Linear Network Coding Capacity Region of The Smart Repeater with Broadcast Erasure Channels

Jaemin Han and Chih-Chun Wang
School of Electrical and Computer Engineering, Purdue University, USA
{han83,chihw}@purdue.edu

***Abstract*—This work considers the smart repeater network where a single source s wants to send two independent packet streams to destinations $\{d_1, d_2\}$ with the help of relay r. The transmission from s or r is modeled by packet erasure channels: For each time slot, a packet transmitted by s may be received, with some probabilities, by a random subset of $\{d_1, d_2, r\}$; and those transmitted by r will be received by a random subset of $\{d_1, d_2\}$. Interference is avoided by allowing at most one of $\{s, r\}$ to transmit in each time slot. One example of this model is any cellular network that supports two cell-edge users when a relay in the middle uses the same downlink resources for throughput/safety enhancement.**

In this setting, we study the capacity region of (R_1, R_2) when allowing linear network coding (LNC). The proposed LNC inner bound introduces more advanced packing-mixing operations other than the previously well-known butterfly-style XOR operation on overheard packets of two co-existing flows. A new LNC outer bound is derived by exploring the inherent algebraic structure of the LNC problem. Numerical results show that, with more than 85% of the experiments, the relative sum-rate gap between the proposed outer and inner bounds is smaller than 0.08%, thus effectively bracketing the LNC capacity of the smart repeater problem.

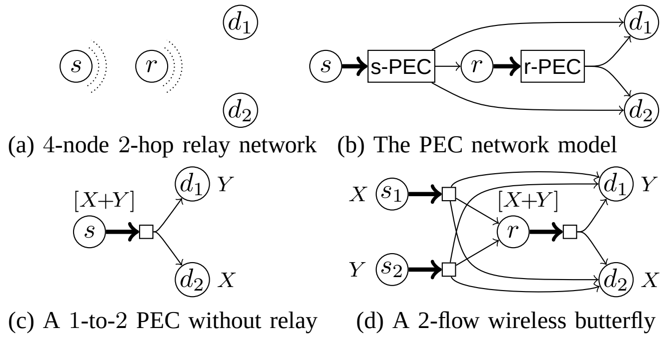


Fig. 1: The 2-flow Smart Repeater Network and its subset scenarios

I. Introduction

Increasing throughput/connectivity within scarce resources has been the main motivation for modern wireless communications. Among the various proposed techniques, the concept of *relaying* has attracted much attention as a cost-effective enabler to extend the network coverage and capacity. In recent 5G discussions, relaying became one of the core parts for the future cellular architecture including techniques of small cell managements and device-to-device communications between users [1].

In network information theory, many intelligent and cooperative relaying strategies have been devised such as decode-and-forward/compress-and-forward for relay networks [2], [3], network coding for noiseless networks [4], and general noisy network coding for discrete memoryless networks [5]. Among them, network coding has emerged as a promising technique for a practical wireless networking solution, which models the underlying wireless channels by a simple but non-trivial random packet erasure network. That is, each node is associated with its own broadcast packet erasure channel (PEC). Namely, each node can choose a symbol $X \in \mathbb{F}_q$ from some finite field $\mathbb{F}_q$, transmits X, and a random subset of receivers will receive the packet. In this setting, [6] proved that the *linear network coding* (LNC), operating only by "linear" packet-mixings, suffices to achieve the single-multicast capacity. Moreover, recent wireless testbeds have also demonstrated substantial LNC throughput gain for multiple-unicasts over the traditional store-and-forward 802.11 routing protocols [7], [8].

Motivated by these results, we are interested in finding an optimal or near-optimal LNC strategy for wireless relaying networks. To simplify the analysis, we consider a 4-node 2-hop network with one source s, two destinations $\{d_1, d_2\}$, and a common relay r inter-connected by two broadcast PECs. See Fig. 1(a-b) for details. We assume time-sharing between s and r so that interference is fully avoided, and causal packet ACKnowledgment feedback [7]–[10]. In this way, we can concentrate on how the relay r and source s can jointly exploit the broadcast channel diversity within the network.

When relay r is not present, Fig. 1(b) collapses to Fig. 1(c), the 2-receiver broadcast PEC. It was shown in [9] that a simple LNC scheme is capacity-achieving. The idea is to exploit the wireless diversity created by random packet erasures, i.e., overhearing packets of other flows. Whenever a packet X intended for d_1 is received only by d_2 and a packet Y intended for d_2 is received only by d_1, s can transmit their linear mixture $[X+Y]$ to benefit both receivers simultaneously. This simple but elegant "butterfly-style" LNC operation achieves the Shannon capacity of Fig. 1(c) [9]. Another related scenario is a 2-flow wireless butterfly network in Fig. 1(d) that contains two separate sources s_1 and s_2 instead of a single source s as in our setting. In this butterfly scenario, each source can only mix packets of their own flow. [10] showed that the same butterfly-style LNC is no longer optimal but very close to optimal. In contrast, in our setting of Fig. 1(b), the two flows are originating from the same source s. Therefore, s can perform "inter-flow NC" to further improve the performance. As we will see, relay r should not just "forward" the packets it has received and need to actively perform coding in order to approach the capacity. This is why we call such a scenario the smart repeater problem.

Contributions: This work investigates the LNC capacity region (R_1, R_2) of the smart repeater network. The outer bound is proposed by leveraging upon the algebraic structure of the underlying LNC problem. For the achievability scheme, we show that the classic butterfly-style is far from optimality and propose new LNC operations that lead to close-to-optimal performance. By numerical simulations, we demonstrate that the proposed outer/inner bounds are very close, thus effectively bracketing the LNC capacity of the smart repeater problem.

II. Problem Definition And Useful Notations

A. Problem Formation for The Smart Repeater Network

The 2-flow wireless smart repeater network with broadcast PECs, see Fig. 1(b), can be modeled as follows. Consider two traffic rates (R_1, R_2) and assume slotted transmissions. Within a total budget of n time slots, source s would like to send nR_k packets, denoted by a row vector $\mathbf{W}_k$, to destination d_k for all $k \in \{1, 2\}$ with the help of relay r. Each packet is chosen uniformly randomly from a finite field $\mathbb{F}_q$ with size $q > 0$. To that end, we denote $\mathbf{W} \triangleq (\mathbf{W}_1, \mathbf{W}_2)$ as an $n(R_1+R_2)$-dimensional row vector of all the packets, and define the linear space $\Omega \triangleq (\mathbb{F}_q)^{n(R_1+R_2)}$ as the *overall message/coding space*.

