approach.

We consider the setting with instant channel output feedback (COF). That is, for the t-th time slot, source s sends out a symbol

$$Y(t) = f_t\left(\{\mathbf{X}_k : \forall k \in [K]\}, \{\mathbf{Z}(\tau) : \tau \in [t-1]\}\right),$$

which is a function $f_t(\cdot)$ based on the information symbols $\{X_{k,j}\}$ and the COF $\{\mathbf{Z}(\tau) : \tau \in [t-1]\}$ of the previous transmissions. In the end of the n-th time slot, each d_k decodes its own desired symbols

$$\hat{\mathbf{X}}_k \triangleq \{\hat{X}_{k,j} : j \in [nR_k]\} = g_k(\{Z_k(t) : \forall t \in [n]\}),$$

where $g_k(\cdot)$ is the decoding function of d_k based on the corresponding observation $Z_k(t)$ for $t \in [n]$. A network code of length n and finite field $\mathsf{GF}(q)$ is thus defined by the corresponding n encoding functions $f_t(\cdot)$, $t \in [n]$, and K decoding functions $g_k(\cdot)$, $k \in [K]$. The functions $f_t(\cdot)$ and $g_k(\cdot)$ may or may not be linear. Note that we assume that the PEC channel parameters $\left\{p_{S\overline{[K]\backslash S}} : \forall S \in 2^{[K]}\right\}$ are available at s before transmission. See Fig. 1 for illustration.

We now define the achievable rate of a 1-to-K broadcast PEC with COF.

Definition 1: A rate vector $(R_1, \cdots, R_K)$ is achievable if for any $\epsilon > 0$, there exists a network code of length n and finite field $\mathsf{GF}(q)$ such that

$$\forall k \in [K],\ \mathsf{Prob}\left(\hat{\mathbf{X}}_k \neq \mathbf{X}_k\right) < \epsilon.$$

Definition 2: The capacity region of a 1-to-K broadcast PEC with COF is defined as the closure of all achievable rate vectors $(R_1, \cdots, R_K)$.

C. Existing Results

Theorem 1 (Theorem 3 in [9]): The capacity region (R_1, R_2) of a 1-to-2 broadcast PEC with COF is described by

$$\begin{cases} \frac{R_1}{p_1} + \frac{R_2}{p_{\cup\{1,2\}}} \leq 1 \\ \frac{R_1}{p_{\cup\{1,2\}}} + \frac{R_2}{p_2} \leq 1 \end{cases}. \tag{2}$$

One scheme that achieves the above capacity region is the 2-phase approach in [9]. That is, for any (R_1, R_2) in the interior of (2), perform the following coding operations.

In Phase 1, the source s sends out uncoded information packets X_{1,j_1} and X_{2,j_2} for all $j_1 \in [nR_1]$ and $j_2 \in [nR_2]$ until each packet is received by at least one receiver. Those X_{1,j_1} packets that are received by d_1 have already reached their intended receiver and thus will not be retransmitted in the second phase. Those X_{1,j_1} packets that are received by d_2 but not by d_1 need to be retransmitted in the second phase, and are thus stored in a separate queue $Q_{1;2\overline{1}}$. Symmetrically, the X_{2,j_2} packets that are received by d_1 but not by d_2 are stored in another queue $Q_{2;1\overline{2}}$. Those "overheard" packets in queues $Q_{1;2\overline{1}}$ and $Q_{2;1\overline{2}}$ will later be linearly mixed together in Phase 2. Each single coded packet in Phase 2 can now serve both d_1 and d_2 simultaneously. The *intersession network coding gain* in Phase 2 allows us to achieve the capacity region in (2).

Based on the same logic, [15] derives an achievability region for 1-to-K broadcast PECs with COF under a *perfectly symmetric setting*. The main idea can be viewed as an extension of the above 2-phase approach. That is, for Phase 1, the source s sends out all $X_{k,j}$, $\forall k \in [K], j \in [nR_k]$, until each of them is received by at least one of the receivers $\{d_k : k \in [K]\}$. Those $X_{k,j}$ packets that are received by d_k have already reached their intended destination and will not be transmitted in Phase 2. Those $X_{k,j}$ packets that are received by some other d_i but not by d_k are the "overheard packets," and could potentially be mixed with packets of the i-th session. In Phase 2, source s takes advantage of all the coding opportunities created in Phase 1 and mixes the packets of different sessions to capitalize the network coding gain. [20] implements such 2-phase approach while taking into account of various practical considerations, such as time-out and network synchronization.

D. The Suboptimality of The 2-Phase Approach

Although being throughput optimal for the simplest $K = 2$ case, the above 2-phase approach does not achieve capacity for the cases in which $K > 2$. To illustrate this point, consider the example in Fig. 2.

In Fig. 2(a), source s would like to serve three receivers d_1 to d_3. Each (s, d_k) session contains a single information packet X_k, and the goal is to convey each X_k to the intended receiver d_k for all $k = 1, 2, 3$. Suppose the 2-phase approach in Section II-C is used. During Phase 1, each packet is sent repeatedly until it is received by at least one receiver, which either conveys the packet to the intended receiver or creates an overheard packet that can be used in Phase 2. Suppose after Phase 1, d_1 has received X_2 and X_3, d_2 has received X_1 and X_3, and d_3 has not received any packet (Fig. 2(a)). Since each packet has reached at least one receiver, source s moves to Phase 2.

One can easily check that if s sends out a coded packet $[X_1 + X_2]$ in Phase 2, such packet can serve both d_1 and d_2. That is, d_1 (resp. d_2) can decode X_1 (resp. X_2) by subtracting X_2 (resp. X_1) from $[X_1 + X_2]$. Nonetheless, since the broadcast PEC is random, the coded packet $[X_1 + X_2]$ may or may not reach d_1 or d_2. Suppose that due to random channel realization, $[X_1 + X_2]$ reaches only d_3, see Fig. 2(a). The remaining question is what s should send for the next time slot. For the following, we compare the existing 2-phase approach and a new optimal decision.

The existing 2-phase approach: We first note that since d_3 received neither X_1 nor X_2 in the past, the newly received $[X_1 + X_2]$ cannot be used by d_3 to decode any information packet. In the existing results [9], [15], [20], d_3 thus discards the overheard $[X_1 + X_2]$, and s would continue sending $[X_1 + X_2]$ for the next time slot in order to capitalize this coding opportunity created in Phase 1.

The optimal decision: It turns out that the broadcast system can actually benefit from the fact that d_3 overhears the coded packet $[X_1 + X_2]$ even though neither X_1 nor X_2 can be decoded by d_3. More explicitly, instead of sending $[X_1 + X_2]$, s should send a new packet $[X_1 + X_2 + X_3]$ that mixes all three sessions together. With the new $[X_1 + X_2 + X_3]$ (see Fig. 2(b) for illustration), d_1 can decode the desired X_1 by subtracting both X_2 and X_3 from $[X_1 + X_2 + X_3]$. d_2 can decode the desired X_2 by subtracting both X_1 and X_3 from $[X_1 + X_2 + X_3]$. For d_3, even though d_3 does not know the values of X_1 and X_2, d_3 can still use the previously overheard $[X_1 + X_2]$ packet to subtract the interference $(X_1 + X_2)$ from $[X_1 + X_2 + X_3]$ and decode its desired packet X_3. As a result, the new coded packet $[X_1 + X_2 + X_3]$ serves all destinations d_1, d_2, and d_3, simultaneously. This new coding decision thus strictly outperforms the existing 2-phase approach.

Two critical observations can be made for this example. First of all, when d_3 overhears a coded $[X_1 + X_2]$ packet, even though d_3 can decode neither X_1 nor X_2, such new side information can still be used for future decoding. More explicitly, as long as s sends packets that are of the form $\alpha(X_1 + X_2) + \beta X_3$, the "aligned interference" $\alpha(X_1 + X_2)$ can be completely removed by d_3 without decoding individual X_1 and X_2. This technique is thus termed "*code alignment*," which is in parallel with the interference alignment method used in Gaussian interference channels [2]. Second of all, in the existing 2-phase approach, Phase 1 has the dual roles of sending uncoded packets to their intended receivers, and, at the same time, creating new coding opportunities (the overheard packets) for Phase 2. It turns out that this dual-purpose Phase-1 operation is indeed optimal (as will be seen in Sections IV and V). The suboptimality of the 2-phase approach for $K > 2$ is actually caused by the Phase-2 operation, in which source s only capitalizes the coding opportunities created in Phase 1 but does not create any new coding opportunities for subsequent packet mixing. One can thus envision that for the cases $K > 2$, an optimal policy should be a multi-phase policy, say an M-phase policy, such that for all $i \in [M-1]$ the packets sent in the i-th phase have dual roles of conveying desired information to their intended receivers and simultaneously creating new coding opportunities for the subsequent Phases $(i+1)$ to M. These two observations will be the building blocks of our achievability results.

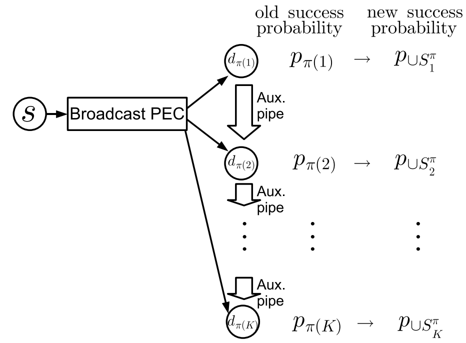


Fig. 3. Illustration of the proof of Proposition 1.

III. The Main Results

We have two groups of results. Section III-A focuses on general 1-to-K broadcast PECs with arbitrary values of the PEC parameters, while Section III-B focuses on 1-to-K broadcast PECs with some restrictive conditions on the values of the PEC parameters.

A. Capacity Results For General 1-to-K Broadcast PECs

We define any bijective function $\pi : [K] \mapsto [K]$ as a K-*permutation* and we sometimes use permutation as shorthand whenever it is clear in the context that we are focusing on $[K]$. There are totally $K!$ distinct K-permutations. Given any K-permutation π, for all $j \in [K]$ we define $S_j^\pi \triangleq \{\pi(l) : \forall l \in [j]\}$ as the set of the first j elements according to the given permutation π. We then have the following capacity outer bound for any 1-to-K broadcast PEC with COF.

Proposition 1: Any achievable rates $(R_1, \cdots, R_K)$ must satisfy the following $K!$ inequalities:

$$\forall \pi, \; \sum_{j=1}^{K} \frac{R_{\pi(j)}}{p_{\cup S_j^\pi}} \leq 1. \tag{3}$$

Proof: Proposition 1 can be proven by a simple extension of the outer bound arguments used in [9], [19]. (Note that when $K = 2$, Proposition 1 collapses to Theorem 3 of [9].)

For any given permutation π, consider a new broadcast channel with $(K-1)$ artificially created information pipes connecting all the receivers d_1 to d_K. More explicitly, for all

$j \in [K-1]$, create an auxiliary pipe from $d_{\pi(j)}$ to $d_{\pi(j+1)}$. See Fig. 3 for illustration. With the auxiliary pipes, any destination $d_{\pi(j)}$, $j \in [K]$, not only observes its corresponding output $Z_{\pi(j)}$ of the broadcast PEC but also has all the information $Z_{\pi(l)}$ of its "upstream receivers" $d_{\pi(l)}$ for all $l \in [j-1]$. Since we only create new pipes, any achievable rates of the original 1-to-K broadcast PEC with COF must also be achievable in the new 1-to-K broadcast PEC with COF in Fig. 3. The capacity of the new 1-to-K broadcast PEC with COF is thus an outer bound on the capacity of the original 1-to-K broadcast PEC with COF.

On the other hand, the new 1-to-K broadcast PEC in Fig. 3 is a physically degraded broadcast channel with the new success probability of d_k being $p_{\cup S_k^\pi}$ instead of $p_{\pi(k)}$ (see Fig. 3). [7] proves that COF does not increase the capacity of any physically degraded 1-to-2 broadcast channel. By extending the derivation steps in [7], one can easily prove that COF does not increase the capacity of any physically degraded 1-to-K broadcast channel for arbitrary K values. Therefore the capacity of the new 1-to-K broadcast PEC with COF is identical to the capacity of the new 1-to-K broadcast PEC without COF. Since (3) is the capacity of the new 1-to-K broadcast PEC without COF, (3) must be an outer bound of the capacity of the original 1-to-K PEC with COF. By considering different permutation π, the proof of Proposition 1 is complete. ■

In the following, we first provide the capacity results for general 1-to-3 broadcast PECs. We then state an achievability inner bound for general 1-to-K broadcast PECs with COF for arbitrary K values, which, together with the outer bound in Proposition 1 can effectively bracket the capacities for the cases in which $K \geq 4$.

Proposition 2: For any parameter values $\left\{p_{S\overline{\{1,2,3\}\backslash S}} : \forall S \in 2^{\{1,2,3\}}\right\}$ of a 1-to-3 broadcast PEC, the capacity outer bound in Proposition 1 is indeed the capacity region of a 1-to-3 broadcast PEC with COF.

The proof of Proposition 2 is provided in Section V-A.

To state the capacity inner bound, we need to define an additional function: $f_p(S\overline{T})$, which takes an input $S\overline{T}$ of two disjoint sets $S, T \in 2^{[K]}$. More explicitly, we define $f_p(S\overline{T})$ as the probability that a packet Y, transmitted through the 1-to-K PEC, is received by all those d_i with $i \in S$ and not received by any d_j with $j \in T$. For example, $f_p(S\overline{[K]\backslash S}) = p_{S\overline{[K]\backslash S}}$ for all $S \in 2^{[K]}$. For arbitrary disjoint S and T, we thus have

$$f_p(S\overline{T}) \triangleq \sum_{\forall S_1: S \subseteq S_1, T \subseteq ([K]\backslash S_1)} p_{S_1\overline{[K]\backslash S_1}}. \tag{4}$$

We also say that a *strict total ordering* "$\prec$" on $2^{[K]}$ is *cardinality-compatible* if

$$\forall S_1, S_2 \in 2^{[K]}, \quad |S_1| < |S_2| \Rightarrow S_1 \prec S_2. \tag{5}$$

Proposition 3: Fix any arbitrary cardinality-compatible, strict total ordering $\prec$ on $2^{[K]}$. For any general 1-to-K broadcast PEC with COF, a rate vector $(R_1, \cdots, R_K)$ can be achieved by a *linear network code* if there exist 2^K non-negative x variables, indexed by $S \in 2^{[K]}$:

$$\left\{x_S \geq 0 : \forall S \in 2^{[K]}\right\}, \tag{6}$$

and $K3^{K-1}$ non-negative w variables, indexed by the tuple $(k; S \to T)$:

$$\Big\{w_{k;S\to T} \geq 0 : \forall k \in [K], \forall S, T \in 2^{[K]}, \text{ satisfying } T \subseteq S \subseteq ([K]\backslash k)\Big\}, \tag{7}$$

such that jointly the following linear inequalities[3] are satisfied:

$$\sum_{\forall S: S \in 2^{[K]}} x_S < 1 \tag{8}$$

$$\forall T \in 2^{[K]}, \forall k \in T, \quad x_T \geq \sum_{\forall S: (T\backslash k) \subseteq S \subseteq ([K]\backslash k)} w_{k;S\to(T\backslash k)} \tag{9}$$

$$\forall k \in [K], \quad w_{k;\emptyset\to\emptyset} \cdot p_{\cup[K]} \geq R_k \tag{10}$$

$$\forall k \in [K], \forall S \subseteq ([K]\backslash k) \text{ satisfying } S \neq \emptyset,$$
$$\left(\sum_{\forall T_1: T_1 \subseteq S} w_{k;S\to T_1}\right) p_{\cup([K]\backslash S)} \geq \sum_{\substack{\forall S_1, T_1 \text{ : such that} \\ T_1 \subseteq S_1 \subseteq ([K]\backslash k), \\ T_1 \subseteq S, S \not\subseteq S_1}} w_{k;S_1\to T_1} \cdot f_p\left((S\backslash T_1)\overline{([K]\backslash S)}\right) \tag{11}$$

$$\forall k \in [K], S, T \in 2^{[K]} \text{ satisfying } T \subseteq S \subseteq ([K]\backslash k), T \neq S,$$
$$\left(w_{k;S\to T} + \sum_{\substack{\forall T_1 \text{ : such that } T_1 \subseteq S, \\ (T_1 \cup \{k\}) \prec (T \cup \{k\})}} w_{k;S\to T_1}\right) p_{\cup([K]\backslash S)} \leq \sum_{\substack{\forall S_1 : S_1 \prec S, \\ T \subseteq S_1 \subseteq ([K]\backslash k)}} w_{k;S_1\to T} \cdot f_p\left((S\backslash T)\overline{([K]\backslash S)}\right) + \sum_{\substack{\forall S_1, T_1 \text{ : such that} \\ T_1 \subseteq S_1 \subseteq ([K]\backslash k), \\ (T_1 \cup \{k\}) \prec (T \cup \{k\}), \\ T_1 \subseteq S, S \not\subseteq S_1}} w_{k;S_1\to T_1} \cdot f_p\left((S\backslash T_1)\overline{([K]\backslash S)}\right). \tag{12}$$

The proof of Proposition 3 is provided in Section V-C.

Remark: For $K \geq 4$ and some general classes of PEC parameters, one can prove that the inner bound of Proposition 3 meets the outer bound in Proposition 1. Two such classes are discussed in the next subsection.

B. Capacity Results For Two Classes of 1-to-K Broadcast PECs

We first provide the capacity results for *symmetric* broadcast PECs.

[3] There are totally $(1 + K2^{K-1} + K3^{K-1})$ inequalities. More explicitly, (8) describes one inequality. There are $K2^{K-1}$ inequalities having the form of (9). There are totally $K3^{K-1}$ inequalities having the form of one of (10), (11), and (12). For comparison, the outer bound in Proposition 1 actually has more inequalities asymptotically ($K!$ of them) than those in Proposition 3.

Definition 3: A 1-to-K broadcast PEC is *symmetric* if the channel parameters $\left\{p_{S\overline{[K]\backslash S}} : \forall S \in 2^{[K]}\right\}$ satisfy

$$\forall S_1, S_2 \in 2^{[K]} \text{ with } |S_1| = |S_2|,\ p_{S_1\overline{[K]\backslash S_1}} = p_{S_2\overline{[K]\backslash S_2}}.$$

That is, the success probability $p_{S\overline{[K]\backslash S}}$ depends only on $|S|$, the size of S, and does not depend on the subset of receivers being considered.

Proposition 4: For any symmetric 1-to-K broadcast PEC with COF, the capacity outer bound in Proposition 1 is indeed the capacity region.

The proof of Proposition 4 is provided in Section V-D.

The perfect channel symmetry condition in Proposition 4 may be a bit restrictive for real environments as most broadcast channels are non-symmetric. A more realistic setting is to allow channel asymmetry while assuming spatial independence between different destinations d_k.

Definition 4: A 1-to-K broadcast PEC is *spatially independent* if the channel parameters $\left\{p_{S\overline{[K]\backslash S}} : \forall S \in 2^{[K]}\right\}$ satisfy

$$\forall S \in 2^{[K]},\ p_{S\overline{[K]\backslash S}} = \left(\prod_{i\in S} p_i\right)\left(\prod_{j\in[K]\backslash S}(1-p_j)\right),$$

where p_k is the marginal success probability of destination d_k.

Note: A symmetric 1-to-K broadcast PEC needs not be spatially independent. A spatially independent PEC is symmetric if $p_1 = p_2 = \cdots = p_K$.

To describe the capacity results for spatially independent 1-to-K PECs, we need the following additional definition.

Definition 5: Consider a 1-to-K broadcast PEC with marginal success probabilities p_1 to p_K. Without loss of generality, assume $p_1 \leq p_2 \leq \cdots \leq p_K$, which can be achieved by relabeling. We say a rate vector $(R_1, \cdots, R_K)$ is *one-sidedly fair* if

$$\forall i < j, \quad R_i \cdot (1-p_i) \geq R_j \cdot (1-p_j).$$

We use Λ_{osf} to denote the collection of all one-sidedly fair rate vectors.

The one-sided fairness contains many practical scenarios of interest. For example, the perfectly fair rate vector $(R, R, \cdots, R)$ by definition is also one-sidedly fair. Another example is when $\min(p_1, \cdots, p_K) \geq \frac{1}{2}$ and we allow the rate R_k to be proportional to the corresponding marginal success probability p_k, i.e., $R_k = p_k R$. Then the rate vector $(p_1 R, p_2 R, \cdots, p_K R)$ is also one-sidedly fair.

We then have the following proposition.

Proposition 5: Suppose the 1-to-K PEC of interest is spatially independent with marginal success probabilities $0 < p_1 \leq p_2 \leq \cdots \leq p_K$. Any one-sidedly fair rate vector $(R_1, \cdots, R_K) \in \Lambda_{\text{osf}}$ is in the capacity region if and only if $(R_1, \cdots, R_K) \in \Lambda_{\text{osf}}$ satisfies

$$\sum_{k=1}^{K} \frac{R_k}{1 - \prod_{l=1}^{k}(1-p_l)} \leq 1. \tag{13}$$

Proposition 5 implies that the region in Proposition 1 is indeed the capacity when focusing on the one-sidedly fair rate region Λ_{osf}. The proof of Proposition 5 is provided in Section V-D.

IV. The Packet Evolution Schemes

In the following, we describe a new class of coding schemes, termed the *packet evolution* (PE) scheme, which embodies the concept of code alignment and is the building block of the capacity / achievability results in Section III.

A. Description Of The Packet Evolution Scheme

Recall that each (s, d_k) session has nR_k information packets $X_{k,1}$ to X_{k,nR_k}. We associate each of the $\sum_{k=1}^{K} nR_k$ information packets with *an intersession coding vector* $\mathbf{v}$ and a set $S \subseteq [K]$. An intersession coding vector is a $\left(\sum_{k=1}^{K} nR_k\right)$-dimensional row vector with each coordinate being a scalar in $\mathsf{GF}(q)$. Before the start of the broadcast, for any $k \in [K]$ and $j \in [nR_k]$ we initialize the corresponding vector $\mathbf{v}$ of $X_{k,j}$ in a way that the only nonzero coordinate of $\mathbf{v}$ is the coordinate corresponding to $X_{k,j}$ and all other coordinates are zero. Without loss of generality, we set the value of the only non-zero coordinate to one. That is, initially the coding vectors $\mathbf{v}$ are set to the elementary basis vectors of the entire $\left(\sum_{k=1}^{K} nR_k\right)$-dimensional message space.

For any $k \in [K]$ and $j \in [nR_k]$ the set S of $X_{k,j}$ is initialized to $\emptyset$. As will be clear shortly after, we call S the *overhearing set*[4] of the packet $X_{k,j}$. For easier reference, we use $\mathbf{v}(X_{k,j})$ and $S(X_{k,j})$ to denote the intersession coding vector and the overhearing set of the given packet $X_{k,j}$.

Throughout the n broadcast time slots, source s constantly updates $S(X_{k,j})$ and $\mathbf{v}(X_{k,j})$ according to the COF. The main structure of a packet evolution scheme can now be described as follows.

---

§ The Packet Evolution Scheme

1: Source s maintains a single flag $\mathsf{f}_{\text{change}}$. Initially, set $\mathsf{f}_{\text{change}} \leftarrow 1$.
2: **for** $t = 1, \cdots, n$, **do**
3: In the beginning of the t-th time slot, do Lines 4 to 10.
4: **if** $\mathsf{f}_{\text{change}} = 1$ **then**
5: Choose two non-empty subsets $T, T_{\text{sel}} \in 2^{[K]}$ satisfying $T_{\text{sel}} \subseteq T$.
6: Run a subroutine Packet Selection, which takes T and T_{sel} as input and outputs a collection of $|T_{\text{sel}}|$ packets $\{X_{k,j_k} : \forall k \in T_{\text{sel}}\}$, one from each session $k \in T_{\text{sel}}$, respectively. The selected packets are termed the *target packets*, and we require that all target packets X_{k,j_k} satisfy $(S(X_{k,j_k}) \cup \{k\}) \supseteq T$ for all $k \in T_{\text{sel}}$.
7: Generate uniformly randomly $|T_{\text{sel}}|$ coefficients $c_k \in \mathsf{GF}(q)$ for all $k \in T_{\text{sel}}$ and construct an intersession coding vector $\mathbf{v}_{\text{tx}} \leftarrow \sum_{k\in T_{\text{sel}}} c_k \cdot \mathbf{v}(X_{k,j_k})$.
8: Set $\mathsf{f}_{\text{change}} \leftarrow 0$.
9: **end if**
10: Send out a linearly intersession coded packet according to the coding vector $\mathbf{v}_{\text{tx}}$. That is, we send

$$Y_{\text{tx}} = \mathbf{v}_{\text{tx}} \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$$

[4]Unlike the existing results [12], in this work the overhearing set does not mean that the receivers d_i in $S(X_{k,j})$ have known the value of $X_{k,j}$. Detailed discussion of the overhearing set $S(X_{k,j})$ are provided in Lemma 2.

where $(X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$ is a column vector consisting of all information symbols.[5]

```
11:     In the end of the t-th time slot, use a subroutine UPDATE to revise the v(X_{k,j_k}) and S(X_{k,j_k}) values of all target packets X_{k,j_k} based on the COF.
12:     if the S(X_{k,j_k}) value changes for at least one target packet X_{k,j_k} after the UPDATE then
13:         Set f_change ← 1.
14:     end if
15: end for
```

In summary, a group of target packets $\{X_{k,j_k}\}$ are selected according to the choice of the subset T. All k must be in T and the subset T_{sel} is used to denote the subset of T from which target packets are actually selected. The corresponding vectors $\{\mathbf{v}(X_{k,j_k})\}$ are used to construct a coding vector $\mathbf{v}_{\text{tx}}$. The same coded packet Y_{tx}, corresponding to $\mathbf{v}_{\text{tx}}$, is then sent repeatedly until one of the target packets X_{k,j_k} *evolves* (when the corresponding $S(X_{k,j_k})$ changes). Then a new subset T is chosen and the process is repeated until we use up all n time slots. Three subroutines are used as the building blocks of a packet evolution method: (i) How to choose the non-empty sets $T, T_{\text{sel}} \in 2^{[K]}$; (ii) For each $k \in T_{\text{sel}}$, how to select a single target packet X_{k,j_k} among all $X_{k,j}$ satisfying $(S(X_{k,j}) \cup \{k\}) \supseteq T$; and (iii) How to update the coding vectors $\mathbf{v}(X_{k,j_k})$ and the overhearing sets $S(X_{k,j_k})$. In the following, we first describe the detailed update rules.

§ UPDATE OF $S(X_{k,j_k})$ AND $\mathbf{v}(X_{k,j_k})$

```
Input: The T, T_sel, and v_tx used for transmission in the current time slot; And S_rx, the set of destinations d_i which receive the transmitted coded packet in the current time slot. (S_rx is obtained through the COF in the end of the current time slot.)
for all k ∈ T_sel do
    if S_rx ⊈ S(X_{k,j_k}) then
        Set S(X_{k,j_k}) ← (T ∩ S(X_{k,j_k})) ∪ S_rx.
        Set v(X_{k,j_k}) ← v_tx.
    end if
end for
```

An Illustrative Example Of The PE Scheme:

Let us revisit the optimal coding scheme of the example in Fig. 2 of Section II-D. Before broadcast, the three information packets X_1 to X_3 have the corresponding $\mathbf{v}$ and S: $\mathbf{v}(X_1) = (1, 0, 0)$, $\mathbf{v}(X_2) = (0, 1, 0)$, and $\mathbf{v}(X_3) = (0, 0, 1)$, and $S(X_1) = S(X_2) = S(X_3) = \emptyset$. We use the following table for summary.

X_1: (1,0,0),$\emptyset$	X_2: (0,1,0),$\emptyset$	X_3: (0,0,1),$\emptyset$

Consider a duration of 5 time slots.

Slot 1: Suppose that s chooses $T = T_{\text{sel}} = \{1\}$. Since $(\emptyset \cup \{1\}) \supseteq T$, PACKET SELECTION outputs X_1. The coding vector $\mathbf{v}_{\text{tx}}$ is thus a scaled version of $\mathbf{v}(X_1) = (1, 0, 0)$. Without loss of generality, we choose $\mathbf{v}_{\text{tx}} = (1, 0, 0)$. Based on $\mathbf{v}_{\text{tx}}$, s transmits a packet $1X_1 + 0X_2 + 0X_3 = X_1$. Suppose $[X_1]$ is received by d_2, i.e., $S_{\text{rx}} = \{2\}$. Since $S_{\text{rx}} = \{2\}$ is not a subset of $S(X_1) = \emptyset$, UPDATE thus sets $S(X_1) = \{2\}$ and $\mathbf{v}(X_1) = \mathbf{v}_{\text{tx}} = (1, 0, 0)$. The packet summary becomes

X_1: (1,0,0),$\{2\}$	X_2: (0,1,0),$\emptyset$	X_3: (0,0,1),$\emptyset$

Slot 2: Suppose that s chooses $T = T_{\text{sel}} = \{2\}$. Since $(\emptyset \cup \{2\}) \supseteq T$, PACKET SELECTION outputs X_2. The coding vector $\mathbf{v}_{\text{tx}}$ is thus a scaled version of $\mathbf{v}(X_2) = (0, 1, 0)$. Without loss of generality, we choose $\mathbf{v}_{\text{tx}} = (0, 1, 0)$ and accordingly $[X_2]$ is sent. Suppose $[X_2]$ is received by d_1, i.e., $S_{\text{rx}} = \{1\}$. Since $S_{\text{rx}} \not\subseteq S(X_2)$, after UPDATE the packet summary becomes

X_1: (1,0,0),$\{2\}$	X_2: (0,1,0),$\{1\}$	X_3: (0,0,1),$\emptyset$

Slot 3: Suppose that s chooses $T = T_{\text{sel}} = \{3\}$ and PACKET SELECTION outputs X_3. The coding vector $\mathbf{v}_{\text{tx}}$ is thus a scaled version of $\mathbf{v}(X_3) = (0, 0, 1)$, and we choose $\mathbf{v}_{\text{tx}} = (0, 0, 1)$. Accordingly $[X_3]$ is sent. Suppose $[X_3]$ is received by d_1 and d_2, i.e., $S_{\text{rx}} = \{1, 2\}$. Then after UPDATE, the packet summary becomes

X_1: (1,0,0),$\{2\}$	X_2: (0,1,0),$\{1\}$	X_3: (0,0,1),$\{1, 2\}$

Slot 4: Suppose that s chooses $T = T_{\text{sel}} = \{1, 2\}$. Since $(S(X_1) \cup \{1\}) \supseteq T$ and $(S(X_2) \cup \{2\}) \supseteq T$, PACKET SELECTION outputs $\{X_1, X_2\}$. $\mathbf{v}_{\text{tx}}$ is thus a linear combination of $\mathbf{v}(X_1) = (1, 0, 0)$ and $\mathbf{v}(X_2) = (0, 1, 0)$. Without loss of generality, we choose $\mathbf{v}_{\text{tx}} = (1, 1, 0)$ and accordingly $[X_1 + X_2]$ is sent. Suppose $[X_1 + X_2]$ is received by d_3, i.e., $S_{\text{rx}} = \{3\}$. Then during UPDATE, for X_1, $S_{\text{rx}} = \{3\} \not\subseteq S(X_1) = \{2\}$. UPDATE thus sets $S(X_1) = \{2, 3\}$ and $\mathbf{v}(X_1) = \mathbf{v}_{\text{tx}} = (1, 1, 0)$. For X_2, $S_{\text{rx}} = \{3\} \not\subseteq S(X_2) = \{1\}$. UPDATE thus sets $S(X_2) = \{1, 3\}$ and $\mathbf{v}(X_2) = \mathbf{v}_{\text{tx}} = (1, 1, 0)$. The packet summary becomes

X_1: (1,1,0),$\{2, 3\}$	X_2: (1,1,0),$\{1, 3\}$
X_3: (0,0,1),$\{1, 2\}$	

Slot 5: Suppose that s chooses $T = T_{\text{sel}} = \{1, 2, 3\}$. By Line 6 of THE PACKET EVOLUTION SCHEME, the subroutine PACKET SELECTION outputs $\{X_1, X_2, X_3\}$. $\mathbf{v}_{\text{tx}}$ is thus a linear combination of $\mathbf{v}(X_1) = (1, 1, 0)$, $\mathbf{v}(X_2) = (1, 1, 0)$, and $\mathbf{v}(X_3) = (0, 0, 1)$, which is of the form $\alpha(X_1 + X_2) + \beta X_3$. *Note that the packet evolution scheme automatically achieves code alignment*, which is the key component of the optimal coding policy in Section II-D. Without loss of generality, we choose $\alpha = \beta = 1$ and $\mathbf{v}_{\text{tx}} = (1, 1, 1)$. $Y_{\text{tx}} = [X_1 + X_2 + X_3]$ is sent accordingly. Suppose $[X_1 + X_2 + X_3]$ is received by $\{d_1, d_2, d_3\}$, i.e., $S_{\text{rx}} = \{1, 2, 3\}$. Then after UPDATE, the summary of the packets becomes

X_1: (1,1,1),$\{1, 2, 3\}$	X_2: (1,1,1),$\{1, 2, 3\}$
X_3: (1,1,1),$\{1, 2, 3\}$	

From the above step-by-step illustration, we see that the optimal coding policy in Section II-D is a special instance of

[5] It is critical to note that the coding operation is based purely on $\mathbf{v}_{\text{tx}}$ rather than on the list of the target packets X_{k,j_k}. It is possible that $\mathbf{v}_{\text{tx}}$ has non-zero coordinates corresponding to some $X_{k',j}$ that are not one of the target packets X_{k,j_k}. By our construction, those $X_{k',j}$ will still participate in creating the coded packet even though they are not one of the target packets.

a packet evolution scheme.

B. Properties of A Packet Evolution Scheme

We term the packet evolution (PE) scheme in Section IV-A a *generic* PE method since it does not specify how to choose T, T_{sel}, and the target packets X_{k,j_k}, and only requires $T_{\text{sel}} \subseteq T$ and the output of PACKET SELECTION satisfying $(S(X_{k,j_k}) \cup \{k\}) \supseteq T, \forall k \in T_{\text{sel}}$. In this subsection, we state some key properties for any generic PE scheme. The intuition of the PE scheme is based on these key properties and will be discussed further in Section IV-C.

We first define the following notation for any linear network code. (Note that the PE scheme is a linear network code.)

Definition 6: Consider any linear network code. For any destination d_k, each of the received packet $Z_k(t)$ can be represented by a vector $\mathbf{w}_k(t)$, which is a $\left(\sum_{k=1}^{K} nR_k\right)$-dimensional vector containing the coefficients used to generate $Z_k(t)$. That is, $Z_k(t) = \mathbf{w}_k(t) \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$. If $Z_k(t)$ is an erasure, we simply set $\mathbf{w}_k(t)$ to be an all-zero vector. The *knowledge space* of destination d_k in the end of time t is denoted by $\Omega_{\mathrm{Z},k}(t)$, which is the linear span of $\mathbf{w}_k(\tau)$, $\tau \leq t$. That is, $\Omega_{\mathrm{Z},k}(t) \triangleq \mathsf{span}(\mathbf{w}_k(\tau) : \forall \tau \in [t])$.

Definition 7: For any non-coded information packet $X_{k,j}$, we can view it as a product $\delta_{k,j} \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$, where $\delta_{k,j}$ is a $\left(\sum_{k=1}^{K} nR_k\right)$-dimensional row vector with a single one in the corresponding coordinate and all other coordinates being zero. The *message space* of d_k is then defined as $\Omega_{M,k} = \mathsf{span}(\delta_{k,j} : \forall j \in [nR_k])$.

With the above definitions, we have the following self-explanatory lemma:

Lemma 1: In the end of time t, destination d_k is able to decode all the desired information packets $X_{k,j}$, $\forall j \in [nR_k]$, if and only if $\Omega_{M,k} \subseteq \Omega_{Z,k}(t)$.

We now define "non-interfering vectors" from the perspective of a destination d_k.

Definition 8: In the end of time t, a vector $\mathbf{v}$ is "non-interfering" from the perspective of d_k if

$$\mathbf{v} \in \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}).$$

We note that any non-interfering vector $\mathbf{v}$ can always be expressed as the sum of two vectors $\mathbf{v}'$ and $\mathbf{w}$, where $\mathbf{v}' \in \Omega_{M,k}$ is a linear combination of all information vectors for d_k and $\mathbf{w} \in \Omega_{Z,k}(t)$ is a linear combination of all the packets received by d_k. If $\mathbf{v}' = 0$, then the packet generated from $\mathbf{v} = \mathbf{w}$ is *transparent* from d_k's perspective since d_k can compute the value of $\mathbf{w} \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$ from its current knowledge space $\Omega_{Z,k}(t)$. If $\mathbf{v}' \neq 0$, then from d_k's perspective, the packet generated from $\mathbf{v} = \mathbf{v}' + \mathbf{w}$ can be viewed as a pure information packet corresponding to $\mathbf{v}' \in \Omega_{M,k}$ by subtracting the content resulted from the unwanted $\mathbf{w}$ vector. In either case, the packet corresponding to $\mathbf{v}$ is *not interfering* with the transmission of the (s, d_k) session, which gives the name of "non-interfering vectors."

The following Lemmas 2 and 3 discuss the time dynamics of the PE scheme. To distinguish different time instants, we add a time subscript and use $S_{t-1}(X_{k,j_k})$ and $S_t(X_{k,j_k})$ to denote the overhearing set of X_{k,j_k} in the end of time $(t-1)$ and t, respectively. Similarly, $\mathbf{v}_{t-1}(X_{k,j_k})$ and $\mathbf{v}_t(X_{k,j_k})$ denote the coding vectors in the end of time $(t-1)$ and t, respectively.

Lemma 2: In the end of the t-th time slot, consider any $X_{k,j}$ out of all the information packets $X_{1,1}$ to X_{K,nR_K}. Its assigned vector $\mathbf{v}_t(X_{k,j})$ is non-interfering from the perspective of d_i for all $i \in (S_t(X_{k,j}) \cup \{k\})$.

To illustrate Lemma 2, consider our 5-time-slot example. In the end of Slot 4, we have $\mathbf{v}(X_1) = (1,1,0)$ and $S(X_1) \cup \{1\} = \{1,2,3\}$. From d_1's perspective, $\Omega_{Z,1}(4) = \mathsf{span}((0,1,0),(0,0,1))$ and $\Omega_{M,1} = \mathsf{span}((1,0,0))$. $\mathbf{v}(X_1) \in \mathsf{span}(\Omega_{Z,1}(4), \Omega_{M,1})$ is indeed non-interfering from d_1's perspective. The same reasoning can be applied to d_2 to show that $\mathbf{v}(X_1)$ is non-interfering from d_2's perspective. For d_3, $\Omega_{Z,3}(4) = \mathsf{span}((1,1,0))$ and $\Omega_{M,3} = \mathsf{span}((0,0,1))$. $\mathbf{v}(X_1) \in \mathsf{span}(\Omega_{Z,3}(4), \Omega_{M,3})$ is indeed non-interfering from d_3's perspective. Lemma 2 holds for our illustrative example.

We now define the *remaining space* of a PE scheme:

Definition 9: In the end of the t-th time slot, the remaining space of destination d_k is defined by $\Omega_{R,k}(t)$:

$$\Omega_{R,k}(t) \triangleq \mathsf{span}(\mathbf{v}_t(X_{k,j}) : \forall j \in [nR_k] \text{ satisfying } k \notin S_t(X_{k,j})).$$

Lemma 3: For any n and any $\epsilon > 0$, there exists a sufficiently large finite field $\mathsf{GF}(q)$ such that for all $k \in [K]$ and $t \in [n]$,

$$\mathsf{Prob}\left(\mathsf{span}(\Omega_{Z,k}(t), \Omega_{R,k}(t)) = \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})\right) > 1 - \epsilon.$$

Intuitively, Lemma 3 says that if in the end of time t we transmit some additional packets $\mathbf{v}_t(X_{k,j}) \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$ from s to d_k through a noise-free information pipe for all the *remaining* coding vectors $\{\mathbf{v}_t(X_{k,j}) : \forall j \in [nR_k], k \notin S_t(X_{k,j})\}$, then with high probability, d_k can successfully decode all the desired information packets $X_{k,1}$ to X_{k,nR_k} (see Lemma 1) by the knowledge space $\Omega_{Z,k}(t)$ and the new information of the remaining space $\Omega_{R,k}(t)$.

Lemma 3 directly implies the following corollary.

Corollary 1: For any n and any $\epsilon > 0$, there exists a sufficiently large finite field $\mathsf{GF}(q)$ such that the following statement holds. If in the end of the n-th time slot, all information packets $X_{k,j}$ have $S_n(X_{k,j}) \ni k$, then

$$\mathsf{Prob}(\forall k, d_k \text{ can decode all its desired } \{X_{k,j}\}) > 1 - \epsilon.$$

Proof: If in the end of the n-th time slot, all $X_{k,j}$ have $S_n(X_{k,j}) \ni k$, then the corresponding $\Omega_{R,k}(n) = \{0\}$ contains only the origin for all $k \in [K]$. Therefore, Corollary 1 is simply a restatement of Lemmas 1 and 3. ■

To illustrate Corollary 1, consider our 5-time-slot example. In the end of Slot 5, since $k \in S(X_k)$ for all $k \in \{1,2,3\}$, Corollary 1 guarantees that with high probability all d_k can decode the desired X_k, which was first observed in the example of Section II-D.

The proofs of Lemmas 2 and 3 are relegated to Appendices A and B, respectively.

C. The Intuitions Of The Packet Evolution Scheme

Lemmas 2 and 3 are the key properties of a PE scheme. In this subsection, we discuss the corresponding intuitions.

Receiving **the information packet** $X_{k,j}$**:** Each information packet keeps a coding vector $\mathbf{v}(X_{k,j})$. Whenever we would like to communicate $X_{k,j}$ to destination d_k, instead of sending a non-coded packet $X_{k,j}$ directly, we send an intersession coded packet according to the coding vector $\mathbf{v}(X_{k,j})$. Lemma 3 shows that if we send the packets corresponding to all coding vectors $\mathbf{v}(X_{k,j})$ that have not been heard by d_k (with $k \notin S(X_{k,j})$) through a noise-free information pipe, then d_k can indeed decode all the desired packets $X_{k,j}$ with close-to-one probability. It also implies, although in an implicit way, that once a $\mathbf{v}(X_{k,j_0})$ is heard by d_k for some j_0 (therefore $k \in S(X_{k,j_0})$), there is no need to transmit this particular $\mathbf{v}(X_{k,j_0})$ in the later time slots. Jointly, these two implications show that we can indeed use the coded packet $\mathbf{v}(X_{k,j}) \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$ as a substitute for $X_{k,j}$ without losing any information. In the broadest sense, we can say that d_k receives a packet $X_{k,j}$ if the corresponding $\mathbf{v}(X_{k,j})$ successfully arrives d_k in some time slot t.