To represent the reception status, for any time slot $t \in \{1, \cdots, n\}$, we define two *channel reception status vectors*:

$$\mathbf{Z}_s(t) = (Z_{s\to d_1}(t), Z_{s\to d_2}(t), Z_{s\to r}(t)) \in \{1, *\}^3,$$
$$\mathbf{Z}_r(t) = (Z_{r\to d_1}(t), Z_{r\to d_2}(t)) \in \{1, *\}^2,$$

where "1" and "$*$" represent successful reception and erasure, respectively. For example, $Z_{s\to d_1}(t) = 1$ and $*$ represents whether d_1 can receive the transmission from source s or not at time slot t. We then use $\mathbf{Z}(t) \triangleq (\mathbf{Z}_s(t), \mathbf{Z}_r(t))$ to describe the 5-dimensional channel reception status vector of the entire network. We also assume that $\mathbf{Z}(t)$ is memoryless and stationary, i.e., $\mathbf{Z}(t)$ is independently and identically distributed over the time axis t.

We assume that either source s or relay r can transmit at each time slot, and express the *scheduling decision* by $\sigma(t) \in \{s, r\}$. For example, if $\sigma(t) = s$, then source s transmits a packet $X_s(t) \in \mathbb{F}_q$; and only when $Z_{s\to h}(t) = 1$, node h (one of $\{d_1, d_2, r\}$) will receive $Y_{s\to h}(t) = X_s(t)$. In all other cases, node h receives an erasure $Y_{s\to h}(t) = *$. The reception $Y_{r\to h}(t)$ of relay r's transmission is defined similarly.

Assuming that the 5-bit $\mathbf{Z}(t)$ vector is broadcast to both s and r after each packet transmission through a separate control channel, a *linear network code* contains n scheduling functions

$$\forall\, t \in \{1, \cdots, n\}, \;\; \sigma(t) = f_{\sigma,t}([\mathbf{Z}]_1^{t-1}), \tag{1}$$

where we use brackets $[\,\cdot\,]_1^\tau$ to denote the collection from time 1 to τ. Namely, at every time t, scheduling is decided based on the network-wide channel state information (CSI) up to time $(t-1)$. If source s is scheduled, then it can send a linear combination of any packets. That is,

$$\text{If } \sigma(t) = s, \text{ then } X_s(t) = \mathbf{c}_t \mathbf{W}^\top \text{ for some } \mathbf{c}_t \in \Omega, \tag{2}$$

where $\mathbf{c}_t$ is a row coding vector in Ω. The choice of $\mathbf{c}_t$ depends on the past CSI vectors $[\mathbf{Z}]_1^{t-1}$, and we assume that $\mathbf{c}_t$ is known causally to the entire network.[1] Therefore, decoding can be performed by simple Gaussian elimination.

We now define two important linear space concepts: The *individual message subspace* and the *knowledge subspace*. To that end, we first define $\mathbf{e}_l$ as an $n(R_1+R_2)$-dimensional elementary row vector with its l-th coordinate being one and all the other coordinates being zero. Recall that the $n(R_1+R_2)$ coordinates of a vector in Ω can be divided into 2 consecutive "intervals", each of them corresponds to the information packets $\mathbf{W}_k$ for each flow from source to destination d_k. We then define the *individual message subspace* Ω_k:

$$\Omega_k \triangleq \mathsf{span}\{\mathbf{e}_l : l \in \text{"interval" associated to } \mathbf{W}_k\}, \tag{3}$$

That is, Ω_k is a linear subspace corresponding to any linear combination of $\mathbf{W}_k$ packets. By (3), each Ω_k is a linear subspace of the overall message space Ω and $\mathsf{rank}(\Omega_k) = nR_k$.

We define the knowledge space for $\{d_1, d_2, r\}$. The knowledge space $S_h(t)$ in the end of time t is defined by

$$S_h(t) \triangleq \mathsf{span}\{\mathbf{c}_\tau : \forall \tau \le t \text{ s.t. node } h \text{ receives the linear combination } (\mathbf{c}_\tau \cdot \mathbf{W}^\top) \text{ successfully in time } \tau\} \tag{4}$$

where $h \in \{d_1, d_2, r\}$. For example, $S_r(t)$ is the linear space spanned by the packets successfully delivered from source to relay up to time t. $S_{d_1}(t)$ is the linear space spanned by the packets received at destination d_1 up to time t, either transmitted by source or by relay.

For shorthand, we use $S_1(t)$ and $S_2(t)$ instead of $S_{d_1}(t)$ and $S_{d_2}(t)$, respectively. Then, by the above definitions, we quickly have that destination d_k can decode the desired packets $\mathbf{W}_k$ as long as $S_k(n) \supseteq \Omega_k$. That is, when the knowledge space in the end of time n contains the desired message space.

With the above linear space concepts, we now can describe the packet transmission from relay. Recall that, unlike the source where the packets are originated, relay can only send a linear mixture of *the packets that it has known.* Therefore, the encoder description from relay can be expressed by

$$\text{If } \sigma(t) = r, \text{ then } X_r(t) = \mathbf{c}_t \mathbf{W}^\top \text{ for some } \mathbf{c}_t \in S_r(t-1). \tag{5}$$

For comparison, in (2), the source s chooses $\mathbf{c}_t$ from Ω. We can now define the LNC capacity region.

Definition 1: Fix the distribution of $\mathbf{Z}(t)$ and finite field $\mathbb{F}_q$. A rate vector (R_1, R_2) is achievable by LNC if for any $\epsilon > 0$ there exists a joint scheduling and LNC scheme with sufficiently large n such that $\mathsf{Prob}(S_k(n) \supseteq \Omega_k) > 1 - \epsilon$ for all $k \in \{1, 2\}$. The LNC capacity region is the closure of all LNC-achievable (R_1, R_2).