For each $X_{k,j}$, the set $S(X_{k,j})$ serves two purposes: (i) Keep track of whether its intended destination d_k has received this $X_{k,j}$ (through the corresponding $\mathbf{v}(X_{k,j})$), and (ii) Keep track of whether $\mathbf{v}(X_{k,j})$ is non-interfering to other destinations d_i, $i \neq k$. We discuss these two purposes separately.

Tracking the reception of the intended d_k**:** We first note that in the end of time 0, d_k has not received any packet and we indeed have $k \notin S(X_{k,j}) = \emptyset$. We then notice that for any given $X_{k,j}$, the set $S(X_{k,j})$ evolves over time. By Line 4 of the UPDATE, we can prove that as time proceeds, the first time t_0 such that $k \in S(X_{k,j})$ must be the first time when $X_{k,j}$ is received by d_k (i.e., $X_{k,j}$ is chosen in the beginning of time t and $k \in S_{\text{rx}}$ in the end of time t). One can also show that for any $X_{k,j}$ once $k \in S_{t_0}(X_{k,j})$ in the end of time t_0 for some t_0, we will have $k \in S_t(X_{k,j})$ for all $t \geq t_0$. By the above reasonings, checking whether $k \in S(X_{k,j})$ indeed tells us whether the intended receiver d_k has received $X_{k,j}$.

Tracking the non-interference from the perspective of $d_i \neq d_k$**:** Lemma 2 also ensures that $\mathbf{v}(X_{k,j})$ is non-interfering from d_i's perspective for any $i \in S(X_{k,j})$, $i \neq k$. Therefore $S(X_{k,j})$ successfully tracks whether $\mathbf{v}(X_{k,j})$ is non-interfering from the perspectives of d_i, $i \neq k$.

Serving multiple destinations simultaneously by mixing non-interfering packets: The above discussion ensures that when we would like to send an information packet X_{k,j_k} to d_k, we can send a coded packet $\mathbf{v}(X_{k,j_k}) \cdot (X_{1,1}, \cdots, X_{K,nR_K})^{\mathrm{T}}$ as an information-lossless substitute. On the other hand, by Lemma 2, such $\mathbf{v}(X_{k,j_k})$ is non-interfering from d_i's perspective for all $i \in (S(X_{k,j_k}) \cup \{k\})$. Therefore, instead of sending a single coded packet corresponding to $\mathbf{v}(X_{k,j_k})$, it is beneficial to *combine* the transmission of two coded packets, corresponding to $\mathbf{v}(X_{k,j_k})$ and $\mathbf{v}(X_{l,j_l})$, respectively, as long as $l \in S(X_{k,j_k})$ and $k \in S(X_{l,j_l})$. By generalizing this idea, a PE scheme first selects a $T_{\text{sel}} \subseteq [K]$ and then choose all X_{k,j_k} such that $k \in T_{\text{sel}}$ and $\mathbf{v}(X_{k,j_k})$ are non-interfering from d_l's perspective for all $l \in T_{\text{sel}} \backslash k$ (see Line 6 of the PE scheme). This thus ensures that the coded packet $\mathbf{v}_{\text{tx}}$ in Line 7 of the PE scheme can serve all destinations $k \in T_{\text{sel}}$ simultaneously.

Creating new coding opportunities while exploiting the existing coding opportunities: As discussed in the example of Section II-D, the suboptimality of the existing 2-phase approach for $K \geq 3$ destinations is due to the fact that it fails to create new coding opportunities while exploiting old coding opportunities. The PE scheme was designed to solve this problem. More explicitly, for each $X_{k,j}$ the $\mathbf{v}(X_{k,j})$ is non-interfering for all d_i satisfying $i \in (S(X_{k,j}) \cup \{k\})$. Therefore, the larger the set $S(X_{k,j})$ is, the larger the number of sessions that can be coded together when transmitting a coded packet corresponding to $\mathbf{v}(X_{k,j})$. To create more coding opportunities, we thus need to be able to enlarge the $S(X_{k,j})$ set over time. Let us temporarily assume that the PACKET SELECTION in Line 6 only chooses the $X_{k,j}$ satisfying $S(X_{k,j}) = T \backslash k$. Then Line 4 of the UPDATE guarantees that if some other d_i, $i \notin T$, overhears the coded transmission, we can update $S(X_{k,j})$ with a strictly larger set $(T \cap S(X_{k,j})) \cup S_{\text{rx}} = S(X_{k,j}) \cup S_{\text{rx}}$. Therefore, new coding opportunity is created since we can now mix more sessions together with $X_{k,j}$. Note that the coding vector $\mathbf{v}(X_{k,j})$ is also updated accordingly. The new $\mathbf{v}(X_{k,j})$ represents the necessary "code alignment" in order to utilize this newly created coding opportunity. The (near-) optimality of the PE scheme is rooted deeply in the concept of code alignment, which continuously aligns the "non-interfering subspaces" through the joint use of $S(X_{k,j})$ and $\mathbf{v}(X_{k,j})$.

V. Quantify The Achievable Rates of PE Schemes

In this section, we describe how to use the PE schemes to attain the capacity of 1-to-3 broadcast PECs with COF (Proposition 2), the achievability results for general 1-to-K broadcast PEC with COF (Proposition 3), the capacity results for symmetric broadcast PECs (Proposition 4) and for spatially independent PECs with one-sided fairness constraints (Proposition 5).

A. Achieving the Capacity of 1-to-3 Broadcast PECs With COF — Detailed Construction and Analysis

Consider a 1-to-3 broadcast PEC with arbitrary channel parameters $\{p_{S\overline{\{1,2,3\}\backslash S}}\}$. Without loss of generality, assume that the marginal success probability $p_k > 0$ for $k = 1, 2, 3$. Given a rate vector (R_1, R_2, R_3) and the PEC channel parameters $\{p_{S\overline{\{1,2,3\}\backslash S}}\}$, we say that destination d_i *dominates* another destination d_k if

$$R_i \left(\frac{1}{p_{\cup(\{1,2,3\}\backslash k)}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \geq R_k \left(\frac{1}{p_{\cup(\{1,2,3\}\backslash i)}} - \frac{1}{p_{\cup\{1,2,3\}}} \right). \quad (14)$$

Lemma 4: For distinct values of $i, k, l \in \{1, 2, 3\}$, if d_i dominates d_k, and d_k dominates d_l, then we must have d_i dominates d_l.

The proof of Lemma 4 is provided in Appendix C.

By Lemma 4, we can assume that d_1 dominates d_2, d_2 dominates d_3, and d_1 dominates d_3, which can be achieved by relabeling the destinations. In the following, we provide a detailed capacity-achieving PE scheme by explicitly describing how to choose the subsets T, T_{sel}, and the corresponding target packets X_{k,j_k}, $k \in T_{\text{sel}}$, in a generic PE scheme. The proposed PE scheme contains four major phases.

---

§ A CAPACITY-ACHIEVING PE SCHEME FOR 1-TO-3 PECS

After sorting and relabeling the destinations according to their dominance relationship as discussed previously, go through the following phases in sequence.

Phase 1: There are three sub-phases denoted by **Phase 1.*k***, $k = 1$ to 3. Move along the sub-phases sequentially for $k = 1$ to 3. When we are in **Phase 1.*k***, do the following: If there exists at least one $X_{k,j}$ satisfying $S(X_{k,j}) = \emptyset$, then choose $T = T_{\text{sel}} = \{k\}$ and select the target packet X_{k,j_k} arbitrarily from such $X_{k,j}$. If not, move to the next sub-phase.

Phase 2: There are three sub-phases denoted by **Phase 2.*k***, $k = 1$ to 3. Move along the sub-phases sequentially for $k = 1$ to 3. When we are in **Phase 2.*k***, do the following: Denote the elements of $\{1, 2, 3\}\backslash k$ by i_1 and i_2. If there exists at least one $X_{i_1,j}$ satisfying $S(X_{i_1,j}) = \{i_2\}$ and at least one $X_{i_2,l}$ satisfying $S(X_{i_2,l}) = \{i_1\}$, then choose $T = T_{\text{sel}} = \{i_1, i_2\}$ and select the target packets $X_{i_1,j_{i_1}}$ and $X_{i_2,j_{i_2}}$ arbitrarily from such $X_{i_1,j}$ and $X_{i_2,l}$ packets. If not, move to the next sub-phase.

Phase 3: There are three sub-phases denoted by **Phase 3.*k***, $k = 1$ to 3. Move along the sub-phases sequentially for $k = 1$ to 3. When we are in **Phase 3.*k***, denote the elements of $\{1, 2, 3\}\backslash k$ by i_1 and i_2 and without loss of generality assume $i_1 < i_2$; Do the following steps:

```
if there exists at least one X_{i_1,j} satisfying S(X_{i_1,j}) = {i_2} then
    T ← {i_1, i_2}. Select a target packet X_{i_1,j_{i_1}} arbitrarily
      from such X_{i_1,j} packets, and set T_sel ← {i_1}.
    if there exists at least one X_{i_2,l} satisfying S(X_{i_2,l}) =
      {k, i_1} then
        Select the second target packet X_{i_2,j_{i_2}} arbitrarily
          from such X_{i_2,l} packets, and reset T_sel ← {i_1, i_2}.
    end if
else
    Move to the next sub-phase.
end if
```

Phase 4: Do the following steps:

```
T ← {1, 2, 3} and T_sel ← ∅.
for i = 1 to 3 do
    if there exists at least one X_{i,j} satisfying S(X_{i,j}) =
      {1, 2, 3}\i then
        Select a target X_{i,j_i} arbitrarily from such X_{i,j} and
          set T_sel ← T_sel ∪ {i}.
    end if
end for
if T_sel = ∅ then
    End of transmission.
end if
```

---

§ *The Analysis of The Capacity-Achieving PE Scheme*

Given any arbitrary rate vector (R_1, R_2, R_3) that is in the interior of the capacity outer bound of Proposition 1, we will prove in the following that the above PE scheme can be finished within n time slots when n is sufficiently large. Moreover, after termination of the above scheme, each d_k can successfully decode its desired packets $\{X_{k,j} : \forall j \in [nR_k]\}$. The proof is constructed by carefully analyzing the $S(\cdot)$ status of the packets after each sub-phases.

Phase 1.*k*: We first note that in average each X_{k,j_k} packet takes $\frac{1}{p_{\cup\{1,2,3\}}}$ time slots before it is received by at least one receiver, which in turn changes the corresponding $S(X_{k,j})$. By the law of large numbers, Phases 1.1, 1.2, and 1.3 thus continue for

$$\approx \frac{nR_1}{p_{\cup\{1,2,3\}}} \text{ time slots,} \tag{15}$$

$$\approx \frac{nR_2}{p_{\cup\{1,2,3\}}} \text{ time slots,} \tag{16}$$

$$\text{and } \approx \frac{nR_3}{p_{\cup\{1,2,3\}}} \text{ time slots, respectively.} \tag{17}$$

The above first-order approximation has precision $o(n)$ with respect to the codeword length n. Such first-order approximation will be used throughout this section.

Phase 2.1: We term all $X_{2,j}$ packets that have $S(X_{2,j}) = \{3\}$ the queue $Q_{2;3\overline{1}}$ packets. Symmetrically, all $X_{3,j}$ packets that have $S(X_{3,j}) = \{2\}$ are termed the queue $Q_{3;2\overline{1}}$ packets. In the end of Phase 1.3, all $Q_{2;3\overline{1}}$ packets were resulted/created in Phase 1.2 when a Phase-1.2 packet was received by d_3 only. Since Phase 1.2 lasts for (16) number of time slots, totally there are $\approx \frac{nR_2 p_{\{3\}\overline{\{1,2\}}}}{p_{\cup\{1,2,3\}}}$ such packets. Symmetrically, all $Q_{3;2\overline{1}}$ packets were resulted/created in Phase 1.3 when a Phase-1.3 packet was received by d_2 only. Totally there are $\approx \frac{nR_3 p_{\{2\}\overline{\{1,3\}}}}{p_{\cup\{1,2,3\}}}$ such packets.

One critical observation of the PE scheme is that when two target packets X_{2,j_2} and X_{3,j_3} are mixed together to generate $\mathbf{v}_{\text{tx}}$, each packet still keeps its own identity X_{2,j_2} and X_{3,j_3}, its own associated sets $S(X_{2,j_2})$ and $S(X_{3,j_3})$ and coding vectors $\mathbf{v}(X_{2,j_2})$ and $\mathbf{v}(X_{3,j_3})$. Even the decision whether to update $S(X)$ or $\mathbf{v}(X)$ is made separately (Line 2 of the UPDATE) for the two target packets X_{2,j_2} or X_{3,j_3}. Therefore, it is as if packets X_{2,j_2} or X_{3,j_3} are sharing time slots in a non-interfering way (like carpooling together). Following this observation, we can quantify the changes of the numbers of $Q_{2;3\overline{1}}$ packets and $Q_{3;2\overline{1}}$ packets during Phase 2.1.

We first take a closer look at when the status $S(X_{3,j_3})$ of a $Q_{3;2\overline{1}}$ packet X_{3,j_3} will evolve. By Line 4 of the UPDATE, the status $S(X_{3,j_3})$ evolves if and only if one of $\{d_1, d_3\}$ has received the coded packet in which X_{3,j_3} participates. Therefore, in average the status $S(X_{3,j_3})$ evolves after the corresponding X_{3,j_3} packet participates in $\frac{1}{p_{\cup\{1,3\}}}$ time slots. Since we have $\approx \frac{nR_3 p_{\{2\}\overline{\{1,3\}}}}{p_{\cup\{1,2,3\}}}$ number of $Q_{3;2\overline{1}}$ packets to begin with, it takes

$$\approx \frac{nR_3 p_{\{2\}\overline{\{1,3\}}}}{p_{\cup\{1,2,3\}}} \frac{1}{p_{\cup\{1,3\}}} = nR_3 \left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{18}$$

number of time slots to completely finish all $Q_{3;2\overline{1}}$ packets. By similar arguments, it takes

$$\approx \frac{nR_2 p_{\{3\}\overline{\{1,2\}}}}{p_{\cup\{1,2,3\}}} \frac{1}{p_{\cup\{1,2\}}} = nR_2 \left(\frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{19}$$

number of time slots to completely use up all $Q_{2;3\overline{1}}$ packets. Since we assume that d_2 dominates d_3, the dominance inequality in (14) implies that (19) is no smaller than (18). Therefore, for sufficiently large n, in Phase 2.1 we finish all $Q_{3;2\overline{1}}$ packets before exhausting the $Q_{2;3\overline{1}}$ packets. As a result, it takes roughly (18) number of time slots to finish Phase 2.1.

Phase 2.2: Similarly we term all $X_{1,j}$ packets that have $S(X_{1,j}) = \{3\}$ the queue $Q_{1;3\overline{2}}$ packets, all $X_{3,j}$ packets that have $S(X_{3,j}) = \{1\}$ the queue $Q_{3;1\overline{2}}$ packets.

By the same dominance-relationship-based arguments as used in the analysis of Phase 2.1, for sufficiently large n, in Phase 2.2 we finish the $Q_{3;1\overline{2}}$ packets before exhausting the $Q_{1;3\overline{2}}$ packets. And it takes roughly

$$\approx \frac{nR_3 p_{\{1\}\overline{\{2,3\}}}}{p_{\cup\{1,2,3\}}} \frac{1}{p_{\cup\{2,3\}}} = nR_3 \left(\frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{20}$$

number of time slots to finish Phase 2.2.

Phase 2.3: We term all $X_{1,j}$ packets that have $S(X_{1,j}) = \{2\}$ the queue $Q_{1;2\overline{3}}$ packets, all $X_{2,j}$ packets that have $S(X_{2,j}) = \{1\}$ the queue $Q_{2;1\overline{3}}$ packets. By similar arguments as used in Phases 2.1 and 2.2 and by the fact that d_1 dominates d_2, for sufficiently large n, in Phase 2.3 we finish the $Q_{2;1\overline{3}}$ packets before exhausting the $Q_{1;2\overline{3}}$ packets. And it takes roughly

$$\approx \frac{nR_2 p_{\{1\}\overline{\{2,3\}}}}{p_{\cup\{1,2,3\}}} \frac{1}{p_{\cup\{2,3\}}} = nR_2 \left(\frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{21}$$

number of time slots to finish Phase 2.3.

Phase 3: Before the analysis of Phase 3, we first summarize the status of all packets in the end of Phase 2.3. For d_3, all $X_{3,j}$ packets that have $S(X_{3,j}) = \emptyset$ have been used up in Phase 1.3. All $X_{3,j}$ packets that have $S(X_{3,j}) = \{1\}$ and $S(X_{3,j}) = \{2\}$ have been used up in Phases 2.2 and 2.1, respectively. As a result, all $X_{3,j}$ packets are either received by d_3 (i.e., having $3 \in S(X_{3,j})$) or have $S(X_{3,j}) = \{1, 2\}$. We term the latter type of $X_{3,j}$ packets the $Q_{3;12}$ packets. Recall the definition of $f_p(S\overline{T})$ in (4). We note that each Phase-1.3 transmission will generate a $Q_{3;12}$ packet when it is received by and only by d_1 and d_2, which happens with probability $p_{12\overline{3}}$. Since Phase 1.3 lasts for $\frac{nR_3}{p_{\cup\{1,2,3\}}}$ time slots, there are $\left(\frac{nR_3}{p_{\cup\{1,2,3\}}}\right) p_{12\overline{3}}$ number of $Q_{3;12}$ packets generated in Phase 1.3. Similarly, each Phase-2.1 transmission will generate a $Q_{3;12}$ packet when it is received by d_1 but not by d_3 (see Line 4 of the UPDATE subroutine), which happens with probability $f_p(1\overline{3})$. Since Phase 2.1 lasts for (18) number of time slots, there are $\left(\frac{nR_3 p_{2\overline{13}}}{p_{\cup\{1,2,3\}} p_{\cup\{1,3\}}}\right) f_p(1\overline{3})$ number of $Q_{3;12}$ packets generated in Phase 2.1. Similarly, there are $\left(\frac{nR_3 p_{1\overline{23}}}{p_{\cup\{1,2,3\}} p_{\cup\{2,3\}}}\right) f_p(2\overline{3})$ number of $Q_{3;12}$ packets generated in Phase 2.2. Totally, we have

$$\approx nR_3 \left(\frac{p_{12\overline{3}}}{p_{\cup\{1,2,3\}}} + \frac{p_{2\overline{13}}}{p_{\cup\{1,2,3\}}} \frac{f_p(1\overline{3})}{p_{\cup\{1,3\}}} + \frac{p_{1\overline{23}}}{p_{\cup\{1,2,3\}}} \frac{f_p(2\overline{3})}{p_{\cup\{2,3\}}} \right) \tag{22}$$

number of $Q_{3;12}$ packets in the beginning of Phase 3. We can further simplify (22) as

$$(22) = nR_3 p_3 \left(\frac{1}{p_3} - \frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{2,3\}}} + \frac{1}{p_{\cup\{1,2,3\}}} \right). \tag{23}$$

For d_2, all $X_{2,j}$ packets that have $S(X_{2,j}) = \emptyset$ have been used up in Phase 1.2. All $X_{2,j}$ packets that have $S(X_{2,j}) = \{1\}$ have been used up in Phase 2.3. As a result, all the $X_{2,j}$ packets must satisfy one of the following: (i) $X_{2,j}$ are received by d_2 (i.e., having $2 \in S(X_{2,j})$), or (ii) have $S(X_{2,j}) = \{3\}$ (i.e., the $Q_{2;3\overline{1}}$ packets), or (iii) have $S(X_{2,j}) = \{1, 3\}$, which are termed the $Q_{2;13}$ packets. By similar arguments as used for the d_3 packets, there are $\left(\frac{nR_2}{p_{\cup\{1,2,3\}}}\right) p_{3\overline{12}}$ number of $Q_{2;3\overline{1}}$ packets generated in Phase 1.2. However, Phase 2.1 uses/destroys some $Q_{2;3\overline{1}}$ packets. More explicitly, each Phase-2.1 transmission will destroy a $Q_{2;3\overline{1}}$ packet when it is received by one of $\{d_1, d_2\}$ (see Line 4 of the UPDATE subroutine), which happens with probability $p_{\cup\{1,2\}}$. Since Phase 2.1 lasts for (18) number of time slots, there are $\left(\frac{nR_3 p_{2\overline{13}}}{p_{\cup\{1,2,3\}} p_{\cup\{1,3\}}}\right) p_{\cup\{1,2\}}$ number of $Q_{2;3\overline{1}}$ packets destroyed in Phase 2.1. As a result, there are

$$\approx nR_2 \frac{p_{3\overline{12}}}{p_{\cup\{1,2,3\}}} - nR_3 \frac{p_{2\overline{13}}}{p_{\cup\{1,2,3\}}} \frac{p_{\cup\{1,2\}}}{p_{\cup\{1,3\}}} \tag{24}$$

number of $Q_{2;3\overline{1}}$ packets in the beginning of Phase 3.