B. A Useful Notation

In our network model, there are two broadcast PECs associated with s and r. For shorthand, we call those PECs the s-PEC and the r-PEC, respectively. The distribution of the network-wide channel status vector $\mathbf{Z}(t) = (\mathbf{Z}_s(t), \mathbf{Z}_r(t))$

[1] Coding vector $\mathbf{c}_t$ can either be appended in the header or be computed by the network-wide causal CSI feedback $[\mathbf{Z}]_1^{t-1}$.

can be described by the probabilities $p_{s\to T\overline{\{d_1,d_2,r\}\backslash T}}$ for all $T\subseteq\{d_1,d_2,r\}$, and $p_{r\to U\overline{\{d_1,d_2\}\backslash U}}$ for all $U\subseteq\{d_1,d_2\}$. In total, there are $8+4=12$ channel parameters.[2]

For notational simplicity, we also define the following two probability functions $p_s(\cdot)$ and $p_r(\cdot)$, one for each PEC. The input argument of p_s is a collection of the elements in $\{d_1,d_2,r,\overline{d_1},\overline{d_2},\overline{r}\}$. The function $p_s(\cdot)$ outputs the probability that the reception event is compatible to the specified collection of $\{d_1,d_2,r,\overline{d_1},\overline{d_2},\overline{r}\}$. For example, $p_s(d_2\overline{r}) = p_{s\to\overline{d_1}d_2\overline{r}}+p_{s\to d_1d_2\overline{r}}$ is the probability that the input of the s-PEC is successfully received by d_2 but not by r. Herein, d_1 is a don't-care receiver and $p_s(d_2\overline{r})$ thus sums two joint probabilities together (whether d_1 receives it or not). Another example is $p_r(d_2)=p_{r\to d_1d_2}+p_{r\to\overline{d_1}d_2}$, which is the marginal probability from r to d_2. To slightly abuse the notation, we further allow $p_s(\cdot)$ and $p_r(\cdot)$ to take multiple input arguments separated by commas. With this new notation, they can represent the probability that the reception event is compatible to at least one of the input arguments. For example, $p_s(d_1,d_2,r)$ represents the probability that a packet sent by s is received by at least one of the three nodes d_1, d_2, and r.

III. Main Results

A. LNC Capacity Outer Bound

Since the coding vector $\mathbf{c}_t$ has $n(R_1+R_2)$ number of coordinates, there are exponentially many ways of jointly designing the scheduling $\sigma(t)$ and the coding vector choices $\mathbf{c}_t$ over time when sufficiently large n and $\mathbb{F}_q$ are used. Therefore, we will first simplify the aforementioned design choices by comparing $\mathbf{c}_t$ to the knowledge spaces $S_h(t-1)$, $h\in\{d_1,d_2,r\}$. Such a simplification allows us to derive Proposition 1, which uses a linear programming (LP) solver to exhaustively search over the entire coding and scheduling choices and thus computes an LNC capacity outer bound.

To that end, we use S_k as shorthand for $S_k(t-1)$, the knowledge space of destination d_k in the end of time $t-1$. We first define the following 7 linear subspaces of Ω.

$$A_1(t)\triangleq S_1,\qquad A_2(t)\triangleq S_2, \tag{6}$$
$$A_3(t)\triangleq S_1\oplus\Omega_1,\qquad A_4(t)\triangleq S_2\oplus\Omega_2, \tag{7}$$
$$A_5(t)\triangleq S_1\oplus S_2, \tag{8}$$
$$A_6(t)\triangleq S_1\oplus S_2\oplus\Omega_1,\qquad A_7(t)\triangleq S_1\oplus S_2\oplus\Omega_2, \tag{9}$$

where $A\oplus B\triangleq\mathsf{span}\{\mathbf{v}:\mathbf{v}\in A\cup B\}$ is the *sum space* of any $A,B\subseteq\Omega$. In addition, we also define the following eight additional subspaces involving $S_r(t-1)$:

$$A_{i+7}(t)\triangleq A_i(t)\oplus S_r\quad\text{for all } i=1,\cdots,7, \tag{10}$$
$$A_{15}(t)\triangleq S_r, \tag{11}$$

where S_r is a shorthand notation for $S_r(t-1)$, the knowledge space of relay r in the end of time $t-1$.

In total, there are $7+8=15$ linear subspaces of Ω. We then partition the overall message space Ω into 2^{15} disjoint subsets by the *Venn diagram* generated by these 15 subspaces. That is, for any given coding vector $\mathbf{c}_t$, we can place it in exactly one of the 2^{15} disjoint subsets by testing whether it belongs to which A-subspaces. This is always true regardless of the time index t, i.e., any coding vector $\mathbf{c}_t$ transmitted by either source or relay always lies in one of the 2^{15} disjoint subsets while the regions of disjoint subsets may change over the course of time. In the following discussion, we often drop the input argument "(t)" when the time instant of interest is clear in the context.

We now use 15 bits to represent each disjoint subset of the overall message space Ω. For any 15-bit string $\mathbf{b}=b_1b_2\cdots b_{15}$, we define "the coding type-$\mathbf{b}$" by

$$\mathsf{TYPE}^{(s)}_{\mathbf{b}}\triangleq\Big(\bigcap_{l:b_l=1}A_l\Big)\setminus\Big(\bigcup_{l:b_l=0}A_l\Big). \tag{12}$$

The superscript "(s)" indicates that source s can send $\mathbf{c}_t$ from any of these 2^{15} types since source s knows all $\mathbf{W}_1$ and $\mathbf{W}_2$ packets to begin with. Note that not all 2^{15} disjoint subsets are feasible. For example, any $\mathsf{TYPE}^{(s)}_{\mathbf{b}}$ with $b_7=1$ but $b_{14}=0$ is always empty because any coding vector that lies in $A_7=S_1\oplus S_2\oplus\Omega_2$ cannot lie outside the larger $A_{14}=S_1\oplus S_2\oplus S_r\oplus\Omega_2$, see (9) and (10), respectively. We say those always empty subsets are *infeasible coding types* and the rest is called *feasible coding types* (FTs). By exhaustive computer search, we can prove that out of $2^{15}{=}32768$ subsets, only 154 of them are feasible. Namely, the entire coding space Ω can be viewed as a union of 154 disjoint coding types. Source s can choose a coding vector $\mathbf{c}_t$ from one of these 154 types. See (2).

For coding vectors that relay r can choose, we can further reduce the number of possible placements of $\mathbf{c}_t$ in the following way. By (5), we know that when $\sigma(t)=r$, the $\mathbf{c}_t$ sent by relay must belong to its knowledge space $S_r(t-1)$. Hence, such $\mathbf{c}_t$ must always lie in $S_r(t-1)$, which is $A_{15}(t)$, see (11). As a result, any coding vector $\mathbf{c}_t$ sent by relay r must lie in those 154 subsets FTs that satisfy:

$$\mathsf{TYPE}^{(r)}_{\mathbf{b}}\triangleq\{\mathsf{TYPE}^{(s)}_{\mathbf{b}}:\mathbf{b}\in\mathsf{FTs}\text{ such that } b_{15}=1\}. \tag{13}$$

Again by computer search, there are 18 such coding types out of 154 subsets FTs. We call those 18 subsets as *relay's feasible coding types* (rFTs). Obviously, $r\mathsf{FTs}\subseteq\mathsf{FTs}$. See Appendix A for the enumeration of those FTs and rFTs.

We can then derive the following upper bound.

Proposition 1: A rate vector (R_1,R_2) is in the LNC capacity region only if there exists 154 non-negative variables $x^{(s)}_{\mathbf{b}}$ for all $\mathbf{b}\in\mathsf{FTs}$, 18 non-negative variables $x^{(r)}_{\mathbf{b}}$ for all $\mathbf{b}\in r\mathsf{FTs}$, and 14 non-negative y-variables, y_1 to y_{14}, such that jointly they satisfy the following three groups of linear conditions:

- Group 1, termed the *time-sharing condition*, has 1 inequality:

$$\Big(\sum_{\forall\mathbf{b}\in\mathsf{FTs}}x^{(s)}_{\mathbf{b}}\Big)+\Big(\sum_{\forall\mathbf{b}\in r\mathsf{FTs}}x^{(r)}_{\mathbf{b}}\Big)\le 1. \tag{14}$$

- Group 2, termed the *rank-conversion conditions*, has 14 equalities:

$$y_{0+k}=\Big(\sum_{\forall\mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{0+k}=0}x^{(s)}_{\mathbf{b}}\cdot p_s(d_k)\Big)+\Big(\sum_{\forall\mathbf{b}\in r\mathsf{FTs}\text{ s.t. }b_{0+k}=0}x^{(r)}_{\mathbf{b}}\cdot p_r(d_k)\Big),\quad\text{for all } k\in\{1,2\}, \tag{15}$$

[2] By allowing some coordinates of $\mathbf{Z}(t)$ to be correlated (i.e., spatially correlated as it is between coordinates, not over the time axis), our setting can also model the scenario in which d_1 and d_2 are situated in the same physical node and thus have perfectly correlated channel success events.

$$y_{2+k}=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{2+k}=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_k)\Big)+\Big(\sum_{\forall \mathbf{b}\in r\mathsf{FTs}\text{ s.t. }b_{2+k}=0} x^{(r)}_{\mathbf{b}}\cdot p_r(d_k)\Big)$$
$$+R_k,\ \text{for all } k\in\{1,2\}, \quad (16)$$

$$y_5=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_5=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_1,d_2)\Big)+\Big(\sum_{\forall \mathbf{b}\in r\mathsf{FTs}\text{ s.t. }b_5=0} x^{(r)}_{\mathbf{b}}\cdot p_r(d_1,d_2)\Big), \quad (17)$$

$$y_{5+k}=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{5+k}=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_1,d_2)\Big)+\Big(\sum_{\forall \mathbf{b}\in r\mathsf{FTs}\text{ s.t. }b_{5+k}=0} x^{(r)}_{\mathbf{b}}\cdot p_r(d_1,d_2)\Big)$$
$$+R_k,\ \text{for all } k\in\{1,2\}, \quad (18)$$

$$y_{7+k}=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{7+k}=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_k,r)\Big),\ \text{for all } k\in\{1,2\}, \quad (19)$$

$$y_{9+k}=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{9+k}=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_k,r)\Big)+R_k,\ \text{for all } k\in\{1,2\}, \quad (20)$$

$$y_{12}=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{12}=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_1,d_2,r)\Big), \quad (21)$$

$$y_{12+k}=\Big(\sum_{\forall \mathbf{b}\in\mathsf{FTs}\text{ s.t. }b_{12+k}=0} x^{(s)}_{\mathbf{b}}\cdot p_s(d_1,d_2,r)\Big)+R_k,\ \forall k\in\{1,2\}, \quad (22)$$

• Group 3, termed the *decodability conditions*, has 5 equalities:

$$y_1=y_3,\quad y_2=y_4,\quad y_8=y_{11},\quad y_9=y_{11}, \quad (23)$$
$$y_5=y_6=y_7=y_{12}=y_{13}=y_{14}=(R_1+R_2). \quad (24)$$

The intuition is as follows. Consider any achievable $\vec{R}$ and the associated LNC scheme. In the beginning of any time t, we can compute the knowledge spaces $S_1(t-1)$, $S_2(t-1)$, and $S_r(t-1)$ by (4) and use them to compute the A-subspaces in (6)–(11). Then suppose that for time t, the given scheme chooses source s to transmit a coding vector $\mathbf{c}_t$. By the previous discussions, we can classify which $\mathsf{TYPE}^{(s)}_{\mathbf{b}}$ this $\mathbf{c}_t$ belongs to, by comparing it to those 15 A-subspaces. After running the given scheme from time 1 to n, we can thus compute the variable $x^{(s)}_{\mathbf{b}} \triangleq \frac{1}{n}\mathbb{E}\left[\sum_{t=1}^n 1_{\{\mathbf{c}_t\in\mathsf{TYPE}^{(s)}_{\mathbf{b}}\}}\right]$ for each $\mathsf{TYPE}^{(s)}_{\mathbf{b}}$ as the *frequency* of scheduling source s with the chosen $\mathbf{c}_t$ happening to be in $\mathsf{TYPE}^{(s)}_{\mathbf{b}}$. Similarly for $\mathsf{TYPE}^{(r)}_{\mathbf{b}}$, we can compute the variable $x^{(r)}_{\mathbf{b}} \triangleq \frac{1}{n}\mathbb{E}\left[\sum_{t=1}^n 1_{\{\mathbf{c}_t\in\mathsf{TYPE}^{(r)}_{\mathbf{b}}\}}\right]$ for each $\mathsf{TYPE}^{(r)}_{\mathbf{b}}$ as the *frequency* of scheduling relay r with the chosen $\mathbf{c}_t$ happening to be in $\mathsf{TYPE}^{(r)}_{\mathbf{b}}$. Obviously, the computed $\{x^{(s)}_{\mathbf{b}}, x^{(r)}_{\mathbf{b}}\}$ satisfy the time-sharing inequality (14). We then compute the y-variables by