For d_1, all $X_{1,j}$ packets that have $S(X_{1,j}) = \emptyset$ have been used up in Phase 1.1. As a result, all the $X_{1,j}$ packets must satisfy one of the following: (i) $X_{1,j}$ are received by d_1 (i.e., having $1 \in S(X_{1,j})$); (ii) have $S(X_{1,j}) = \{2\}$ (i.e., the $Q_{1;2\overline{3}}$ packets); (iii) have $S(X_{1,j}) = \{3\}$ (i.e., the $Q_{1;3\overline{2}}$ packets); or (iv) have $S(X_{1,j}) = \{2, 3\}$, which are termed the $Q_{1;23}$ packets. By similar arguments as used for the d_2 packets, there are

$$\approx nR_1 \frac{p_{2\overline{13}}}{p_{\cup\{1,2,3\}}} - nR_2 \frac{p_{1\overline{23}}}{p_{\cup\{1,2,3\}}} \frac{p_{\cup\{1,3\}}}{p_{\cup\{2,3\}}} \tag{25}$$

number of $Q_{1;2\overline{3}}$ packets in the beginning of Phase 3, where the first term is the number of $Q_{1;2\overline{3}}$ packets generated in Phase 1.1 and the second term corresponds to the number of $Q_{1;2\overline{3}}$ packets that are used up in Phase 2.3. Similarly, there are

$$\approx nR_1 \frac{p_{3\overline{12}}}{p_{\cup\{1,2,3\}}} - nR_3 \frac{p_{1\overline{23}}}{p_{\cup\{1,2,3\}}} \frac{p_{\cup\{1,2\}}}{p_{\cup\{2,3\}}} \tag{26}$$

number of $Q_{1;3\overline{2}}$ packets in the beginning of Phase 3, where the first term is the number of $Q_{1;3\overline{2}}$ packets generated in Phase 1.1 and the second term corresponds to the number of $Q_{1;3\overline{2}}$ packets that are used up in Phase 2.2.

We are now ready to analyze the status changes in Phase 3.

Phase 3.1: The goal of this subphase is to clean up the $Q_{2;3\overline{1}}$ packets that have not been used in Phase 2.1. By Line 4 of the UPDATE, the $S(X_{2,j_2})$ of a $Q_{2;3\overline{1}}$ packet X_{2,j_2} will change if and only if it is received by any one of $\{d_1, d_2\}$. Therefore, in average the status $S(X_{2,j_2})$ of a $Q_{2;3\overline{1}}$ packet X_{2,j_2} evolves after it participates in $\frac{1}{p_{\cup\{1,2\}}}$ number of time slots. Since we have (24) number of $Q_{2;3\overline{1}}$ packets to begin with, it will take

$$\approx \frac{(24)}{p_{\cup\{1,2\}}} = nR_2\left(\frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) - nR_3\left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) \quad (27)$$

number of time slots to finish Phase 3.1, where the equality follows from straightforward arithmetic simplification.

Phase 3.2: Similar to Phase 3.1, Phase 3.2 serves the role of cleaning up the $Q_{1;3\overline{2}}$ packets that have not been used in Phase 2.2. By Line 4 of the UPDATE, the $S(X_{1,j_1})$ of a $Q_{1;3\overline{2}}$ packet X_{1,j_1} will change if and only if it is received by any one of $\{d_1, d_2\}$. Therefore, in average the status $S(X_{1,j_1})$ of a $Q_{1;3\overline{2}}$ packet X_{1,j_1} evolves after it participates in $\frac{1}{p_{\cup\{1,2\}}}$ number of time slots. Since we have (26) number of $Q_{1;3\overline{2}}$ packets to begin with, it will take

$$\approx \frac{(26)}{p_{\cup\{1,2\}}} = nR_1\left(\frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) - nR_3\left(\frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) \quad (28)$$

number of time slots to finish Phase 3.2.

Phase 3.3: Similar to Phases 3.1 and 3.2, Phase 3.3 serves the role of cleaning up the $Q_{1;2\overline{3}}$ packets that have not been used in Phase 2.3. By similar analysis, it will take

$$\approx \frac{(25)}{p_{\cup\{1,3\}}} = nR_1\left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) - nR_2\left(\frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) \quad (29)$$

number of time slots to finish Phase 3.3.

Phase 4: We first summarize the status of all the packets in the end of Phase 3.3. For d_3, all the $X_{3,j}$ packets are either received by d_3 (i.e., having $3 \in S(X_{3,j})$) or have $S(X_{3,j}) = \{1, 2\}$, i.e., the $Q_{3;12}$ packets. By Line 4 of the UPDATE, the $S(X_{3,j_3})$ of a $Q_{3;12}$ packet X_{3,j_3} will change if and only if it is received by d_3. Therefore, each transmission in Phases 3.1 and 3.2 will in average use up p_3 number of participated $Q_{3;12}$ packets. Since Phases 3.1 and 3.2 have duration (27) and (28), respectively, in the end of Phase 3.3 the total number of $Q_{3;12}$ packets thus becomes

$$\approx (\text{Eq.}(23) - p_3 \cdot \text{Eq.}(27) - p_3 \cdot \text{Eq.}(28))^+, \quad (30)$$

where the first term is the number of $Q_{3;12}$ packets in the end of Phase 2, the second and the third terms are the numbers of $Q_{3;12}$ packets used/destroyed in Phases 3.1 and 3.2, respectively, and $(\cdot)^+ = \max(\cdot, 0)$ is the projection to the non-negative reals. The reason we need the $(\cdot)^+$ operation is that when there is no more $Q_{3,;12}$ packet to select from, we will stop selecting $Q_{3;12}$ packets and the actually selected set T_{sel} will be a strict subset of T (see Lines 2 and 4 of Phase 3).

For d_2, all $X_{2,j}$ packets that have $S(X_{2,j}) = \emptyset$ and $S(X_{2,j}) = \{1\}$ have been used up in Phases 1.2 and 2.3, respectively. All $X_{2,j}$ packets that have $S(X_{2,j}) = \{3\}$ have been used up in Phases 2.1 and 3.1. As a result, all the $X_{2,j}$ packets are either received by d_2 (i.e., having $2 \in S(X_{2,j})$) or have $S(X_{2,j}) = \{1, 3\}$, i.e, the $Q_{2;13}$ packets. By Line 4 of the UPDATE, the $S(X_{2,j_2})$ of a $Q_{2;13}$ packet X_{2,j_2} will change if and only if it is received by d_2. Therefore, each transmission in Phase 3.3 will in average use up p_2 number of participated $Q_{2;13}$ packets. Since Phase 3.3 has duration (29), in the end of Phase 3.3 the total number of $Q_{2;13}$ packets becomes

$$\approx \big(p_{13\overline{2}} \cdot \text{Eq.}(16) + f_p(3\overline{2}) \cdot \text{Eq.}(21) + f_p(1\overline{2}) \cdot \text{Eq.}(18) + f_p(1\overline{2}) \cdot \text{Eq.}(27) - p_2 \cdot \text{Eq.}(29)\big)^+ \quad (31)$$

where the first four terms are the numbers of $Q_{2;13}$ packets generated in Phases 1.2, 2.3, 2.1, and 3.1, respectively, and the fifth term is the number of $Q_{2;13}$ packets used/destroyed in Phase 3.3. The summation of the first four terms of (31) can be further simplified to:

$$\text{Summation of the first four terms of (31)} = nR_2p_2\left(\frac{1}{p_2} - \frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{2,3\}}} + \frac{1}{p_{\cup\{1,2,3\}}}\right). \quad (32)$$

For d_1, all $X_{1,j}$ packets that have $S(X_{1,j}) = \emptyset$, $S(X_{1,j}) = \{2\}$, and $S(X_{1,j}) = \{3\}$ have been used up in Phases 1.1, 2.3+3.3, and 2.2+3.2, respectively. As a result, all the $X_{1,j}$ packets are either received by d_1 (i.e., having $1 \in S(X_{1,j})$) or have $S(X_{1,j}) = \{2, 3\}$, i.e., the $Q_{1;23}$ packets. By similar computation, in the end of Phase 3.3 the total number of $Q_{1;23}$ packets is

$$\approx nR_1\left(\frac{p_{23\overline{1}}}{p_{\cup\{1,2,3\}}} + \frac{p_{2\overline{13}}}{p_{\cup\{1,2,3\}}}\frac{f_p(3\overline{1})}{p_{\cup\{1,3\}}} + \frac{p_{3\overline{12}}}{p_{\cup\{1,2,3\}}}\frac{f_p(2\overline{1})}{p_{\cup\{1,2\}}}\right), \quad (33)$$

where the first, second, and the third terms correspond to the numbers of $Q_{1;23}$ packets generated in Phase 1.1, 2.3+3.3, and 2.2+3.2, respectively. We can further simplify (33) as

$$(33) = nR_1p_1\left(\frac{1}{p_1} - \frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,3\}}} + \frac{1}{p_{\cup\{1,2,3\}}}\right). \quad (34)$$

The goal of Phase 4 is to clean up the remaining packets. Since in average the status $S(X_{i,j})$ of a $Q_{i;\{1,2,3\}\backslash i}$ packet $X_{i,j}$ evolves after it participates in $\frac{1}{p_i}$ number of time slots, Phase 4 thus takes

$$\approx \max\left(\frac{\text{Eq.}(34)}{p_1}, \frac{\text{Eq.}(31)}{p_2}, \frac{\text{Eq.}(30)}{p_3}\right). \quad (35)$$

number of time slots to finish. Once we finish Phase 4, for all $k \in \{1, 2, 3\}$, $j \in [nR_k]$, we have $k \in S(X_{k,j})$. By Lemma 3 of any generic PE scheme, all d_k can decode their desired packets with close-to-one probability.

What remains to be shown is that with sufficiently large n, we can finish transmissions of all four phases within n time slots. That is, we need to prove that

$$(15) + (16) + (17) + (18) + (20) + (21) + (27) + (28) + (29) + (35) \leq n. \quad (36)$$

The summation of the first nine terms of the left-hand side of (36) can be simplified to

$$A_{1.1\text{–}3.3} \triangleq nR_1 \left(\frac{1}{p_{\cup\{1,2\}}} + \frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) + nR_2 \frac{1}{p_{\cup\{1,2\}}} + nR_3 \frac{1}{p_{\cup\{1,2,3\}}},$$

where $A_{1.1\text{–}3.3}$ is the total number of time slots in Phases 1.1 to 3.3. Since (35) is the maximum of three terms, proving (36) is thus equivalent to proving that the following three inequalities hold simultaneously.

$$A_{1.1\text{–}3.3} + \frac{(34)}{p_1} \leq n,$$
$$A_{1.1\text{–}3.3} + \frac{(31)}{p_2} \leq n,$$
$$\text{and } A_{1.1\text{–}3.3} + \frac{(30)}{p_3} \leq n.$$

With straightforward simplification, proving the above three inequalities is equivalent to proving

$$\frac{nR_1}{p_1} + \frac{nR_2}{p_{\cup\{1,2\}}} + \frac{nR_3}{p_{\cup\{1,2,3\}}} \leq n,$$
$$\frac{nR_1}{p_{\cup\{1,2\}}} + \frac{nR_2}{p_2} + \frac{nR_3}{p_{\cup\{1,2,3\}}} \leq n,$$
$$\text{and } \frac{nR_1}{p_{\cup\{1,3\}}} + \frac{nR_2}{p_{\cup\{1,2,3\}}} + \frac{nR_3}{p_3} \leq n.$$

Since the expressions of the numbers of time slots in Phase 1.1 to Phase 4: (15), (16), (17), (18), (20), (21), (27), (28), (29), and (35) are of precision $o(n)$, the last three inequalities hold with arbitrarily close-to-one probability for sufficiently large n for any rate vector (R_1, R_2, R_3) in the interior of the capacity outer bound in Proposition 1. The proof of Proposition 2 is thus complete.

B. Achieving the Capacity of 1-to-3 Broadcast PECs With COF — High-Level Discussion

As discussed in Section V-A, one advantage of a PE scheme is that although different packets X_{k,j_k} and X_{i,j_i} with $k \neq i$ may be mixed together, the corresponding evolution of X_{k,j_k} (the changes of $S(X_{k,j_k})$ and $\mathbf{v}(X_{k,j_k})$) is independent from the evolution of X_{i,j_i} and can thus be easily traced. Also by Lemma 2, two different packets X_{k,j_k} and X_{i,j_i} can share the same time slot without interfering each other as long as $i \in S(X_{k,j_k})$ and $k \in S(X_{i,j_i})$. These two observations enable us to relate the achievability problem of a PE scheme to the following "time slot packing problem."

Let us focus on the (s, d_1) session. For any $X_{1,j}$ packet, initially $S(X_{1,j}) = \emptyset$. Then as time proceeds, each $X_{1,j}$ starts to participate in packet transmission. The corresponding $S(X_{1,j})$ evolves to different values, depending on the set of destinations that receive the transmitted packet in which $X_{1,j}$ participates. Since in this subsection we focus mostly on $S(X_{1,j})$, we sometimes use $S(X)$ as shorthand if it is unambiguous from the context. Fig. 4 describes how $S(X)$ evolves through different values. In Fig. 4, we use circles to represent the five different states according to the $S(X)$ value.

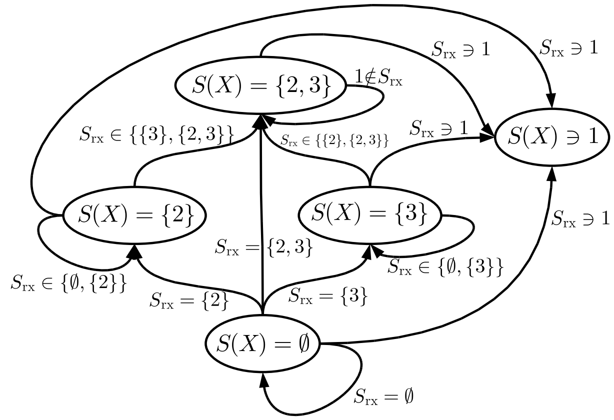


Fig. 4. The state transition diagram for destination d_1 when applying the packet evolution scheme to a 1-to-3 broadcast PEC.

The receiving set S_{rx}, the set of destinations who successfully receive the transmitted coded packet, decides the transition between different states. In Fig. 4, we thus mark each transition arrow (between different states) by the value(s) of S_{rx} that enables the transition. For example, by Line 4 of the UPDATE, when the initial state is $S(X) = \emptyset$, if the receiving set $S_{\text{rx}} \ni 1$, then the new set satisfies $S(X) \ni 1$. Similarly, when the initial state is $S(X) = \emptyset$, if $S_{\text{rx}} = \{2,3\}$, then the new $S(X)$ becomes $S(X) = \{2,3\}$. (Note that the corresponding $\mathbf{v}(X_{1,j})$ also evolves over time to maintain the non-interfering property in Lemma 2, which is not illustrated in Fig. 4.)

Since $S(X_{1,j}) \ni 1$ if and only if d_1 receives $X_{1,j}$, it thus takes $\frac{nR_1}{p_1}$ *logical time slots* to finish the transmission of nR_1 information packets. On the other hand, some logical time slots for the (s, d_1) session can be "packed/shared" jointly with the logical time slots for the (s, d_k) session, $k \neq 1$, or, equivalently, one physical time slot can serve two sessions simultaneously. In the following, we quantify how many logical time slots of the (s, d_1) session are *compatible* to those of other sessions. For any $S_0 \in 2^{\{2,3\}}$, let $A_{1;S_0}$ denote the number of logical time slots (out of the total $\frac{nR_1}{p_1}$ time slots) such that during those time slots, the transmitted $X_{1,j}$ has $S(X_{1,j}) = S_0$. Initially, there are nR_1 packets $X_{1,j}$. If any one of $\{d_1, d_2, d_3\}$ receives the transmitted packet (equivalently $S_{\text{rx}} \neq \emptyset$), we have $S(X_{1,j}) \leftarrow S_{\text{rx}}$, which is no longer an empty set. Therefore, each $X_{1,j}$ contributes to $\frac{1}{p_{\cup\{1,2,3\}}}$ logical time slots with $S(X_{1,j}) = \emptyset$. We thus have

$$A_{1;\emptyset} = nR_1 \left(\frac{1}{p_{\cup\{1,2,3\}}} \right). \tag{37}$$

We also note that during the evolution process of $X_{1,j}$, if and only if one of $\{d_1, d_3\}$ receives the transmitted packet (equivalently $S_{\text{rx}} \cap \{1,3\} \neq \emptyset$), then $S(X)$ value will move from one of the two states "$S(X) = \emptyset$" and "$S(X) = \{2\}$" to one of the three states "$S(X) = \{3\}$," "$S(X) = \{2,3\}$," and "$S(X) \ni 1$." Therefore, each $X_{1,j}$ contributes to $\frac{1}{p_{\cup\{1,3\}}}$ logical time slots during which we either have $S(X_{1,j}) = \emptyset$ or $S(X_{1,j}) = \{2\}$. By the above reasoning, we have

$$A_{1;\{2\}} + A_{1;\emptyset} = nR_1 \left(\frac{1}{p_{\cup\{1,3\}}} \right). \tag{38}$$

Similarly, during the evolution process of $X_{1,j}$, if and only if one of $\{d_1, d_2\}$ receives the transmitted packet (equivalently $S_{\text{rx}} \cap \{1,2\} \neq \emptyset$), then $S(X)$ value will move from one of the two states "$S(X) = \emptyset$" and "$S(X) = \{3\}$" to one of the three states "$S(X) = \{2\}$," "$S(X) = \{2,3\}$," and "$S(X) \ni 1$." Therefore, each $X_{1,j}$ contributes to $\frac{1}{p_{\cup\{1,2\}}}$ logical time slots during which either $S(X_{1,j}) = \emptyset$ or $S(X_{1,j}) = \{3\}$. By the above reasoning, we have

$$A_{1;\{3\}} + A_{1;\emptyset} = nR_1 \left(\frac{1}{p_{\cup\{1,2\}}} \right). \tag{39}$$

Before $S(X)$ evolves to the state "$S(X) \ni 1$," any logical time slot contributed by such an X must have one of the following four states: "$S(X) = \emptyset$," "$S(X) = \{2\}$," "$S(X) = \{3\}$," and "$S(X) = \{2,3\}$." As a result, we must have

$$A_{1;\{2,3\}} + A_{1;\{2\}} + A_{1;\{3\}} + A_{1;\emptyset} = nR_1 \left(\frac{1}{p_1} \right). \tag{40}$$

Solving (37), (38), (39), and (40), we have

$$A_{1;\emptyset} = nR_1 \left(\frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{41}$$

$$A_{1;\{2\}} = nR_1 \left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{42}$$

$$A_{1;\{3\}} = nR_1 \left(\frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{43}$$

$$A_{1;\{2,3\}} = nR_1 \left(\frac{1}{p_1} - \frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,3\}}} + \frac{1}{p_{\cup\{1,2,3\}}} \right). \tag{44}$$

We can also define $A_{k;S_0}$ as the number of logical time slots of the (s, d_k) session with $S(X_{k,j_k}) = S_0$. By similar derivation arguments, for any distinct indices k, i_1, and i_2 in $\{1,2,3\}$, we have

$$A_{k;\emptyset} = nR_k \left(\frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{45}$$

$$A_{k;\{i_1\}} = nR_k \left(\frac{1}{p_{\cup(\{1,2,3\}\backslash i_1)}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \tag{46}$$

$$A_{k;\{i_1,i_2\}} = nR_k \left(\frac{1}{p_k} - \frac{1}{p_{\cup\{k,i_1\}}} - \frac{1}{p_{\cup\{k,i_2\}}} + \frac{1}{p_{\cup\{1,2,3\}}} \right). \tag{47}$$

The achievability problem of a PE scheme thus relates to the following time slot packing problem.

> Consider 12 types of logical time slots and each type is denoted by $(k; S_0)$ for some $k \in \{1,2,3\}$ and $S_0 \in 2^{(\{1,2,3\}\backslash k)}$. The numbers of logical time slots of each type are described in (45) to (47). Two logical time slots of types $(k_1; S_1)$ and $(k_2; S_2)$ are *compatible* if $k_1 \neq k_2$, $k_1 \in S_2$, and $k_2 \in S_1$. Any compatible logical time slots can be packed together in the same physical time slot. The time slot packing problem is thus: *Can we pack all the logical time slots within n physical time slots?*

The detailed 4-phase PE scheme in Section V-A thus corresponds to the time-slot-packing policy depicted in Fig. 5. Namely, we first use Phases 1.1 to 1.3 send all the logical time slots that cannot be packed with any other logical time slots. Totally, it takes $A_{1;\emptyset} + A_{2;\emptyset} + A_{3;\emptyset}$ number of time slots to finish Phases 1.1 to 1.3. We then use Phases 2.1 to 2.3 to pack those logical time slots that can be packed with exactly one other logical time slot from a different session. By the assumption that d_1 dominates d_2 and d_3, and d_2 dominates d_3, we have $A_{1;\{2\}} \geq A_{2;\{1\}}$, $A_{1;\{3\}} \geq A_{3;\{1\}}$, and $A_{2;\{3\}} \geq A_{3;\{2\}}$. Therefore, it takes $A_{3;\{2\}} + A_{3;\{1\}} + A_{2;\{1\}}$ number of physical time slots to finish Phases 2.1 to 2.3.