$$y_l \triangleq \frac{1}{n}\mathbb{E}\left[\mathsf{rank}\big(A_i(n)\big)\right],\ \forall l\in\{1,2,\cdots,14\}, \quad (25)$$

as normalized expected ranks of A-subspaces in the end of time n. We now claim that these variables satisfy (15) to (24). This claim implies that for any LNC-achievable (R_1,R_2), there exists $x^{(s)}_{\mathbf{b}}$, $x^{(r)}_{\mathbf{b}}$, and y-variables satisfying Proposition 1, thus constituting an outer bound on the LNC capacity.

To prove that (15) to (22) are true,[3] consider an A-subspace, say $A_3(t)=S_1(t-1)\oplus\Omega_1=RS_{d_1}(t-1)\oplus\Omega_1$ as defined in (7) and (4). In the beginning of time 1, destination d_1 has not received any packet yet, i.e., $RS_{d_1}(0)=\{\mathbf{0}\}$. Thus the rank of $A_3(1)$ is $\mathsf{rank}(\Omega_1)=nR_1$.

The fact that $S_1(t-1)$ contributes to $A_3(t)$ implies that $\mathsf{rank}(A_3(t))$ will increase by one whenever destination d_1 receives a packet $\mathbf{c}_t\mathbf{W}^\top$ satisfying $\mathbf{c}_t\notin A_3(t)$. Whenever source s sends a $\mathbf{c}_t$ in $\mathsf{TYPE}^{(s)}_{\mathbf{b}}$ with $b_3=0$, such $\mathbf{c}_t$ is not in $A_3(t)$. Whenever destination d_1 receives it, $\mathsf{rank}(A_3(t))$ increases by 1. Moreover, whenever relay r sends a $\mathbf{c}_t$ in $\mathsf{TYPE}^{(r)}_{\mathbf{b}}$ with $b_3=0$ and destination d_1 receives it, $\mathsf{rank}(A_3(t))$ also increases by 1. Therefore, in the end of time n, we have

$$\begin{aligned}\mathsf{rank}(A_3(n)) &= \sum_{t=1}^n 1_{\left\{\substack{\text{source } s \text{ sends } \mathbf{c}_t\in\mathsf{TYPE}^{(s)}_{\mathbf{b}} \text{ with } b_3=0,\\ \text{and destination } d_1 \text{ receives it}}\right\}}\\ &+\sum_{t=1}^n 1_{\left\{\substack{\text{relay } r \text{ sends } \mathbf{c}_t\in\mathsf{TYPE}^{(r)}_{\mathbf{b}} \text{ with } b_3=0,\\ \text{and destination } d_1 \text{ receives it}}\right\}}\\ &+\mathsf{rank}(A_3(0)).\end{aligned} \quad (26)$$

Taking the normalized expectation of (26), we have proven (16). By similar *rank-conversion* arguments, (15) to (22) can be shown to be true.

In the end of time n, since both d_1 and d_2 can decode the desired packets $\mathbf{W}_1$ and $\mathbf{W}_2$, respectively, we have $S_1(n)\supseteq\Omega_1$ and $S_2(n)\supseteq\Omega_2$, or equivalently $S_k(n)=S_k(n)\oplus\Omega_k$ for all $k\in\{1,2\}$. This implies that the ranks of $A_1(n)$ and $A_3(n)$, and the ranks of $A_2(n)$ and $A_4(n)$ are equal, respectively. Together with (25), we thus have the first two equalities in (23). Similarly, one can prove that the remaining equalities in (23) and (24) are satisfied as well. The claim is thus proven.

B. LNC Capacity Inner Bound

In the smart repeater problem of our interest, if the r-PEC is weaker than the s-PEC, then there is no need to do relaying since we can simply let s take over relay's operations. We thus assume

Definition 2: The smart repeater network with $\{d_1,d_2\}$ is *strong-relaying* if

$$\begin{aligned}p_r(d_1) &> p_s(d_1),\\ p_r(d_1\overline{d_2}) &> p_s(d_1\overline{d_2}),\\ p_r(d_2) &> p_s(d_2),\\ p_r(\overline{d_1}d_2) &> p_s(\overline{d_1}d_2),\\ \text{and } p_r(d_1,d_2) &> p_s(d_1,d_2).\end{aligned}$$

That is, the given r-PEC is stronger than the given s-PEC for all non-empty subsets of two destinations $\{d_1,d_2\}$.

We now describe our capacity-approaching achievability scheme.

Proposition 2: A rate vector (R_1,R_2) is LNC-achievable if there exist 2 non-negative variables t_s and t_r, $(6\times 2+8)$ non-negative s-variables:

$$\big\{s^k_{\mathsf{UC}},\ s^k_{\mathsf{PM1}},\ s^k_{\mathsf{PM2}},\ s^k_{\mathsf{RC}},\ s^k_{\mathsf{DX}},\ s^{(k)}_{\mathsf{DX}}\ :\ \text{for all } k\in\{1,2\}\big\},$$
$$\big\{s_{\mathsf{CX};l}\,(l=1,\cdots,8)\big\},$$

and $(3\times 2+3)$ non-negative r-variables:

$$\big\{r^k_{\mathsf{UC}},\ r^{(k)}_{\mathsf{DT}},\ r^{[k]}_{\mathsf{DT}}\ :\ \text{for all } k\in\{1,2\}\big\},\ \ \big\{r_{\mathsf{RC}},\ r_{\mathsf{XT}},\ r_{\mathsf{CX}}\big\},$$

[3] For rigorous proofs, we need to invoke the law of large numbers and take care of the ϵ-error probability. For ease of discussion, the corresponding technical details are omitted when discussing the intuition of Proposition 1.

such that jointly they satisfy the following five groups of linear conditions:

• Group 1, termed the *time-sharing conditions*, has 3 inequalities:

$$1 > t_s + t_r, \tag{27}$$

$$t_s \geq \sum_{k\in\{1,2\}} \left(s^k_{\mathsf{UC}} + s^k_{\mathsf{PM1}} + s^k_{\mathsf{PM2}} + s^k_{\mathsf{RC}} + s^k_{\mathsf{DX}} + s^{\textcircled{k}}_{\mathsf{DX}} \right) + \sum_{l=1}^{8} s_{\mathsf{CX};l}, \tag{28}$$

$$t_r \geq \sum_{k\in\{1,2\}} \left(r^k_{\mathsf{UC}} + r^{\textcircled{k}}_{\mathsf{DT}} + r^{[k]}_{\mathsf{DT}} \right) + r_{\mathsf{RC}} + r_{\mathsf{XT}} + r_{\mathsf{CX}}. \tag{29}$$

• Group 2, termed the *packets-originating condition*, has 2 inequalities: Consider any $i, j \in \{1, 2\}$ satisfying $i \neq j$. For each (i, j) pair (out of the two choices $(1, 2)$ and $(2, 1)$),

$$R_i \geq \left(s^i_{\mathsf{UC}} + s^i_{\mathsf{PM1}} \right) \cdot p_s(d_i, d_j, r), \tag{E}$$

• Group 3, termed the *packets-mixing condition*, has 4 inequalities: For each (i, j) pair,

$$\begin{aligned}\left(s^i_{\mathsf{UC}} + s^i_{\mathsf{PM1}} \right) \cdot p_{s\to\overline{d_i d_j} r} \geq (s^j_{\mathsf{PM1}} + s^i_{\mathsf{PM2}}) \cdot p_s(d_i, d_j) \\ + r^i_{\mathsf{UC}} \cdot p_r(d_i, d_j),\end{aligned} \tag{A}$$

$$s^i_{\mathsf{PM1}} \cdot p_{s\to\overline{d_i} d_j \overline{r}} \geq s^i_{\mathsf{RC}} \cdot p_s(d_i, d_j, r), \tag{B}$$

and the following one inequality:

$$\begin{aligned}&s^1_{\mathsf{PM1}} \cdot p_s(d_1, d_2 r) + s^2_{\mathsf{PM1}} \cdot p_s(d_2, d_1 r) + s^1_{\mathsf{PM2}} \cdot p_s(\overline{d_1} d_2) + \\ &s^2_{\mathsf{PM2}} \cdot p_s(d_1 \overline{d_2}) + \left(s^1_{\mathsf{RC}} + s^2_{\mathsf{RC}} \right) \cdot p_{s\to\overline{d_1 d_2} r} \geq r_{\mathsf{RC}} \cdot p_r(d_1, d_2).\end{aligned} \tag{M}$$

• Group 4, termed the *classic XOR condition by source only*, has 4 inequalities: For each (i, j) pair,

$$\begin{aligned}\left(s^i_{\mathsf{UC}} + s^i_{\mathsf{RC}} \right) p_{s\to\overline{d_i} d_j \overline{r}} \geq \left(s^j_{\mathsf{PM2}} + s^i_{\mathsf{DX}} \right) \cdot p_s(d_i, r) + \\ (s_{\mathsf{CX};1} + s_{\mathsf{CX};1+i}) \cdot p_s(d_i, r) + s_{\mathsf{CX};4+i} \cdot p_s(d_i, r),\end{aligned} \tag{S}$$

$$\begin{aligned}s^j_{\mathsf{RC}} \cdot p_{s\to\overline{d_i} d_j \overline{r}} \geq s^{\textcircled{i}}_{\mathsf{DX}} \cdot p_s(d_i, r) + r^{\textcircled{i}}_{\mathsf{DT}} \cdot p_r(d_i, d_j) + \\ (s_{\mathsf{CX};1+j} + s_{\mathsf{CX};4}) \cdot p_s(d_i, r) + s_{\mathsf{CX};6+i} \cdot p_s(d_i, r).\end{aligned} \tag{T}$$

• Group 5, termed the *XOR condition*, has 3 inequalities:

$$\sum_{l=1}^{4} s_{\mathsf{CX};l} \cdot p_{s\to\overline{d_1 d_2} r} \geq r_{\mathsf{XT}} \cdot p_r(d_1, d_2), \tag{X0}$$

and for each (i, j) pair,

$$\begin{aligned}&s^j_{\mathsf{PM2}} \cdot p_s(d_i d_j, \overline{d_i} r) + \left(s^i_{\mathsf{UC}} + s^i_{\mathsf{RC}} + s^j_{\mathsf{RC}} + \sum_{l=1}^{4} s_{\mathsf{CX};l} \right) \cdot p_{s\to\overline{d_i} d_j r} \\ &+ \left(s_{\mathsf{CX};4+i} + s_{\mathsf{CX};6+i} + s^i_{\mathsf{DX}} + s^{\textcircled{i}}_{\mathsf{DX}} \right) \cdot p_s(\overline{d_i} r) \\ &+ \left(r^i_{\mathsf{UC}} + r_{\mathsf{RC}} + r^{\textcircled{i}}_{\mathsf{DT}} + r_{\mathsf{XT}} \right) \cdot p_{r\to\overline{d_i} d_j} \\ &\geq (s_{\mathsf{CX};7-i} + s_{\mathsf{CX};9-i}) \cdot p_s(d_i) + \left(r_{\mathsf{CX}} + r^{[i]}_{\mathsf{DT}} \right) \cdot p_r(d_i).\end{aligned} \tag{X}$$

• Group 6, termed the *decodability condition*, has 2 inequalities: For each (i, j) pair,

$$\begin{aligned}&\left(s^i_{\mathsf{UC}} + s^j_{\mathsf{PM2}} + \sum_{k\in\{1,2\}} s^k_{\mathsf{RC}} + \sum_{l=1}^{8} s_{\mathsf{CX};l} + s^i_{\mathsf{DX}} + s^{\textcircled{i}}_{\mathsf{DX}} \right) \cdot p_s(d_i) + \\ &\left(r^i_{\mathsf{UC}} + r_{\mathsf{RC}} + r_{\mathsf{XT}} + r_{\mathsf{CX}} + r^{\textcircled{i}}_{\mathsf{DT}} + r^{[i]}_{\mathsf{DT}} \right) \cdot p_r(d_i) \geq R_i.\end{aligned} \tag{D}$$