Phases 3.1 to 3.3 are to clean up the remaining logical time slots of types $(2; \{3\})$, $(1; \{3\})$, and $(1; \{2\})$. We notice that in Phase 3.1 when sending a logical time slot of type $(2; \{3\})$, there is no type-$(3, \{2\})$ logical time slot that can be packed together. On the other hand, there are still some type-$(3, \{1,2\})$ logical time slots, which can also be packed with the logical time slots of the (s, d_2) session. Therefore, when we send a logical time slot of type $(2; \{3\})$, the optimal way is to pack it with a type-$(3, \{1,2\})$ logical time slots together as illustrated in Phase 3.1 of Fig. 5. It is worth emphasizing that although those type-$(3, \{1,2\})$ logical time slots may later be packed with two other logical time slots simultaneously, there is no point to save the type-$(3, \{1,2\})$ logical time slots for future time slot packing. The reason is that when Phase 3.1 cleans up the remaining type-$(2; \{3\})$ logical time slots, it actually provides a zero-cost free ride for any logical time slot that is compatible to a type-$(2; \{3\})$ logical time slot. Therefore, piggybacking a type-$(2; \{3\})$ logical time slot with a type-$(3, \{1,2\})$ logical time slot is optimal. Similarly, we also take advantage of the free ride by packing logical time slots of type-$(1; \{3\})$ with that of type-$(3; \{1,2\})$ in Phase 3.2, and by packing logical time slots of type-$(1; \{2\})$ with that of type-$(2; \{1,3\})$ in Phase 3.3. It thus takes

$$(A_{2;\{3\}} - A_{3;\{2\}}) + (A_{1;\{3\}} - A_{3;\{1\}}) + (A_{1;\{2\}} - A_{2;\{1\}})$$

number of time slots to finish Phases 3.1 to 3.3.

In Phase 4, we clean up and pack together all the remaining logical time slots of types $(1; \{2,3\})$, $(2; \{1,3\})$, and $(3; \{1,2\})$. We thus need

$$\max\big((A_{3;\{1,2\}} - (A_{2;\{3\}} - A_{3;\{2\}}) - (A_{1;\{3\}} - A_{3;\{1\}}))^+, \\ (A_{2;\{1,3\}} - (A_{1;\{2\}} - A_{2;\{1\}}))^+, A_{1;\{2,3\}}\big) \tag{48}$$

number of time slots to finish Phase 4. Depending on which of the three terms in (48) is the largest, the total number of physical time slots is one of the following three expressions:

$$A_{3;\emptyset} + A_{3;\{1\}} + A_{3;\{2\}} + A_{3;\{1,2\}} + A_{1;\emptyset} + A_{1;\{2\}} + A_{2;\emptyset},$$
$$A_{2;\emptyset} + A_{2;\{1\}} + A_{2;\{3\}} + A_{2;\{1,3\}} + A_{1;\emptyset} + A_{1;\{3\}} + A_{3;\emptyset},$$
$$\text{or } A_{1;\emptyset} + A_{1;\{2\}} + A_{1;\{3\}} + A_{1;\{2,3\}} + A_{2;\emptyset} + A_{2;\{3\}} + A_{3;\emptyset}.$$

By (45) to (47), one can easily check that all three equations are less than n for any (R_1, R_2, R_3) in the interior of the outer bound of Proposition 1, which answers the time-slot-packing problem in an affirmative way. One can also show that the packing policy in Fig. 5, motivated by the capacity-achieving PE scheme in Section V-A, is indeed the tightest among all packing policies.

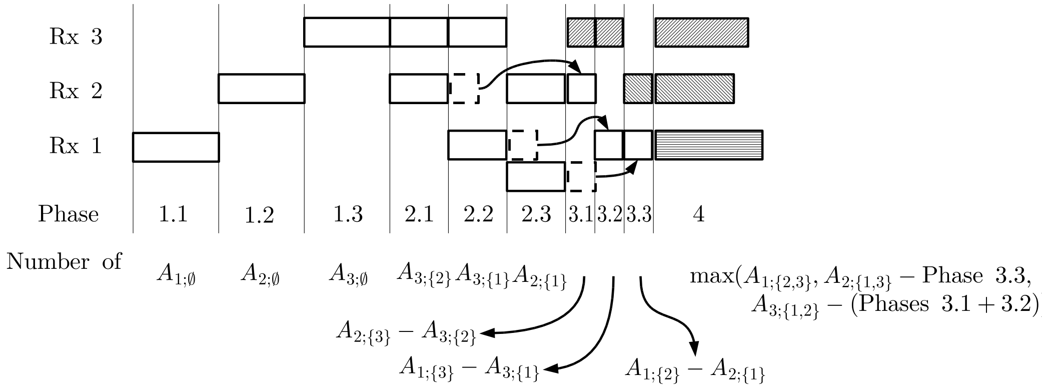


Fig. 5. The time-slot packing policy that corresponds to the 4-Phase solution for 1-to-3 broadcast PECs. The shaded rectangles represent the logical time slots of types $(1;\{2,3\})$, $(2;\{1,3\})$, and $(3;\{1,2\})$.

C. The Achievability Results of General 1-to-K Broadcast PECs With COF

In Section V-B, we show that the capacity-achieving PE scheme for a 1-to-3 PEC can be viewed as a tightest solution to the time-slot-packing problem. However, the converse may not hold due to the *causality constraint* of the PE scheme.

One major difference between the tightest solution of the time-slot-packing problem in Fig. 5 and the detailed PE scheme in Section V-A is that for the former, we can pack the time slots in any order. There is no need to first pack those logical time slots that cannot be shared with any other time slots. Any packing order will result in the same amount of physical time slots in the end. On the other hand, for the PE scheme it is critical to perform the 4 phases (10 sub-phases) in sequence since many packets used in the later phases are generated by the previous phases. For example, all the packets in Phases 2 to 4 are generated in Phases 1.1 to 1.3. Therefore it is imperative to conduct Phase 1 first before Phases 2 to 4. Similarly, the $Q_{3;\{1,2\}}$ packets used in Phases 3.1 and 3.2 are generated in Phases 1.3, 2.1, and 2.2. Therefore, the number of $Q_{3;\{1,2\}}$ packets in the end of Phase 1.3 alone (without those generated in Phases 2.1 and 2.2) may not be sufficient for mixing with $Q_{1;\{3\}}$ packets. As a result, it can be suboptimal to perform Phase 3.1 before Phases 2.1 and 2.2.

The causality constraints for a 1-to-K PEC complicate the design and analysis of PE schemes. In the following, we thus consider only *acyclic construction of PE schemes*, which allows tractable analysis but at the cost of potentially being throughput suboptimal.[6]

The main feature of the proposed PE scheme is that we choose the mixing set T in *an acyclic fashion.* For comparison, the T parameters used in the capacity-achieving PE scheme of Section V-A are $\{1\}$, $\{2\}$, $\{3\}$, $\{2,3\}$, $\{1,3\}$, $\{1,2\}$, $\{2,3\}$, $\{1,3\}$, $\{1,2\}$, and $\{1,2,3\}$ in Phases 1.1 to 4, respectively. We notice that $T=\{1,2\}$ is visited twice in Phases 2.3 and 3.3. We thus call the capacity-achieving PE scheme a *cyclic* PE scheme. For an acyclic PE scheme, we never revisit any T value during all the phases.

[6]In Section VI-E, it was shown numerically that for most PEC parameter values, the proposed acyclic PE scheme is sufficient to achieve the capacity.

Consider any rate vector $(R_1,\cdots,R_K)$ satisfying the inner bound in Proposition 3. We describe the corresponding achievability PE scheme by specifying how to choose the sets T, T_{sel}, and how to choose the target packets X_{k,j_k}. Since $x_\emptyset$ only participates in the left-hand side of (8), we can assume $x_\emptyset = 0$ without loss of generality. For the following discussion, we further assume that the variables $\{x_S\}$ and $\{w_{k;S\to T}\}$ in Proposition 3 satisfy (10) and (11) with equality. In the end of this subsection, we discuss how to relax this assumption.

For simplicity, we also assume a sufficiently large n so that the proposed PE scheme can be described and analyzed based on the corresponding first order approximation.

§ A PE Scheme For The Achievability Results of General 1-to-K PECs

The scheme contains 2^K phases indexed by $T\in 2^{[K]}$. We go through Phase T sequentially according to the *cardinality-compatible* total ordering $\prec$. That is, Phase T_1 precedes Phase T_2 if and only if $T_1\prec T_2$.

Phase T: Phase T lasts for nx_T time slots. Since $x_\emptyset=0$, we simply skip Phase $\emptyset$ and focus on the discussion for $T\neq\emptyset$. We then describe how to choose the actually selected set $T_{\text{sel}}\subseteq T$ and the target packets X_{k,j_k}, $k\in T_{\text{sel}}$. In the beginning of Phase T, set $T_{\text{sel}}\leftarrow T$.

For each time slot, each destination d_k chooses its target packet X_{k,j_k} independently from the choices of other destinations d_i, $i\in T_{\text{sel}}\backslash k$. In the following, we describe the subroutine that chooses X_{k,j_k} for a fixed $k\in T_{\text{sel}}$. The subroutine contains $2^{K-|T|}$ stages, indexed by a set $S\in 2^{[K]}$ satisfying $(T\backslash k)\subseteq S\subseteq([K]\backslash k)$. Again, we move sequentially through the stages according to the cardinality-compatible total ordering $\prec$. That is, Stage S_1 precedes Stage S_2 if and only if $S_1\prec S_2$.

Stage S: Stage S lasts for $nw_{k;S\to(T\backslash k)}$ time slots. Throughout this stage, choose the target packet X_{k,j_k} arbitrarily from all $X_{k,j}$ having $S(X_{k,j})=S$. After $nw_{k;S\to(T\backslash k)}$ time slots, we move to the next stage.

Once we have finished all stages for d_k (i.e. after finishing Stage $([K]\backslash k)$), we simply stop choosing any $X_{k,j}$ in this phase by setting $T_{\text{sel}}\leftarrow T_{\text{sel}}\backslash k$.

§ *The Analysis of The Achievability PE Scheme*

We first prove that for sufficiently large n, the above PE scheme is always feasible.

- It is guaranteed by (8) that the above PE scheme can finish all phases within n time slots. It is also guaranteed by (9) that for each Phase T, each d_k, $k \in T$ can finish all its stages within the allocated nx_T time slots for Phase T.
- We also need to show that for any $k \in T$, during the corresponding Stage S of Phase T there is always at least one $X_{k,j}$ having $S(X_{k,j}) = S$, from which a target packet X_{k,j_k} will be chosen. We term those packets the $Q_{k;S}$ packets.

The last bullet is proven as follows. Consider the beginning of Stage S in Phase $(T \cup \{k\})$ for some $k \notin T$, where they satisfy $T \subseteq S \subseteq ([K]\backslash k)$. Assume by induction that the proposed PE scheme is feasible from time 0 to the beginning of Stage S in Phase $(T \cup \{k\})$. Consider two cases depending on whether $T = S$ or not.

Case 1: $T \neq S$. By Line 4 of the UPDATE, the status $S(X_{k,j})$ of a $Q_{k;S}$ packet $X_{k,j}$ evolves if and only if one of the d_i with $i \in ([K]\backslash S)$ receives the coded transmission. Therefore, sending $Q_{k;S}$ packets for $nw_{k;S\to T}$ number of time slots in Stage S of Phase $(T \cup \{k\})$ will consume $nw_{k;S\to T} \cdot p_{\cup([K]\backslash S)}$ number of $Q_{k;S}$ packets. Therefore, we need to prove that in the beginning of Stage S of Phase $(T \cup \{k\})$, there are no fewer than $nw_{k;S\to T} \cdot p_{\cup([K]\backslash S)}$ number of $Q_{k;S}$ packets.

To that end, we first quantify the total number of $Q_{k;S}$ packets that have been created before Stage S of Phase $(T \cup \{k\})$. We notice that the $Q_{k;S}$ packets can be created either in a prior Stage S_1 of the current Phase $(T \cup \{k\})$ with $S_1 \prec S$; or in a prior Phase $(T_1 \cup \{k\})$ with $(T_1 \cup \{k\}) \prec (T \cup \{k\})$. For the former case, for each time slot in which we transmit a $Q_{k;S_1}$ packet in Phase $(T \cup \{k\})$, there is some chance that the packet will evolve into a $Q_{k;S}$ packet. More explicitly, by Line 4 of the UPDATE, a $Q_{k;S_1}$ packet in Phase $(T \cup \{k\})$ evolves into a $Q_{k;S}$ packet if and only if the packet is received by all d_i with $i \in (S\backslash T)$ and not by any d_i with $i \in ([K]\backslash S)$. As a result, in average each such transmission will create $f_p((S\backslash T)\overline{([K]\backslash S)})$ number of $Q_{k;S}$ packets. Since we previously sent $Q_{k;S_1}$ packets for a total $nw_{k;S_1\to T}$ number of time slots, the first term of the right-hand side of (12) is indeed the normalized number of $Q_{k;S}$ packets created in a prior Stage S_1 of the current Phase $(T \cup \{k\})$.

For the latter case, for each time slot in which we transmit a $Q_{k;S_1}$ packet in Stage S_1 of Phase $(T_1 \cup \{k\})$, there is some chance that the packet will evolve into a $Q_{k;S}$ packet. More explicitly, by Line 4 of the UPDATE, a $Q_{k;S_1}$ packet in Phase $(T_1 \cup \{k\})$ evolves into a $Q_{k;S}$ packet if and only if

$$\begin{cases} T_1 \subseteq S \\ S_{\text{rx}} \nsubseteq S_1 \\ (S\backslash T_1) \subseteq S_{\text{rx}} \\ ([K]\backslash S) \subseteq ([K]\backslash S_{\text{rx}}) \end{cases} \text{or equivalently} \quad \begin{cases} T_1 \subseteq S \\ S \nsubseteq S_1 \\ (S\backslash T_1) \subseteq S_{\text{rx}} \\ ([K]\backslash S) \subseteq ([K]\backslash S_{\text{rx}}) \end{cases},$$

where the above equivalence relationship follows from the fact that when considering Stage S_1 of Phase $(T_1 \cup \{k\})$ we implicitly imply $T_1 \subseteq S_1 \subseteq ([K]\backslash k)$. Therefore, for any (S_1, T_1) pair satisfying $T_1 \subseteq S$ and $S \nsubseteq S_1$, a $Q_{k;S_1}$ packet in Stage S_1 of Phase $(T_1 \cup \{k\})$ will have $f_p((S\backslash T_1)\overline{([K]\backslash S)})$ probability to evolve into a $Q_{k;S}$ packet. Since we previously sent $Q_{k;S_1}$ packets in Stage S_1 of Phase $(T_1 \cup \{k\})$ for a total $nw_{k;S_1\to T_1}$ number of time slots, the second term of the right-hand side of (12) is indeed the normalized number of $Q_{k;S}$ packets created during prior phases. In sum, the right-hand side of (12) is the normalized total number of $Q_{k;S}$ packets created before Stage S of Phase $(T \cup \{k\})$.

On the other hand, among all $Q_{k;S}$ packets created, some of them may have been used up in prior phases. More explicitly, $Q_{k;S}$ packets may be consumed in Stage S of a prior Phase $(T_1 \cup \{k\})$ such that $T_1 \subseteq S$ and $(T_1 \cup \{k\}) \prec (T \cup \{k\})$. By the same arguments as when we previously quantify the number of $Q_{k;S}$ packets to be consumed in Phase $(T \cup \{k\})$, Stage S of a prior Phase $(T_1 \cup \{k\})$ will consume $n \cdot w_{k;S\to T_1} \cdot p_{\cup([K]\backslash S)}$ number of $Q_{k;S}$ packets. The second term of the left-hand side of (12) thus represents the normalized total number of $Q_{k;S}$ packets that have been consumed before the beginning of Stage S of Phase $(T \cup \{k\})$. As a result, (12) guarantees that in the beginning of Stage S of Phase $(T \cup \{k\})$ with $T \neq S$, there are no fewer than $nw_{k;S\to T} \cdot p_{\cup([K]\backslash S)}$ number of $Q_{k;S}$ packets. There is thus always a $X_{k,j}$ packet with $S(X_{k,j}) = S$ to choose from throughout Stage S of Phase $(T \cup \{k\})$ for any $T \neq S$.

Case 2: $T = S$. We have two subcases depending on whether $S = \emptyset$. **Case 2.1:** $T = S = \emptyset$. In the beginning of Stage $\emptyset$ of Phase $(\emptyset \cup \{k\})$, no $X_{k,j}$ has ever been transmitted. Therefore, all $X_{k,j}$ have $S(X_{k,j}) = \emptyset$, and we have nR_k number of $Q_{k;\emptyset}$ packets. By similar reasons as in the previous analysis, Stage $\emptyset$ of Phase $(\emptyset \cup \{k\})$ will consume $nw_{k;\emptyset\to\emptyset} \cdot p_{\cup[K]}$ number of $Q_{k;\emptyset}$ packets. With the assumption that (10) is satisfied with equality, it is thus guaranteed that we have enough $Q_{k;\emptyset}$ packets for Stage $\emptyset$ of Phase $(\emptyset \cup \{k\})$.

Case 2.1: $T = S \neq \emptyset$. Similar to Case 1, we again quantify the total number of $Q_{k;S}$ packets that have been created before Stage S of Phase $(T \cup \{k\})$. However, with the assumption that $T = S \neq \emptyset$, the first term of the right-hand side of (12) no longer exists since the two conditions $S = T \subseteq S_1$ and $S_1 \prec S$ cannot hold simultaneously. Therefore, there is only a single term in the right-hand side of (11) that corresponds to the total number of $Q_{k;S}$ packets that have been created. In addition, the assumption $T = S \neq \emptyset$ enables us to combine the two terms of the left-hand side of (12) into a single term as in the left-hand side of (11). Since we assume that (11) is satisfied with equality, it is thus guaranteed that we have enough $Q_{k;S}$ packets for Stage S of Phase $(T \cup \{k\})$ when $T = S \neq \emptyset$. The discussion of Cases 1, 2.1, and 2.1 guarantees that the proposed PE scheme is feasible for sufficiently large n.

We now prove the decodability of the proposed PE scheme.

- When the proposed PE terminates, all d_k can decode their desired $X_{k,j}$ packets with close-to-one probability when sufficiently large n and $\mathsf{GF}(q)$ are used.

Consider destination d_k. Recall the assumption that (10) and (11) are satisfied with equality. The decodability can be proved by noticing that in the discussion of Cases 2.1 and 2.2 of the feasibility analysis, all $Q_{k;S}$ packets are used up completely after Stage S of Phase $(S \cup \{k\})$ for all $S \in 2^{([K]\backslash k)}$. We then notice that any subsequent Stage S_1 of the current Phase $(S \cup \{k\})$ must satisfy $S_1 \neq S$ and $S_1 \supseteq S$. As a result, by Line 4 of the UPDATE, no $Q_{k;S}$ packets will be generated in any subsequent Stage S_1 of the current Phase $(S \cup \{k\})$.

In the following, we prove by contradiction that no $X_{k,j}$ packet in any Stage S_1 of a subsequent Phase T can have its status $S(X_{k,j})$ evolving into S. To that end, we first note that since we consider a subsequent phase, we must have $(S \cup \{k\}) \prec T$, $S_1 \supseteq (T\backslash k)$ and $k \notin S_1$. Let S_{rx} denote the set of destinations receiving the transmitted packet. For the status of a $X_{k,j}$ packet to evolve into S, by Line 4 of the UPDATE, we must have $S_{\text{rx}} \not\subseteq S_1$, and

$$S = (T \cap S_1) \cup S_{\text{rx}} = (T\backslash k) \cup S_{\text{rx}}. \tag{49}$$

Since $(S \cup \{k\}) \prec T$ and $k \notin S$, we have $|S|+1 \leq |T|$, which implies $|S| \leq |T\backslash k|$. Together with (49), we have $(T\backslash k) \supseteq S_{\text{rx}}$. This contradicts the inequalities $S_1 \supseteq (T\backslash k)$ and $S_{\text{rx}} \not\subseteq S_1$. Thus no $X_{k,j}$ packet in any Stage S_1 of a subsequent Phase T can have its status $S(X_{k,j})$ evolving into S.

The above discussion proves that once we use up all $Q_{k;S}$ packets in Stage S of Phase $(S \cup \{k\})$, there will be no $X_{k,j}$ with $S(X_{k,j}) = S$ until the end of the PE scheme. As a result, once the PE scheme terminates, there will be no $X_{k,j}$ with $S(X_{k,j}) \in 2^{([K]\backslash k)}$. Therefore, all $X_{k,j}$ must have $S(X_{k,j}) \ni k$. By Lemma 3, the decodability of the proposed PE scheme is guaranteed.

We now discuss how to relax the assumption that (10) and (11) are satisfied with equality. For example, suppose (10) is a strict inequality. In this case, we simply add $n(w_{k;\emptyset\to\emptyset} \cdot p_{\cup[K]} - R_k)$ dummy all-zero packets to the (s, d_k) session so that the new rate $\tilde{R}_k$, including both the information and the dummy packets, satisfies (10) with equality. Then our previous analysis proves that our PE scheme[7] can send all $n\tilde{R}_k$ packets to d_k. The destination d_k simply discards the dummy packets when decoding.

Similarly, suppose for some $S \neq \emptyset$, (11) is a strict inequality with the difference between the left-hand side and the right-hand-side being Δ. In this case, we simply add $n \cdot \Delta$ dummy all-zero packets $X_{k,\text{dummy}}$ to the (s, d_k) session in the beginning of Stage S of Phase $(S \cup \{k\})$, and set their status $S(X_{k,\text{dummy}})$ to S. Since the dummy packets are all-zero, we can set their overhearing status to any set without affecting the decodability of other packets. In this way, we again convert a strict inequality to equality and our previous analysis follows.