The intuition is as follows. The proposed LNC inner bound is derived based on the ideas of describing the packet movements in a queueing network, where movements are governed by LNC operations. Each variable (except t-variables for time-sharing) in Proposition 2 is associated with a specific LNC operation. Note that s-variables are associated with LNC operations performed by the source s, while r-variables are associated with LNC operations performed by the relay r. The inequalities (E) to (D) then describe the queueing process for packet movements, where the LHS and the RHS of each inequality implies the packet insertion and removal conditions, respectively, of the corresponding queue by the related LNC operations. For notational convenience, we define the following queue notations associated with these 14 inequalities (E) to (D):

TABLE I

Queue denominations for the inequalities (E) to (D)

(E1): Q^1_ϕ	(B1): $Q^{m\|2}_{\{d_2\}\|\{r\}}$	(S1): $Q^1_{\{d_2\}}$	(X0): $Q^{m\mathsf{CX}}_{\{r\}}$
(E2): Q^2_ϕ	(B2): $Q^{m\|1}_{\{d_1\}\|\{r\}}$	(T1): $Q^{(1)\|1}_{\{d_2\}\|\{r\}}$	(X1): $Q^{[1]}_{\{r d_2\}}$
(A1): $Q^1_{\{r\}}$	(M): Q_{mix}	(S2): $Q^2_{\{d_1\}}$	(X2): $Q^{[2]}_{\{r d_1\}}$
(A2): $Q^2_{\{r\}}$		(T2): $Q^{(2)\|2}_{\{d_1\}\|\{r\}}$	(D1): Q^1_{dec}
			(D2): Q^2_{dec}

where we use the index-after-reference to distinguish the session (i.e., flow) of focus of an inequality. For example, (E1) and (E2) are to denote the inequality (E) when $(i, j) = (1, 2)$ and $(i, j) = (2, 1)$, respectively.

For example, suppose that $\mathbf{W}_1 = (X_1, \cdots, X_{nR_1})$ packets and $\mathbf{W}_2 = (Y_1, \cdots, Y_{nR_2})$ packets are initially stored in queues Q^1_ϕ and Q^2_ϕ, respectively, at source s. The superscript $k \in \{1, 2\}$ indicates that the queue is for the packets intended to destination d_k. The subscript indicates that those packets have not been heard by any of $\{d_1, d_2, r\}$. The LNC operation corresponding to the variable s^1_{UC} (resp. s^2_{UC}) is to send a session-1 packet X_i (resp. a session-2 packet Y_j) uncodedly. Then the inequality (E1) (resp. (E2)) implies that whenever it is received by at least one of $\{d_1, d_2, r\}$, this packet is removed from the queue of Q^1_ϕ (resp. Q^2_ϕ).

Depending on the reception status, the packet will be either remained in the same queue or moved to another queue. For example, the use of s^1_{UC} (sending $X_i \in \mathbf{W}_1$ uncodedly from source) will take X_i from Q^1_ϕ and insert it into Q^1_{dec} when the reception status is $p_s(d_1)$, i.e., when the intended destination d_1 correctly receives it. Similarly, when the reception status is $p_{s\to\overline{d_1 d_2} r}$, this packet will be inserted to the queue $Q^1_{\{r\}}$ according to the packet movement rule of (A1); inserted to $Q^1_{\{d_2\}}$ when $p_{s\to\overline{d_1} d_2 \overline{r}}$ by (S1); and inserted to $Q^{[1]}_{\{r d_2\}}$ when $p_{s\to\overline{d_1} d_2 r}$ by (X1). Obviously when $p_{s\to\overline{d_1 d_2 r}}$, since any node in $\{d_1, d_2, r\}$ has not received at all, the packet X_i simply remains in Q^1_ϕ.

Fig. 2 illustrates the queueing network represented by Proposition 2. The detailed descriptions of the proposed LNC operations and the corresponding packet movement process following the inequalities in Proposition 2 are relegated to Appendix B.