With the use of dummy packets, we can convert all (10) and (11) to equalities. The proof of Proposition 3 is thus complete.

D. Attaining The Capacity Of Two Classes of PECs

In this section, we prove the capacity results for symmetric 1-to-K broadcast PECs in Proposition 4 and for spatially independent broadcast PECs with one-sided fairness constraints in Proposition 5.

[7] The dummy packets are also associated with the status $S(\cdot)$ and the coding vector $\mathbf{v}(\cdot)$. And they follow all operations of the PE scheme.

Proof of Proposition 4: Since the broadcast channel is symmetric, for any $S_1, S_2 \in 2^{[K]}$, we have

$$p_{\cup S_1} = p_{\cup S_2} \text{ if } |S_1| = |S_2|.$$

Without loss of generality, also assume that $R_1 \geq R_2 \geq \cdots \geq R_K$. By the above simplification, the outer bound in Proposition 1 collapses to the following single linear inequality:

$$\sum_{k=1}^{K} \frac{R_k}{p_{\cup[k]}} \leq 1. \tag{50}$$

We use the results in Proposition 3 to prove that (50) is indeed achievable. To that end, we first fix an arbitrary cardinality-compatible total ordering. Then for any $S \subseteq ([K]\backslash k)$, we choose

$$w_{k;S\to S} = R_k \cdot \sum_{i=K-|S|}^{K} \left(\sum_{\substack{\forall S_1 : |S_1| = i \\ ([K]\backslash S) \subseteq S_1 \subseteq [K]}} \frac{(-1)^{i-(K-|S|)}}{p_{\cup S_1}} \right), \tag{51}$$

$$w_{k;S\to T} = 0, \text{ for all } T \subseteq S \text{ and } T \neq S. \tag{52}$$

We also choose

$$\forall T \neq \emptyset,\ x_T = \max_{\forall k \in T} \left(\sum_{\forall S:(T\backslash k) \subseteq S \subseteq ([K]\backslash k)} w_{k;S\to(T\backslash k)} \right), \tag{53}$$

$$x_\emptyset = 0. \tag{54}$$

The symmetry of the broadcast PEC, the assumption that $R_1 \geq R_2 \geq \cdots \geq R_K$, and (53) jointly imply that

$$x_T = w_{k^*;(T\backslash k^*)\to(T\backslash k^*)} \text{ where } k^* \triangleq \min\{i : i \in T\} \tag{55}$$

for all $T \neq \emptyset$.

By simple probability arguments as first described[8] in Section V-B, we can show that the above choices of $w_{k;S\to T}$ and x_T are all non-negative and jointly satisfy the inequalities (9) to (12).

The remaining task is to show that inequality (8) is satisfied for any $(R_1, \cdots, R_K)$ in the interior of the capacity outer bound (50). To that end, we simply need to verify the following equalities by some simple arithmetic computation.

$$\begin{aligned} \forall k \in [K], \quad & \sum_{\forall T \in 2^{[K]}: k \in T, [k-1] \cap T = \emptyset} x_T \\ &= \sum_{\forall T \in 2^{[K]}: k \in T, [k-1] \cap T = \emptyset} w_{k;(T\backslash k)\to(T\backslash k)} \\ &= \frac{R_k}{p_{\cup[k]}}. \end{aligned} \tag{56}$$

Summing (56) over different k values, we thus show that any $(R_1, \cdots, R_K)$ in the interior of the capacity outer bound (50) indeed satisfies (8). The proof of Proposition 4 is complete. ∎

[8] Some detailed discussion can also be found in the proof of Lemma 5 in Appendix D.

Proof of Proposition 5: Consider an arbitrary spatially independent broadcast PEC with $0 < p_1 \leq p_2 \leq \cdots \leq p_K$. The capacity outer bound in Proposition 1 implies that any achievable rate vector $(R_1, \cdots, R_K)$ must satisfy

$$\sum_{k=1}^{K} \frac{R_k}{1 - \prod_{l=1}^{k}(1-p_l)} \leq 1. \tag{57}$$

We use the results in Proposition 3 to prove that any one-sidedly fair rate vector $(R_1, \cdots, R_K) \in \Lambda_{\text{osf}}$ that is in the interior of (57) is indeed achievable. To that end, we first fix an arbitrary cardinality-compatible total ordering. Then we choose $w_{k;S\to T}$ as in (51) and (52) and choose x_T as in (53) and (54). By Lemma 5 in Appendix D and by (53), we have

$$\begin{aligned} x_T &= \max_{\forall k\in T} \left(w_{k;(T\backslash k)\to(T\backslash k)}\right) \\ &= w_{k^*;(T\backslash k^*)\to(T\backslash k^*)} \text{ where } k^* \triangleq \min\{i : i \in T\} \end{aligned}$$

for all $T \neq \emptyset$.

The remaining proof can be completed by following the same steps after (55) in the proof of Proposition 4. ■

VI. Further Discussion of The Main Results

A. Accounting Overhead

Thus far we assume that the individual destination d_k knows the global coding vector $\mathbf{v}_{\text{tx}}$ that is used to generate the coded symbols (see Line 10 of the main PE scheme). Since $\mathbf{v}_{\text{tx}}$ is generated randomly, this assumption generally does not hold, and the coding vector $\mathbf{v}_{\text{tx}}$ also needs to be conveyed to the destinations. Otherwise, destinations d_k cannot decode the original information symbols $X_{k,j}$ from the received coded symbols $Z_k(t)$, $t \in [n]$. The cost of sending the coding vector $\mathbf{v}_{\text{tx}}$ is termed the *coding overhead* or the *accounting overhead.*

We use the generation-based scheme in [3] to absorb the accounting overheard. Namely, we first choose sufficiently large n and finite field size q such that the PE scheme can achieve $(1-\epsilon)$-portion of the capacity with close-to-one probability when assuming there is no accounting overhead. Once n and q are fixed, we choose an even larger finite field $\mathsf{GF}(q^{M+\sum_{k=1}^{K} nR_k})$ for some large integer M. The large finite field is then treated as a vector of dimension $M+\sum_{k=1}^{K} nR_k$. Although each information symbol (vector) is chosen from $X_{k,j} \in \mathsf{GF}(q^{M+\sum_{k=1}^{K} nR_k})$, we limit the range of the $X_{k,j}$ vector value such that the first $\sum_{k=1}^{K} nR_k$ coordinates are always zero. We can thus view the entire systems as sending M coordinates in each vector. In the PE scheme, we then focus on coding over each coordinate, respectively. The same coding vector $\mathbf{v}_{\text{tx}}$ is used repeatedly to encode the last M coordinates. And we use the first $\sum_{k=1}^{K} nR_k$ coordinates to store the coding vector $\mathbf{v}_{\text{tx}}$.

Since only the last M coordinates are used to carry information, overall the transmission rate is reduced by a factor $\frac{M}{M+\sum_{k=1}^{K} nR_k}$. By choosing a sufficiently large M, we have averaged out and absorbed the accounting overhead.

B. Minimum Finite Field Size

The PE scheme in Section IV is presented in a form of random linear network coding, which uses a sufficiently large finite field size $\mathsf{GF}(q)$ and one can prove that the desired properties hold with close-to-one probability. In contrast, the following proposition focuses on the existence of a non-random algorithm by quantifying the corresponding minimum size of the finite field.

Proposition 6: Consider the 1-to-K broadcast PEC problem with COF. For any fixed finite field $\mathsf{GF}(q_0)$ satisfying $q_0 > K$, all the achievability results in Propositions 2, 3, 4, and 5 can be attained by a deterministic PE algorithm on $\mathsf{GF}(q_0)$ that deterministically computes the mixing coefficients $\{c_k : \forall k \in T_{\text{sel}}\}$ in Line 7 of the PE scheme.

The proof of Proposition 6 is relegated to Appendix E.

Remark: In practice, the most commonly used finite field is $\mathsf{GF}(2^8)$. Proposition 6 guarantees that $\mathsf{GF}(2^8)$ is sufficient for coding over $K \leq 255$ sessions together.

C. The Accounting Overhead For Real-World Applications

Although the accounting overheard is fully absorbed under the theoretic capacity definition (see Definitions 1 and 2), in practice, such an absorption-based technique relies on the assumption that one has the freedom to increase the packet size $\mathsf{GF}(q)$ without affecting the erasure probabilities. In this subsection, we briefly discuss the performance of the PE scheme for a realistic setting in which the packet size is fixed to L bits and there are K destinations.

Proposition 6 ensures that we can use $\mathsf{GF}(q)$ with $q = K+1$. In this way, each $\mathsf{GF}(q)$ symbol can be represented by $\log_2(K+1) \approx \log_2(K)$ bits. Therefore, the overhead of transmitting the coding vector is upper bounded by $Kn\log_2(K)$ bits. The achievable rate is thus strictly away from the capacity but is no smaller than $\frac{L-Kn\log_2(K)}{L}$ fraction of the outer bound. Note that for the PE scheme, the coding vector $\mathbf{v}_{\text{tx}}$ in the header is usually sparse and highly compressible. For example, in the very beginning of the PE transmission, each $\mathbf{v}_{\text{tx}}$ is simply an elementary vector with all but one of the coordinates being zero. A heuristic estimate of the overhead of transmitting a *compressed* coding vector is $K(\log_2(n)+\log_2(K))$, where K represents the number of sessions, $\log_2(n)$ bits are used to describe the indices of the coordinates that are non-zero, and $\log_2(K)$ bits are used to describe the values of the non-zero coordinates. Using compressed coding vectors, the overhead can be drastically reduced to $\approx \frac{K(\log_2(n)+\log_2(K))}{L}$. It is of practical interest to explicitly quantify the compressibility of coding vector $\mathbf{v}_{\text{tx}}$ in the proposed PE scheme, which is beyond the scope of this work.

D. The Asymptotic Sum-Rate Capacity of Large K Values

We first define the sum-rate capacity as follows:

Definition 10: The sum-rate capacity R^*_{sum} is defined as

$$R^*_{\text{sum}} = \sup\left\{\sum_{k=1}^{K} R_k : (R_1, \cdots, R_K) \text{ is achievable}\right\}.$$

Proposition 5 quickly implies the following corollary.

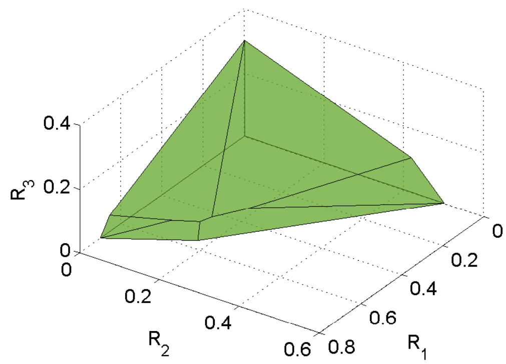


Fig. 6. The 3-D capacity region of a 1-to-3 spatially independent broadcast PEC with marginal success probabilities $p_1 = 0.7$, $p_2 = 0.5$, and $p_3 = 0.3$.

Corollary 2: Consider any spatially independent 1-to-K broadcast PECs with marginal success probabilities $0 < p_1 \leq p_2 \leq \cdots \leq p_K < 1$. With COF, the sum-rate capacity satisfies

$$\frac{\sum_{k=1}^{K} \frac{1}{1-p_k}}{\sum_{k=1}^{K} \frac{1}{(1-p_k)(1-\prod_{l=1}^{k}(1-p_l))}} \leq R^*_{\text{sum}} \leq 1.$$

If we further enforce perfect fairness, i.e., $R_1 = R_2 = \cdots = R_K$, then the corresponding sum-rate capacity $R^*_{\text{sum,perf.fair}}$ becomes

$$R^*_{\text{sum,perf.fair}} = \frac{K}{\sum_{k=1}^{K} \frac{1}{(1-\prod_{l=1}^{k}(1-p_l))}}.$$

Proof: Since the sum-rate capacity nR^*_{sum} is no larger than the total available time slots n, we have the upper bound $R^*_{\text{sum}} \leq 1$. Since the rate vector $\left(\frac{R}{1-p_1}, \frac{R}{1-p_2}, \cdots, \frac{R}{1-p_K}\right)$ is one-sidedly fair, Proposition 5 leads to the lower bound of R^*_{sum}. Since a perfectly fair rate vector $(R, R, \cdots, R)$ is also one-sidedly fair, Proposition 5 gives the exact value of $R^*_{\text{sum,perf.fair}}$. ■

Corollary 2 implies the following: Consider any fixed $p > 0$. Consider a symmetric, spatially independent 1-to-K broadcast PEC with marginal success probability $p_1 = p_2 = \cdots = p_K = p$. When K is sufficiently large, both the sum-rate capacities R^*_{sum} and $R^*_{\text{sum,perf.fair}}$ approach one. That is, for sufficiently large K, network coding completely removes all the channel uncertainty by taking advantage of the spatial diversity among different destinations d_i. Therefore, each (s, d_k) session can sustain rate $\frac{1-\epsilon}{K}$ for some $\epsilon > 0$ where $\epsilon \to 0$ when $K \to \infty$. Note that when compared to the MIMO capacity gain, the setting in this paper is more conservative in a sense that it assumes no coordination among destinations.

This relationship was first observed and proven in [15] by identifying a lower bound of $R^*_{\text{sum,perf.fair}}$ for symmetric, spatially independent PECs. Compared to the results in [15], Corollary 2 focuses on the more general setting of non-symmetric spatially independent PECs, and provides the exact value of $R^*_{\text{sum,perf.fair}}$ and a lower bound on R^*_{sum}.

E. Numerical Evaluation

Fig. 6 illustrates the 3-dimensional capacity region of (R_1, R_2, R_3) of a spatially independent, 1-to-3 broadcast PEC with COF. The corresponding marginal probabilities are $p_1 = 0.7$, $p_2 = 0.5$, and $p_3 = 0.3$. The six facets in Fig. 6 correspond to the six different permutations used in Proposition 1.

For general 1-to-K PECs with $K \geq 4$, we can use the outer and inner bounds in Propositions 1 and 3 to bracket the actual capacity region. Since there is no tightness guarantee for $K \geq 4$ except for the two special classes of channels in Section III-B, we use computer to numerically evaluate the tightness of the outer and inner bound pairs. To that end, for any fixed K value, we consider spatially independent 1-to-K broadcast PEC with the marginal success probabilities p_k chosen randomly from $(0, 1)$. To capture the K-dimensional capacity region, we first choose a search direction $\vec{v} = (v_1, \cdots, v_K)$ uniformly randomly from a K-dimensional unit ball. With the chosen values of p_k and $\vec{v}$, we use a linear programming (LP) solver to find the largest t_{outer} such that $(R_1, \cdots, R_K) = (v_1 \cdot t_{\text{outer}}, \cdots, v_K \cdot t_{\text{outer}})$ satisfies the capacity outer bound in Proposition 1.

To evaluate the capacity inner bound, we need to choose a cardinality-compatible total ordering. For any set $S \subseteq [K]$, the corresponding incidence vector $\mathbf{1}_S$ is a K-dimensional binary vector with the i-th coordinate being one if and only if $i \in S$. We can also view $\mathbf{1}_S$ as a binary number, where the first coordinate is the most significant bit and the K-th coordinate is the least significant bit. For example, for $K = 4$, $S = \{1, 2, 4\}$ has $\mathbf{1}_S = (1, 1, 0, 1) = 13$. For two sets $S_1 \neq S_2$, we define $S_1 \prec S_2$ if and only if either (i) $|S_1| = |S_2|$ and $\mathbf{1}_{S_1} < \mathbf{1}_{S_2}$, or (ii) $|S_1| < |S_2|$. Based on this cardinality-compatible total ordering, we again use the LP solver to find the largest t_{inner} such that $(R_1, \cdots, R_K) = (v_1 \cdot t_{\text{inner}}, \cdots, v_K \cdot t_{\text{inner}})$ satisfies the capacity inner bound in Proposition 3. The deficiency is then defined as $\text{defi} \triangleq \frac{t_{\text{outer}} - t_{\text{inner}}}{t_{\text{outer}}}$. We then repeat the above experiment for 10^4 times for $K = 4$, 5, and 6, respectively.

Note that although there is no tightness guarantee for $K \geq 4$ except in the one-sidedly fair rate region, all our numerical experiments (totally 3×10^4) have defi $\leq 0.1\%$. Actually, in our experiments with $K \leq 6$, we have not found any instance of the input parameters $(p_1, \cdots, p_K)$ and $\vec{v}$, for which defi is greater than the numerical precision of the LP solver. This shows that Propositions 1 and 3 indeed describe the capacity region from the practical perspective.

To illustrate the broadcast network coding gain, we compare the sum-rate capacity versus the sum rate achievable by time sharing. Figs. 7 and 8 consider symmetric, spatially independent PECs with marginal success probabilities $p_1 = \cdots = p_K = p$. We plot the sum rate capacity $R^*_{\text{sum,perf.fair}}$ versus p for a perfectly fair system. The baseline is the largest sum rate that can be achieved by time sharing for a perfectly fair system. As seen in Figs. 7 and 8, the network coding gains are substantial even when we only have $K = 4$ destinations. We also note that $R^*_{\text{sum,perf.fair}}$ approaches one for all $p \in (0, 1]$ as predicted by Corollary 2.

We are also interested in the sum rate capacity under asymmetric channel profiles (also known as heterogeneous channel profiles). Consider asymmetric, spatially independent PECs. For each p value, we let the channel gains p_1 to p_K be equally

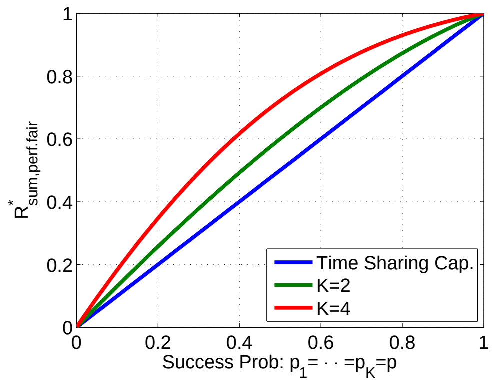


Fig. 7. The sum-rate capacity $R^*_{\text{sum,perf.fair}}$ in a perfectly fair system versus the marginal success probability p of a symmetric, spatially independent 1-to-K broadcast PEC, $K = 2$ and 4.

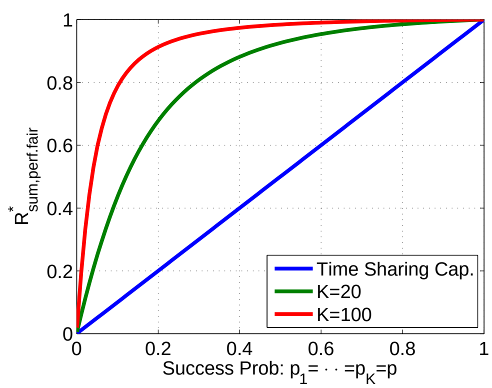


Fig. 8. The sum-rate capacity $R^*_{\text{sum,perf.fair}}$ in a perfectly fair system versus the marginal success probability p of a symmetric, spatially independent 1-to-K broadcast PEC, $K = 20$ and 100.

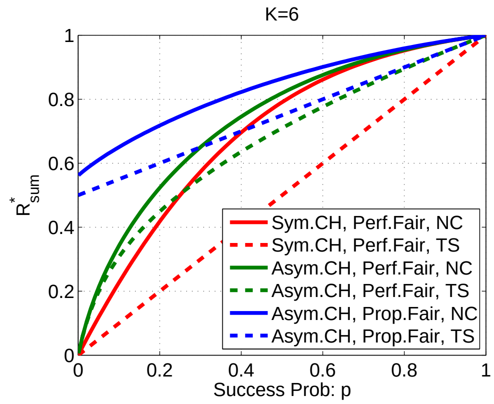


Fig. 9. The sum-rate capacities for a 6-destination heterogenous channel profiles with the success probabilities p_1 to p_6 evenly spaced between $(p, 1)$.

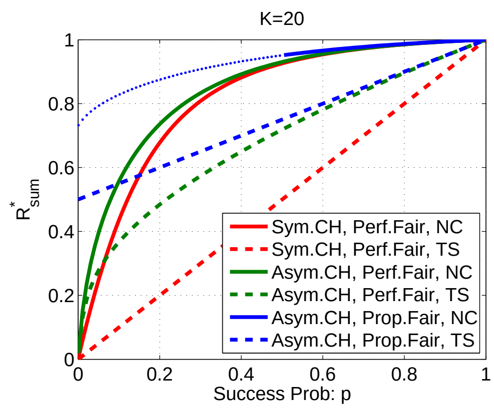


Fig. 10. The sum-rate capacities for a 20-destination heterogenous channel profiles with the success probabilities p_1 to p_{20} evenly spaced between $(p, 1)$.

spaced between $(p, 1)$, i.e., $p_k = p+(k-1)\frac{1-p}{K-1}$. We then plot the sum rate capacities for different p values. Fig. 9 describes the case for $K = 6$. The sum rate capacities are depicted by solid curves, which is obtained by solving the linear inequalities in the outer and inner bounds of Propositions 1 and 3. For all the parameter values used to plot Fig. 9, the outer and inner bounds meet and we thus have the exact sum rate capacities for the case of $K = 6$. The best achievable rate of time sharing are depicted by dashed curves in Fig. 9. We consider both a perfectly fair system $(R, R, \cdots, R)$ or a proportionally fair system $(p_1R, p_2R, \cdots, p_KR)$ for which the rate of the (s, d_k) session is proportional to the marginal success probability p_k (the optimal rate when all other sessions are silent). To highlight the impact of channel heterogeneity, we also redraw the curves of perfectly symmetric PECs with $p_1 = \cdots = p_K = p$.

As seen in Fig. 9, for perfectly fair systems, the sum-rate capacity gain does not increase much when moving from symmetric PECs $p_1 = \cdots = p_K = p$ to the heterogeneous channel profile with p_1 to p_K evenly spaced between $(p, 1)$. The reason is due to that the worst user d_1 (with the smallest p_1) dominates the system performance in a perfectly fair system. When we allow proportional fairness, network coding again provides substantial improvement for all p values. However, the gain is not as large as the case of symmetric channels. For example, when p_1 to p_K are evenly spaced between $(0, 1)$. The sum rate capacity of a proportionally fair system is 0.56 ($p = 0$). However, if all p_1 to p_K are concentrated on their mean 0.5, then the sum rate capacity of the symmetric channel ($p = 0.5$) is 0.79. The results show that for practical implementation, it is better to group together all the sessions of similar marginal success rates and perform intersession network coding within the same group.

We also repeat the same experiment of Fig. 9 but for the case $K = 20$ in Fig. 10. In this case of a moderate-sized $K = 20$, the sum-rate capacity of a perfectly fair system is characterized by Proposition 5. On the other hand, the sum-rate capacity of a proportionally fair system are characterized by Proposition 5 only when all p_1 to p_K are in the range of $[0.5, 1]$ (see the discussion of one-sidedly fair systems in Section V-D). Since the evaluations of both the outer and inner bounds have prohibitively high complexity for the case $K = 20$, we use the capacity formula of Proposition 5 as a substitute[9] of the sum-rate capacity for $p < 0.5$, which is illustrated in Fig. 10 by the fine dotted extension of the solid curve for the region

[9]When all p_1 to p_K are in $[0.5, 1]$, the formula in Proposition 5 describes the capacity. When some p_1 to p_K is outside $[0.5, 1]$, the formula in Proposition 5 describes an outer bound of the capacity.

of $p \in [0.5, 1]$. Again, the more sessions ($K = 20$) to be encoded together, the higher the network coding gain over the best time sharing rate.

VII. Conclusion

The recent development of practical network coding schemes [3] has brought attentions back to the study of packet erasure channels (PECs), which is a generalization of the classic binary erasure channels. Since per-packet feedback (such as ARQ) is widely used in today's network protocols, it is thus of critical importance to study PECs with channel output feedback (COF). This work has focused on deriving the capacity of general 1-to-K broadcast PECs with COF, which was previously known only for the case $K = 2$.

In this work, we have proposed a new class of intersession network coding schemes, termed the packet evolution (PE) schemes, for the broadcast PECs. Based on the PE schemes, we have derived the capacity region for general 1-to-3 broadcast PECs, and a pair of capacity outer and inner bounds for general 1-to-K broadcast PECs, both of which can be easily evaluated by any linear programming solver for the cases $K \leq 6$. It has also been proven that the outer and inner bounds meet for two classes of 1-to-K broadcast PECs: the symmetric broadcast PECs, and the spatially independent broadcast PECs with the one-sided fairness rate constraints. Extensive numerical experiments have shown that the outer and inner bounds meet for almost all broadcast PECs encountered in practical scenarios. Therefore, we can effectively use the outer/inner bounds as the substitute for the capacity region in practical applications. The capacity results in this paper also show that for large K values, the noise of the broadcast PECs can be effectively removed by exploiting the inherent spatial diversity of the system, even without any coordination among the destinations.

Acknowledgment

This work was supported in parts by NSF grants CCF-0845968 and CNS-0905331. The author would also like to thank for the insightful suggestions of Profs. Anant Sahai and David Tse.

Appendix A
A Proof Of Lemma 2

Proof: We prove Lemma 2 by induction on the time index t. First consider the end of the 0-th time slot (before any transmission). Since $S_0(X_{k,j}) = \emptyset$ for all $X_{k,j}$ and the only d_i satisfying $i \in (S_0(X_{k,j}) \cup \{k\})$ is d_k, we only need to check whether $\mathbf{v}_0(X_{k,j})$ is in the linear space $\mathsf{span}(\Omega_{Z,k}(0), \Omega_{M,k})$. Since in the end of time 0 the coding vector $\mathbf{v}_0(X_{k,j})$ is the elementary vector $\delta_{k,j} \in \Omega_{M,k}$, Lemma 2 holds in the end of time 0.

Suppose Lemma 2 is satisfied in the end of time $(t-1)$. Consider the end of time t. We use T and T_{sel} to denote the subsets chosen in the beginning of time t and use $\{X_{k,j_k} : \forall k \in T_{\text{sel}}\}$ to denote the corresponding target packets. Consider the following cases:

Case 1: Consider those X_{k,j_k} such that $S_t(X_{k,j_k}) = S_{t-1}(X_{k,j_k})$. We first note that if Line 4 of the Update is executed, then $S_t(X_{k,j_k}) \neq S_{t-1}(X_{k,j_k})$. Therefore, for those X_{k,j_k} such that $S_t(X_{k,j_k}) = S_{t-1}(X_{k,j_k})$, we must have that Lines 4 and 5 of the Update are not executed, which implies that $\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{t-1}(X_{k,j_k})$.

By definition, $\Omega_{Z,i}(t-1) \subseteq \Omega_{Z,i}(t)$ for all $i \in [K]$ and $t \in [n]$. By the induction assumption, we thus have that for all d_i with $i \in (S_t(X_{k,j_k}) \cup \{k\}) = (S_{t-1}(X_{k,j_k}) \cup \{k\})$,

$$\begin{aligned}&\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{t-1}(X_{k,j_k})\\ &\in \mathsf{span}(\Omega_{Z,i}(t-1), \Omega_{M,i}) \subseteq \mathsf{span}(\Omega_{Z,i}(t), \Omega_{M,i}).\end{aligned}$$

Vector $\mathbf{v}_t(X_{k,j_k})$ is thus non-interfering from the perspectives of all d_i, $i \in (S_t(X_{k,j_k}) \cup \{k\})$.

Case 2: Consider those $X_{k',j'}$ that are not selected as a target packet (i.e., $k' \notin T_{\text{sel}}$). Since those packets do not participate in time t, their $S(X_{k',j'})$ and $\mathbf{v}(X_{k',j'})$ do not change from time $(t-1)$ to time t. The same arguments of Case 1 thus hold verbatim for this case.

Case 3: Consider those target packets X_{k,j_k} such that $S_t(X_{k,j_k}) \neq S_{t-1}(X_{k,j_k})$. We must have $S_t(X_{k,j_k}) = (T \cap S_{t-1}(X_{k,j_k})) \cup S_{\text{rx}}$ and $\mathbf{v}_t(X_k, j_k) = \mathbf{v}_{\text{tx}}$ by Lines 4 and 5 of the Update, respectively. Consider any d_i such that $i \in (S_t(X_{k,j_k}) \cup \{k\})$. We have two subcases: **Case 3.1:** $i \in S_{\text{rx}}$. Since all such d_i must explicitly receive the transmitted packet corresponding to $\mathbf{v}_{\text{tx}} = \mathbf{v}_t(X_{k,j_k})$ in the end of time t, we must have

$$\begin{aligned}\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{\text{tx}} \in \mathsf{span}(\mathbf{v}_{\text{tx}}) &= \mathsf{span}(\mathbf{w}_i(t))\\ &\subseteq \Omega_{Z,i}(t) \subseteq \mathsf{span}(\Omega_{Z,i}(t), \Omega_{M,i}),\end{aligned}$$

where $\mathbf{w}_i(t)$ is the coding vector corresponding to $Z_i(t)$. Such $\mathbf{v}_t(X_{k,j_k})$ is thus non-interfering from d_i's perspective. **Case 3.2:** $i \in (S_t(X_{k,j_k}) \cup \{k\}) \backslash S_{\text{rx}}$. We first notice that

$$\begin{aligned}S_t(X_{k,j_k}) \cup \{k\} &= (T \cap S_{t-1}(X_{k,j_k})) \cup S_{\text{rx}} \cup \{k\}\\ &= ((T \cup \{k\}) \cap (S_{t-1}(X_{k,j_k}) \cup \{k\})) \cup S_{\text{rx}}\\ &= (T \cap (S_{t-1}(X_{k,j_k}) \cup \{k\})) \cup S_{\text{rx}} \quad (58)\\ &= T \cup S_{\text{rx}}, \quad (59)\end{aligned}$$

where (58) follows from that $k \in T_{\text{sel}} \subseteq T$ since X_{k,j_k} is a target packet. (59) follows from that $(S_{t-1}(X_{k,j_k}) \cup \{k\}) \supseteq T$ by Line 6 of the main structure of the PE scheme. From (59), the i value in this case must satisfy

$$i \in (S_t(X_{k,j_k}) \cup \{k\}) \backslash S_{\text{rx}} = (T \cup S_{\text{rx}}) \backslash S_{\text{rx}} = T \backslash S_{\text{rx}}. \quad (60)$$

Also by Line 6 of the main structure of the PE scheme, for all i that satisfy (60) we must have $i \in (T \backslash S_{\text{rx}}) \subseteq T \subseteq (S_{t-1}(X_{l,j_l}) \cup \{l\})$ for all $l \in T_{\text{sel}}$. By induction, the $\mathbf{v}_{t-1}(X_{l,j_l})$ vectors used to generate the new $\mathbf{v}_{\text{tx}}$ (totally $|T_{\text{sel}}|$ of them) must all be non-interfering from d_i's perspective. Therefore

$$\begin{aligned}\forall l \in T,\ \mathbf{v}_{t-1}(X_{l,j_l}) &\in \mathsf{span}(\Omega_{Z,i}(t-1), \Omega_{M,i}) \quad (61)\\ &= \mathsf{span}(\Omega_{Z,i}(t), \Omega_{M,i}),\end{aligned}$$

where the last equality follows from the fact that d_i, $i \in T \backslash S_{\text{rx}}$, does not receive any packet in time t. Since $\mathbf{v}_{\text{tx}}$ is a linear combination of $\mathbf{v}_{t-1}(X_{l,j_l})$ for all $l \in T_{\text{sel}}$, we thus have

$$\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{\text{tx}} \in \mathsf{span}(\Omega_{Z,i}(t), \Omega_{M,i}).$$

Based on the above reasoning, $\mathbf{v}_t(X_{k,j_k})$ is non-interfering for all d_i with $i \in (S_t(X_{k,j_k}) \cup \{k\}) \backslash S_{\text{rx}}$.

The proof is completed by induction on the time index t. ■

APPENDIX B
A PROOF OF LEMMA 3

Proof of Lemma 3: We focus on proving a statement that is slightly stronger than that of Lemma 3: For any n and $\epsilon > 0$, there exists a sufficiently large finite field $\mathsf{GF}(q)$ such that for all $t \in [n]$,

$$\begin{aligned}\mathsf{Prob}(\forall k \in [K],\ & \mathsf{span}(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \\ & = \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})) > 1 - \epsilon. \end{aligned} \quad (62)$$

We prove the above statement by induction on time t. In the end of time $t = 0$, since for all $k \in [K]$

$$\begin{aligned}\Omega_{R,k}(0) &= \mathsf{span}(\mathbf{v}_0(X_{k,j}) : \forall j \in [nR_k], k \notin S_0(X_{k,j}) = \emptyset) \\ &= \mathsf{span}(\delta_{k,j} : \forall j \in [nR_k]) = \Omega_{M,k},\end{aligned}$$

we have

$$\begin{aligned}\mathsf{Prob}(\forall k \in [K],\ & \mathsf{span}(\Omega_{Z,k}(0), \Omega_{R,k}(0)) \\ & = \mathsf{span}(\Omega_{Z,k}(0), \Omega_{M,k})) = 1.\end{aligned}$$

(62) is thus satisfied.

Consider the end of time $t > 0$. By induction, the following event is of close-to-one probability:

$$\begin{aligned}\forall k \in [K],\ & \mathsf{span}(\Omega_{Z,k}(t-1), \Omega_{R,k}(t-1)) \\ & = \mathsf{span}(\Omega_{Z,k}(t-1), \Omega_{M,k}).\end{aligned} \quad (63)$$

The following proofs are conditioned on the event that (63) is satisfied.

We use T and T_{sel} to denote the subsets chosen in the beginning of time t and use $\{X_{k,j_k}\}$ to denote the corresponding target packets. Consider the following cases:

Case 1: Consider those $k \in T_{\text{sel}}$ such that the corresponding target packet X_{k,j_k} either has $S_t(X_{k,j_k}) = S_{t-1}(X_{k,j_k})$ or has $k \in S_{t-1}(X_{k,j_k})$. For the former subcase $S_t(X_{k,j_k}) = S_{t-1}(X_{k,j_k})$, by Line 4 of the UPDATE, we must have $\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{t-1}(X_{k,j_k})$. Since X_{k,j_k} is the only packet among $\{X_{k,j} : \forall j \in [nR_k]\}$ that participate in time t, for which the corresponding $\mathbf{v}(X_{k,j})$ coding vector may change, we must have $\mathbf{v}_t(X_{k,j}) = \mathbf{v}_{t-1}(X_{k,j})$ for all $j \in [nR_k]$. We then have

$$\begin{aligned}\Omega_{R,k}(t) &= \mathsf{span}(\mathbf{v}_t(X_{k,j}) : \forall j \in [nR_k], k \notin S_t(X_{k,j}) \\ &= \mathsf{span}(\mathbf{v}_{t-1}(X_{k,j}) : \forall j \in [nR_k], k \notin S_{t-1}(X_{k,j})) \\ &= \Omega_{R,k}(t-1).\end{aligned} \quad (64)$$

We note that for the latter subcase $k \in S_{t-1}(X_{k,j_k})$, we must have $T \subseteq (S_{t-1}(X_{k,j_k}) \cup \{k\}) = S_{t-1}(X_{k,j_k})$ by Line 6 of the main PE scheme. Therefore Line 4 of the UPDATE implies that $k \in S_t(X_{k,j_k})$ as well. Since the remaining space $\Omega_{R,k}$ only counts the vectors $\mathbf{v}(X_{k,j})$ with $k \notin S(X_{k,j})$, (64) holds for the latter subcase as well. For both subcases, let $\mathbf{w}_k(t)$ denote the corresponding coding vector of $Z_k(t)$, which may or may not be an erasure. We then have

$$\begin{aligned}&\mathsf{span}(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \\ &= \mathsf{span}(\mathbf{w}_k(t), \Omega_{Z,k}(t-1), \Omega_{R,k}(t)) \\ &= \mathsf{span}(\mathbf{w}_k(t), \Omega_{Z,k}(t-1), \Omega_{R,k}(t-1)) \\ &= \mathsf{span}(\mathbf{w}_k(t), \Omega_{Z,k}(t-1), \Omega_{M,k}) \quad (65) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}),\end{aligned}$$

where (65) is obtained by the induction condition (63). (62) thus holds for the k values satisfying Case 1.

Case 2: Consider those d_l with $l \notin T_{\text{sel}}$. Since no $X_{l,j}$ packets participate in time t and their $S(X_{l,j})$ and $\mathbf{v}(X_{l,j})$ do not change in time t, the same arguments of Case 1 hold verbatim for this case.

Case 3: Consider those $k \in T_{\text{sel}}$ such that the corresponding target packet X_{k,j_k} has $S_t(X_{k,j_k}) \neq S_{t-1}(X_{k,j_k})$ and $k \notin S_{t-1}(X_{k,j_k})$. Define Ω'_R as

$$\Omega'_R \triangleq \mathsf{span}(\mathbf{v}_{t-1}(X_{k,j}) : \forall j \in [nR_k] \backslash j_k, k \notin S_{t-1}(X_{k,j})). \quad (66)$$

Note that the conditions of Case 3 and (66) jointly imply that $\Omega_{R,k}(t-1) = \mathsf{span}(\mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R)$. We have two subcases depending on whether $k \in S_t(X_{k,j_k})$.

Case 3.1: $k \notin S_t(X_{k,j_k})$. By Line 4 of the UPDATE, we have $k \notin S_{\text{rx}}$, i.e., d_k receives an erasure in time t. Therefore $\Omega_{Z,k}(t) = \Omega_{Z,k}(t-1)$. We will first show that $\mathsf{span}\,(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \subseteq \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})$.

Since the target d_k satisfies $k \in T_{\text{sel}} \subseteq T \subseteq (S_{t-1}(X_{l,j_l}) \cup \{l\})$, for all $l \in T_{\text{sel}}$, by Lemma 2 all those $\mathbf{v}_{t-1}(X_{l,j_l})$ are non-interfering from d_k's perspective. That is,

$$\begin{aligned}\forall l \in T_{\text{sel}},\ \mathbf{v}_{t-1}(X_{l,j_l}) &\in \mathsf{span}(\Omega_{Z,k}(t-1), \Omega_{M,k}) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}).\end{aligned} \quad (67)$$

As a result, we have $\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{\text{tx}} \in \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})$ since $\mathbf{v}_{\text{tx}}$ is a linear combination of all $\mathbf{v}_{t-1}(X_{l,j_l})$ for all $l \in T_{\text{sel}}$. Therefore, we have

$$\begin{aligned}&\mathsf{span}\,(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \\ &= \mathsf{span}\,(\Omega_{Z,k}(t), \mathbf{v}_t(X_{k,j_k}), \Omega'_R) \\ &\subseteq \mathsf{span}(\Omega_{Z,k}(t), \Omega_{Z,k}(t), \Omega_{M,k}, \Omega'_R) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}, \Omega'_R).\end{aligned} \quad (68)$$

Since we condition on the event that (63) holds, we have

$$\begin{aligned}\mathsf{span}(\Omega_{Z,k}(t), \Omega'_R) &\subseteq \mathsf{span}\,(\Omega_{Z,k}(t), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R) \\ &= \mathsf{span}\,(\Omega_{Z,k}(t), \Omega_{R,k}(t-1)) \\ &= \mathsf{span}\,(\Omega_{Z,k}(t-1), \Omega_{R,k}(t-1)) \\ &= \mathsf{span}\,(\Omega_{Z,k}(t-1), \Omega_{M,k}) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}).\end{aligned} \quad (69)$$

Jointly (68) and (69) show that $\mathsf{span}\,(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \subseteq \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})$.

To prove (62) for Case 3.1, it remains to show that the event $\mathsf{span}\,(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \supseteq \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})$ is of close-to-one probability, conditioning on (63) being true. We consider two subcases: depending on whether the following equation is satisfied.

$$\mathbf{v}_{t-1}(X_{k,j_k}) \in \mathsf{span}\left(\Omega_{Z,k}(t-1), \Omega'_R\right) \\ = \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right). \tag{70}$$

Case 3.1.1: If (70) is satisfied, then we have

$$\begin{aligned} &\mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,k}(t)\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \mathbf{v}_t(X_{k,j_k}), \Omega'_R\right) \\ &\supseteq \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R\right) \quad (71) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,k}(t-1)\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t-1), \Omega_{R,k}(t-1)\right) \\ &= \mathsf{span}(\Omega_{Z,k}(t-1), \Omega_{M,k}) \quad (72) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}), \quad (73) \end{aligned}$$

where (71) follows from (70), and (72) follows from the induction condition (63).

Case 3.1.2: (70) is not satisfied. By the equality between (71) and (73), we have

$$\mathsf{span}\left(\Omega_{Z,k}(t), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R\right) = \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}). \tag{74}$$

Recall that $\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{\text{tx}}$ is a linear combination of $\mathbf{v}_{t-1}(X_{l,j_l})$ satisfying (67). By (67), (74), and the assumption that (70) is not satisfied, we thus have that each $\mathbf{v}_{t-1}(X_{l,j_l})$ can be written as a unique linear combination: $\alpha\mathbf{v}_{t-1}(X_{k,j_k})+\mathbf{w}$ where α is a $\mathsf{GF}(q)$ coefficient and $\mathbf{w}$ is a vector satisfying $\mathbf{w} \in \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right)$. By the same reasoning, we can rewrite $\mathbf{v}_t(X_{k,j_k})$ as

$$\begin{aligned} \mathbf{v}_t(X_{k,j_k}) &= c_k\mathbf{v}_{t-1}(X_{k,j_k}) + \sum_{\forall l \in T_{\text{sel}}\backslash k} c_l \mathbf{v}_{t-1}(X_{l,j_l}) \\ &= c_k\mathbf{v}_{t-1}(X_{k,j_k}) + (\alpha\mathbf{v}_{t-1}(X_{k,j_k}) + \mathbf{w}) \\ &= (c_k+\alpha)\mathbf{v}_{t-1}(X_{k,j_k}) + \mathbf{w}. \end{aligned} \tag{75}$$

where α is a $\mathsf{GF}(q)$ coefficient, $\mathbf{w}$ is a vector satisfying $\mathbf{w} \in \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right)$, and the values of α and $\mathbf{w}$ depend on the random coefficients c_l for all $l \in T_{\text{sel}}\backslash k$. As a result, we have

$$\begin{aligned} &\mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,k}(t)\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \mathbf{v}_t(X_{k,j_k}), \Omega'_R\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \left((c_k+\alpha)\mathbf{v}_{t-1}(X_{k,j_k}) + \mathbf{w}\right), \Omega'_R\right). \end{aligned}$$

Since (70) is not satisfied and $\mathbf{w} \in \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right)$, we have

$$\begin{aligned} &\mathsf{span}\left(\Omega_{Z,k}(t), \left((c_k+\alpha)\mathbf{v}_{t-1}(X_{k,j_k}) + \mathbf{w}\right), \Omega'_R\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R\right) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}) \end{aligned} \tag{76}$$

if and only if $(c_k+\alpha) \neq 0$. Since c_k is uniformly distributed in $\mathsf{GF}(q)$ and the random variables c_k and α are independent, the event that (76) is true has conditional probability $\frac{q-1}{q}$, conditioning on (63) being true. For sufficiently large q values, the conditional probability approaches one.

Case 3.2: $k \in S_t(X_{k,j_k})$. Recall that for Case 3, we consider those k such that $k \notin S_{t-1}(X_{k,j_k})$. By Line 4 of the UPDATE, we must have $k \in S_{\text{rx}}$, i.e., d_k receives the transmitted packet perfectly in time t. Therefore, in the end of time t, $\Omega_{R,k}(t) = \Omega'_R$, which was first defined in (66).

We consider two subcases: depending on whether the following equation is satisfied.

$$\mathbf{v}_{t-1}(X_{k,j_k}) \in \mathsf{span}\left(\Omega_{Z,k}(t-1), \Omega'_R\right). \tag{77}$$

Case 3.2.1: If (77) is satisfied, then we have

$$\begin{aligned} &\mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,k}(t)\right) = \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \Omega'_R\right) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R\right) \quad (78) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \Omega_{R,k}(t-1)\right) \\ &= \mathsf{span}(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \Omega_{M,k}) \quad (79) \\ &= \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k}), \end{aligned}$$

where (78) follows from (77), and (79) follows from the induction assumption (63).

Case 3.2.2: (77) is not satisfied. By the induction assumption (63), we have

$$\begin{aligned} &\mathsf{span}\left(\Omega_{Z,k}(t-1), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R\right) \\ &= \mathsf{span}(\Omega_{Z,k}(t-1), \Omega_{M,k}). \end{aligned} \tag{80}$$

Since the target d_k satisfies $k \in T_{\text{sel}} \subseteq T \subseteq (S_{t-1}(X_{l,j_l}) \cup \{l\})$, for all $l \in T_{\text{sel}}$, by Lemma 2 all those $\mathbf{v}_{t-1}(X_{l,j_l})$ are non-interfering from d_k's perspective. That is,

$$\forall l \in T_{\text{sel}},\ \mathbf{v}_{t-1}(X_{l,j_l}) \in \mathsf{span}(\Omega_{Z,k}(t-1), \Omega_{M,k}). \tag{81}$$

By (80), (81), and the assumption that (77) is not satisfied, each $\mathbf{v}_{t-1}(X_{l,j_l})$ can thus be written as a unique linear combination: $\alpha\mathbf{v}_{t-1}(X_{k,j_k})+\mathbf{w}$ where α is a $\mathsf{GF}(q)$ coefficient and $\mathbf{w}$ is a vector satisfying $\mathbf{w} \in \mathsf{span}\left(\Omega_{Z,k}(t-1), \Omega'_R\right)$. Since $\mathbf{v}_t(X_{k,j_k}) = \mathbf{v}_{\text{tx}}$ is a linear combination of $\mathbf{v}_{t-1}(X_{l,j_l})$, by the same reasoning, we can rewrite $\mathbf{v}_t(X_{k,j_k})$ as

$$\begin{aligned} \mathbf{v}_t(X_{k,j_k}) &= c_k\mathbf{v}_{t-1}(X_{k,j_k}) + \sum_{\forall l \in T_{\text{sel}}\backslash k} c_l \mathbf{v}_{t-1}(X_{l,j_l}) \\ &= c_k\mathbf{v}_{t-1}(X_{k,j_k}) + (\alpha\mathbf{v}_{t-1}(X_{k,j_k}) + \mathbf{w}) \\ &= (c_k+\alpha)\mathbf{v}_{t-1}(X_{k,j_k}) + \mathbf{w}. \end{aligned} \tag{82}$$

where α is a $\mathsf{GF}(q)$ coefficient, $\mathbf{w}$ is a vector satisfying $\mathbf{w} \in \mathsf{span}\left(\Omega_{Z,k}(t-1), \Omega'_R\right)$, and the values of α and $\mathbf{w}$ depend on the random coefficients c_l for all $l \in T_{\text{sel}}\backslash k$. As a result, we have

$$\begin{aligned} &\mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,k}(t)\right) = \mathsf{span}\left(\Omega_{Z,k}(t), \Omega'_R\right) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \Omega'_R\right) \quad (83) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \mathbf{v}_{t-1}(X_{k,j_k}), \Omega'_R\right) \quad (84) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \Omega_{R,k}(t-1)\right) \\ &= \mathsf{span}\left(\mathbf{v}_t(X_{k,j_k}), \Omega_{Z,k}(t-1), \Omega_{M,k}\right) \\ &= \mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{M,k}\right), \end{aligned}$$

where the equality from (83) to (84) is true if and only if the $(c_k+\alpha)$ in (82) is not zero, since (77) is not satisfied and $\mathbf{w} \in \mathsf{span}\left(\Omega_{Z,k}(t-1), \Omega'_R\right)$.

Since c_k is uniformly distributed in $\mathsf{GF}(q)$ and the random variables c_k and α are independent, the event that

$\mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,k}(t)\right) = \mathsf{span}\left(\Omega_{Z,k}(t), \Omega_{R,M}\right)$ has the conditional probability $\frac{q-1}{q}$, conditioning on (63) being true. For sufficiently large q values, the conditional probability approaches one.

Combining all cases: Let T and T_{sel} denote the subsets chosen in time t and let $\mathcal{A}_{k,t}$ denote the event that $\mathsf{span}(\Omega_{Z,k}(t), \Omega_{R,k}(t)) = \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})$. Note that Cases 3.1.2 and 3.2.2 are the only two subcases such that the conditional probability of $\mathcal{A}_{k,t}$ given $\bigcap_{\forall j\in[K]} \mathcal{A}_{j,t-1}$ is lower bounded by $\frac{q-1}{q}$ and for all other subcases the conditional probability is one. Also note that both Cases 3.1.2 and 3.2.2 have $k \in T_{\text{sel}}$. As a result, the discussion of Cases 1 to 3.2 plus the union bound lead to the following inequalities:

$$\mathsf{Prob}\left(\bigcap_{\forall k\in[K]} \mathcal{A}_{k,t} \middle| T, T_{\text{sel}}, \bigcap_{\forall j\in[K]} \mathcal{A}_{j,t-1}\right) \geq \left(1 - \frac{|T_{\text{sel}}|}{q}\right).$$

Since for any $T_{\text{sel}} \subseteq T \subseteq [K]$ we must have $|T_{\text{sel}}| \leq K$, we then have

$$\mathsf{Prob}\left(\bigcap_{\forall k\in[K]} \mathcal{A}_{k,t} \middle| \bigcap_{\forall j\in[K]} \mathcal{A}_{j,t-1}\right) \geq \left(1 - \frac{K}{q}\right). \quad (85)$$

By concatenating the conditional probabilities, we thus have

$$\begin{aligned}&\mathsf{Prob}(\forall k \in [K],\ \mathsf{span}(\Omega_{Z,k}(t), \Omega_{R,k}(t)) \\ &\qquad\qquad = \mathsf{span}(\Omega_{Z,k}(t), \Omega_{M,k})) \\ &= \mathsf{Prob}\left(\bigcap_{\forall k\in[K]} \mathcal{A}_{k,t}\right) \geq \left(1 - \frac{K}{q}\right)^t \geq \left(1 - \frac{K}{q}\right)^n. \end{aligned} \quad (86)$$

As a result, for any fixed K and n values, we can choose a sufficiently large finite field $\mathsf{GF}(q)$ such that (86) approaches one. (62) thus holds for all $t \in [n]$.

■

APPENDIX C
A PROOF OF LEMMA 4

Proof of Lemma 4: Suppose Lemma 4 is not true and we have d_i dominates d_k, d_k dominates d_l, and d_l dominates d_i. By definition, we must have

$$R_i\left(\frac{1}{p_{\cup(\{1,2,3\}\backslash k)}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) \geq R_k\left(\frac{1}{p_{\cup(\{1,2,3\}\backslash i)}} - \frac{1}{p_{\cup\{1,2,3\}}}\right), \quad (87)$$

$$R_k\left(\frac{1}{p_{\cup(\{1,2,3\}\backslash l)}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) \geq R_l\left(\frac{1}{p_{\cup(\{1,2,3\}\backslash k)}} - \frac{1}{p_{\cup\{1,2,3\}}}\right), \quad (88)$$

$$R_l\left(\frac{1}{p_{\cup(\{1,2,3\}\backslash i)}} - \frac{1}{p_{\cup\{1,2,3\}}}\right) \geq R_i\left(\frac{1}{p_{\cup(\{1,2,3\}\backslash l)}} - \frac{1}{p_{\cup\{1,2,3\}}}\right). \quad (89)$$

We then notice that the product of the left-hand sides of (87), (88), and (89) equals the product of the right-hand sides of (87), (88), and (89). As a result, all three inequalities (87), (88), and (89) must be equalities. Since (89) is an equality, we can also say that d_i dominates d_l. The proof of Lemma 4 is complete. ■

APPENDIX D
A KEY LEMMA FOR THE PROOF OF PROPOSITION 5

For any $S \subseteq [K]$ and $S \neq [K]$, define

$$L_S \triangleq \sum_{i=K-|S|}^{K} \left(\sum_{\substack{\forall S_1 : |S_1| = i \\ ([K]\backslash S) \subseteq S_1 \subseteq [K]}} \frac{(-1)^{i-(K-|S|)}}{p_{\cup S_1}} \right).$$

We then have the following lemma:

Lemma 5: Suppose the 1-to-K broadcast PEC is spatially independent with marginal success probabilities $0 < p_1 \leq \cdots \leq p_K$. Consider any one-sidedly fair rate vector $(R_1, \cdots, R_K) \in \Lambda_{\text{osf}}$ and any $T \subseteq [K]$. For any $k_1, k_2 \in T$ with $k_1 < k_2$, we have

$$R_{k_1} \cdot L_{T\backslash k_1} \geq R_{k_2} \cdot L_{T\backslash k_2}.$$

Proof: Consider K independent geometric random variables X_1 to X_K with success probability p_1 to p_K. That is, the probability mass function $F_k(t)$ of any X_k satisfies

$$F_k(t) \triangleq \mathsf{Prob}(X_k = t) = p_k(1 - p_k)^{t-1},$$

for all strictly positive integer t. For the sake of simplicity, here we omit the discussion of the degenerate case in which $p_k = 1$. We say that the geometric random trial X_k is finished at time t if $X_k = t$. For any $S \subseteq [K]$ and $S \neq [K]$, define three random variables

$$Y_{[K]\backslash S} \triangleq \min\{X_i : i \in [K]\backslash S\} \quad (90)$$

$$W_S \triangleq \max\{X_i : i \in S\} \quad (91)$$

$$\Gamma_S \triangleq Y_{[K]\backslash S} - \min(Y_{[K]\backslash S}, W_S). \quad (92)$$

For (91), we use the convention $\max \emptyset = 0$.

Intermediate Step 1: We will first show that

$$L_S = \mathsf{E}\left\{\Gamma_S\right\}.$$

To that end, for any time t, we mark each time instant t by a set $I_t \triangleq \{i \in [K] : X_i < t\}$. We then have

$$\Gamma_S = Y_{[K]\backslash S} - \min(Y_{[K]\backslash S}, W_S) = \sum_{t=1}^{\infty} 1_{\{I_t = S\}}.$$

By noting that

$$t \leq Y_{[K]\backslash S} \Longleftrightarrow I_t \subseteq S,$$

we also have

$$Y_{[K]\backslash S} = \sum_{t=1}^{\infty} 1_{\{t \leq Y_{[K]\backslash S}\}} = \sum_{t=1}^{\infty} 1_{\{I_t \subseteq S\}} = \sum_{\forall S' : S' \subseteq S} \Gamma_{S'}. \quad (93)$$

Taking the expectation of (93), we then have for all $S \subseteq [K]$ and $S \neq [K]$,

$$\sum_{\forall S': S' \subseteq S} \mathsf{E}\left\{\Gamma_{S'}\right\} = \mathsf{E}\left\{Y_{[K]\backslash S}\right\} = \frac{1}{p_{\cup([K]\backslash S)}}. \quad (94)$$

Solving the simultaneous equations (94), we have

$$\begin{aligned} \mathsf{E}\left\{\Gamma_{S'}\right\} &= \sum_{i=K-|S'|}^{K} \left(\sum_{\substack{\forall S_1 \,:\, |S_1| = i \\ ([K]\backslash S') \subseteq S_1 \subseteq [K]}} \frac{(-1)^{i-(K-|S'|)}}{p_{\cup S_1}} \right) \\ &= L_{S'}, \end{aligned}$$

for all $S' \subseteq [K]$ and $S' \neq [K]$.

Intermediate Step 2: We will show that for any non-empty subset $T \subseteq [K]$ and any $k_1, k_2 \in T$ with $k_1 < k_2$, we have

$$\frac{L_{T\backslash k_1}}{1 - p_{k_1}} \geq \frac{L_{T\backslash k_2}}{1 - p_{k_2}}. \quad (95)$$

For any realization $(X_1, \cdots, X_K) = (x_1, \cdots, x_K)$, we use $y_{[K]\backslash S}$, w_S, and γ_S to denote the corresponding values of $Y_{[K]\backslash S}$, W_S, and Γ_S according to (90), (91), and (92), respectively. We then have

$$\begin{aligned} \mathsf{E}\left\{\Gamma_{T\backslash k_1}\right\} &= \sum_{\forall (x_1, \cdots, x_K)} \gamma_{T\backslash k_1} \prod_{k=1}^{K} F_k(x_k) \\ &= \sum_{\forall (x_1, \cdots, x_K): \gamma_{T\backslash k_1} > 0} \gamma_{T\backslash k_1} \prod_{k=1}^{K} F_k(x_k). \quad (96) \end{aligned}$$

Note that the only difference between $\mathsf{E}\left\{\Gamma_{T\backslash k_1}\right\}$ and $\mathsf{E}\left\{\Gamma_{T\backslash k_k}\right\}$ is the underlying measures of X_{k_1} and X_{k_2}. Therefore, by the change of measure formula, we have

$$\begin{aligned} \mathsf{E}\left\{\Gamma_{T\backslash k_2}\right\} = &\sum_{\forall (x_1, \cdots, x_K): \gamma_{T\backslash k_1} > 0} \gamma_{T\backslash k_1} \cdot \\ &\left(\frac{F_{k_2}(x_{k_1})}{F_{k_1}(x_{k_1})} \frac{F_{k_1}(x_{k_2})}{F_{k_2}(x_{k_2})} \right) \prod_{k=1}^{K} F_k(x_k). \quad (97) \end{aligned}$$

Note that when $\gamma_{T\backslash k_1} > 0$, we must have $y_{([K]\backslash T)\cup\{k_1\}} > w_{T\backslash k_1}$, which in turn implies that $x_{k_1} \geq x_{k_2} + 1$ since $k_2 \in (T\backslash k_1)$. We then have

$$\begin{aligned} &\frac{F_{k_2}(x_{k_1})}{F_{k_1}(x_{k_1})} \frac{F_{k_1}(x_{k_2})}{F_{k_2}(x_{k_2})} \\ &= \frac{p_{k_2}(1 - p_{k_2})^{x_{k_1}-1}}{p_{k_1}(1 - p_{k_1})^{x_{k_1}-1}} \frac{p_{k_1}(1 - p_{k_1})^{x_{k_2}-1}}{p_{k_2}(1 - p_{k_2})^{x_{k_2}-1}} \\ &= \left(\frac{1 - p_{k_2}}{1 - p_{k_1}} \right)^{x_{k_1} - x_{k_2}} \\ &\leq \left(\frac{1 - p_{k_2}}{1 - p_{k_1}} \right), \quad (98) \end{aligned}$$

where the last inequality follows from $p_{k_1} \leq p_{k_2}$ and $x_{k_1} \geq x_{k_2} + 1$. Combining (96), (97), and (98), we thus have

$$\mathsf{E}\{\Gamma_{T\backslash k_1}\} \left(\frac{1 - p_{k_2}}{1 - p_{k_1}} \right) \geq \mathsf{E}\{\Gamma_{T\backslash k_2}\},$$

which implies (95).

Final Step 3: Since $(R_1, \cdots, R_K) \in \Lambda_{\text{osf}}$, by the definition of one-sided fairness, we have

$$R_{k_1}(1 - p_{k_1}) \geq R_{k_2}(1 - p_{k_2}). \quad (99)$$

Multiplying (95) and (99) together, the proof of Lemma 5 is complete.

■

APPENDIX E
A PROOF OF PROPOSITION 6

Proof of Proposition 6: To prove this proposition, we only need to prove that for any $q_0 > K$, the source s can always *compute* the mixing coefficients $\{c_k : \forall k \in T\}$ in Line 7 of the PE scheme, such that the key properties in Lemmas 2 and 3 hold with probability one. Then for any PE scheme, we can use the computed mixing coefficients $\{c_k : \forall k \in T\}$ instead of the randomly chosen ones, while attaining the same desired throughput performance during our construction of capacity-achieving PE schemes in Sections V-A and V-C.

We first notice that the proof of Lemma 2 does not involve any probabilistic arguments. Therefore, Lemma 2 holds for any choices of the mixing coefficients with probability one.

For Lemma 3, we notice that for individual time t, the probability that (62), a stronger version of Lemma 3, is satisfied conditioning on the satisfaction in time $(t-1)$ is lower bounded by (85). As a result, when $q_0 > K$, the conditional probability is strictly positive. Therefore, there must exist one realization of random $\{c_k : k \in T_{\text{sel}}\}$ at time t that satisfies (62). By deterministically choosing the coefficients $\{c_k : k \in T_{\text{sel}}\}$ from one of those good realizations, we can satisfy (62) for time t with probability one. By deterministically choosing good $\{c_k : k \in T_{\text{sel}}\}$ for all $t \in [n]$, the proof is complete.

■

REFERENCES

[1] P. Bergmans, "Random coding theorem for broadcast channels with degraded components," *IEEE Trans. Inf. Theory*, vol. 19, pp. 197–207, March 1973.

[2] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, August 2008.

[3] 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.

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

[5] 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.

[6] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," in *Proc. IEEE Int'l Symp. Inform. Theory*. Austin, Texas, USA, June 2010.

[7] A. El Gamal, "The feedback capacity of degraded broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 379–381, March 1978.

[8] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Multiuser broadcast erasure channel with feedback — capacity and algorithms," in *Proc. NetCoop*, 2010.

[9] 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.

[10] 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.

[11] C. Jiang, B. Smith, B. Hassibi, and S. Vishwanath, “Multicast in wireless erasure networks with feedback,” in *IEEE Symp. Computer & Commun.*, July 2008, pp. 562–565.
[12] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “XORs in the air: Practical wireless network,” in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*, 2006.
[13] J. Körner and K. Marton, “General broadcast channels with degraded message sets,” *IEEE Trans. Inf. Theory*, vol. 23, no. 1, pp. 60–64, January 1977.
[14] D. Koutsonikolas, C.-C. Wang, and Y. Hu, “CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments,” in *Proc. 29th IEEE Conference on Computer Communications (INFOCOM)*. San Diego, USA, March 2010, pp. 1–9.
[15] P. Larsson and N. Johansson, “Multi-user ARQ,” in *Proc. IEEE Vehicular Technology Conference*, 2006.
[16] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, February 2003.
[17] X. Li, C.-C. Wang, and X. Lin, “Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints,” in *Proc. 29th IEEE Conference on Computer Communications (INFOCOM)*. San Diego, USA, March 2010, pp. 1–5.
[18] M. Maddah-Ali and D. Tse, “Completely stale transmitter channel state information is still very useful,” in *Proc. 48th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, September 2010.
[19] 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.
[20] E. Rozner, A. Iyer, Y. Mehta, L. Qiu, and M. Jafry, “ER: Efficient retranmission scheme for wireless LANs,” in *Proc. ACM CoNEXT*. New York, USA, December 2007.
[21] B. Smith and B. Hassibi, “Wireless erasure networks with feedback,” in *Proc. IEEE Int’l Symp. Inform. Theory*, July 2008, pp. 339–343.
[22] J. Sundararajan, D. Shah, and M. Médard, “ARQ for network coding,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Toronto, Canada, July 2008.
[23] C.-C. Wang, “On the capacity of wireless 1-hop intersession network coding — a broadcast packet erasure channel approach,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Austin, TX, USA, June 2010.
[24] C.-C. Wang, A. Khreishah, and N. Shroff, “Cross-layer optimizations for intersession network coding on practical 2-hop relay networks,” in *Proc. 43rd Asilomar Conference on Signals, Systems and Computers*. Pacific Grove, CA, USA, November 2009, pp. 771–775.
[25] C.-C. Wang and N. Shroff, “Pairwise intersession network coding on directed networks,” *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3879–3900, August 2010.
[26] H. Weingarten, Y. Steinberg, and S. Shamai, “The capacity region of the Gaussian multiple-input multiple-output broadcast channel,” *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936–3964, September 2006.
[27] Y. Wu, “Broadcasting when receivers know some messages a priori,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Nice, France, June 2007, pp. 1141–1145.
[28] F. Xue and X. Yang, “Network coding and packet-erasure broadcast channel,” in *Proc. the 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops (SECON)*. San Francisco, CA, USA, July 2008.

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

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