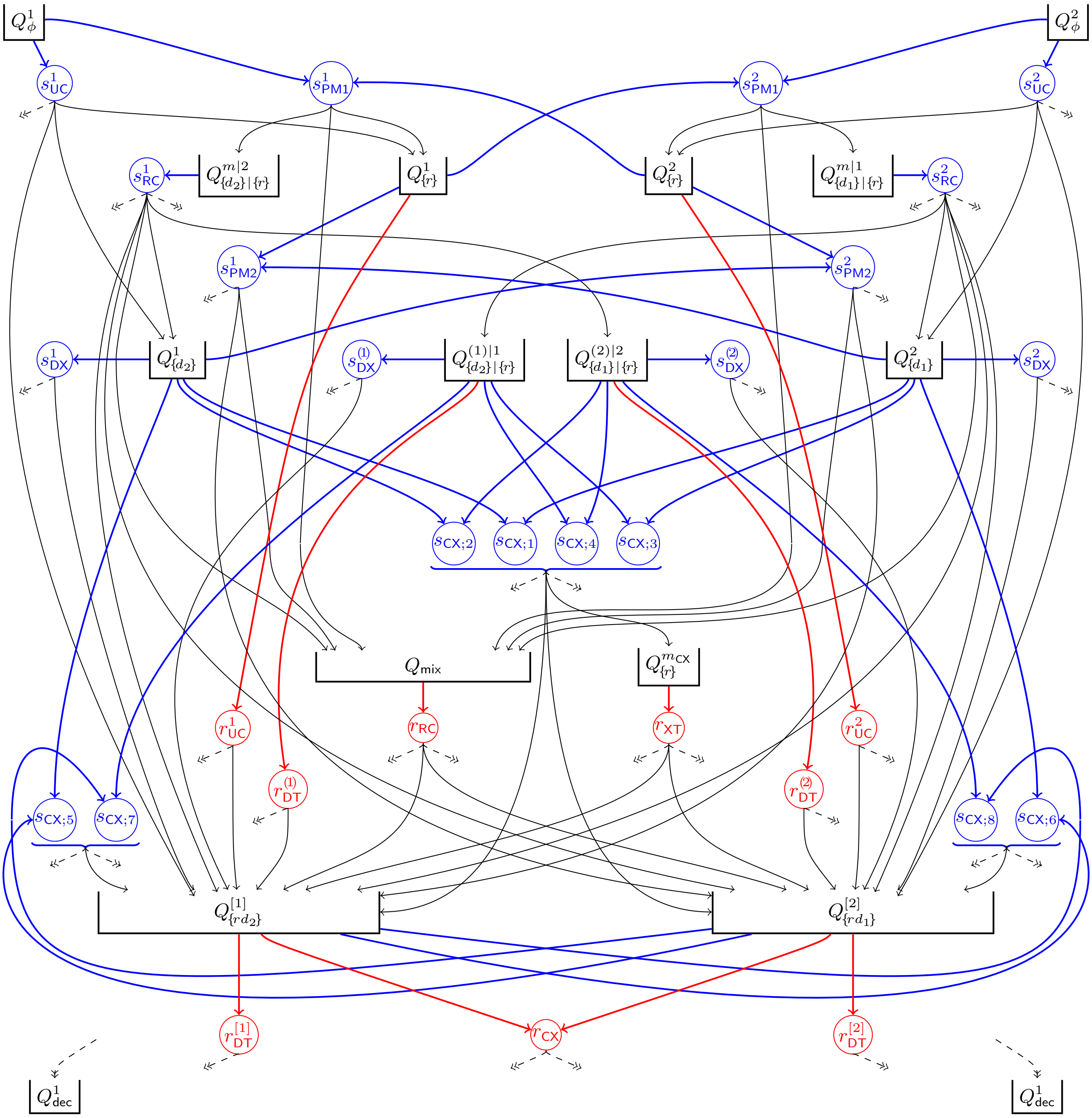


Fig. 2: Illustrations of The Queueing Network described by the inequalities (E) to (D) in Proposition 2. The upper-side-open rectangle represents the queue, and the circle represents LNC encoding operation, where the blue means the encoding by the source s and the red means the encoding by the relay r. The black outgoing arrows from a LNC operation (or from a set of LNC operations grouped by a curly brace) represent the packet movements process depending on the reception status, where the southwest and southeast dashed arrows are especially for into Q^1_{dec} and into Q^2_{dec}, respectively.

C. The Properties of Queues and The Correctness Proof

Each queue in the queueing network, see Fig. 2, is carefully designed to store packets in a specific format such that the queue itself can represent a certain case to be beneficial. In this subsection, we highlight the properties of the queues, which will be later used to prove the correctness of our achievability scheme of Proposition 2.

To that end, we first describe the properties of Q^1_ϕ, Q^1_{dec}, $Q^1_{\{r\}}$, and $Q^1_{\{d_2\}}$ since their purpose is clear in the sense that the queue collects pure session-1 packets (indicated by the superscript), but heard only by the nodes (in the subscript $\{\cdot\}$) or correctly decoded by the desired destination d_1 (by the subscript dec). After that, we describe the property of Q_{mix}, and then explain $Q^{m|2}_{\{d_2\}|\{r\}}$, $Q^{(1)|1}_{\{d_2\}|\{r\}}$, and $Q^{[1]}_{\{rd_2\}}$ focusing on the queues related to the session-1 packets. For example, $Q^{m|2}_{\{d_2\}|\{r\}}$ implies the queue related to a session-1 packet that is mixed with a session-2 packet, where such mixture is known by d_2 but the session-2 packet is known by r as well. The properties of the queues related to the session-2 packets, i.e., Q^2_ϕ, Q^2_{dec}, $Q^2_{\{r\}}$, $Q^2_{\{d_1\}}$, $Q^{m|1}_{\{d_1\}|\{r\}}$, $Q^{(2)|2}_{\{d_1\}|\{r\}}$, and $Q^{[2]}_{\{rd_1\}}$, will be symmetrically explained by simultaneously swapping (a) session-1 and session-2 in the superscript; (b) X and Y; (c) i and j; and (d) d_1 and d_2, if applicable. The property of $Q^{m\mathsf{cx}}_{\{r\}}$ will be followed at last.

To help aid the explanations, we also define for each node in $\{d_1, d_2, r\}$, the *reception list* $\mathsf{RL}_{\{d_1\}}$, $\mathsf{RL}_{\{d_2\}}$, and $\mathsf{RL}_{\{r\}}$, respectively, that records how the received packet is constituted. The reception list is a binary matrix of its column size fixed to $n(R_1+R_2)$ but its row size being the number of received packets and thus variable (increasing) over the course of total time slots. For example, suppose that d_1 has received a pure session-1 packet X_1, a self-mixture $[X_1+X_2]$, and a cross-mixture $[X_3+Y_1]$. Then $\mathsf{RL}_{\{d_1\}}$ will be

$$\begin{array}{cc} \overbrace{\begin{matrix} 1\,0\cdots & \cdots & \cdots & \cdots \\ 1\,1\,0\cdot & \cdots & \cdots & \cdots \\ 0\,0\,1\,0\cdots & \cdots & \cdots & \end{matrix}}^{nR_1} & \overbrace{\begin{matrix} 0\,0\cdots & \cdots & \cdots & \cdots \\ 0\,0\cdots & \cdots & \cdots & \cdots \\ 1\,0\cdots & \cdots & \cdots & \cdots \end{matrix}}^{nR_2} \end{array}$$

such that the first row vector represents the pure X_1 received, the second row vector represents the mixture $[X_1 + X_2]$ received, and the third row vector represents the mixture $[X_3+Y_1]$ received, all in a binary format. Namely, whenever a node receives a packet, whether such packet is pure or not, a new $n(R_1+R_2)$-dimensional row vector is inserted into the reception list by marking the corresponding entries of X_i or Y_j as flagged ("1") or not flagged ("0") accordingly. From the previous example, $[X_1+X_2]$ in the reception list $\mathsf{RL}_{\{d_1\}}$ means that the list contains a $n(R_1+R_2)$-dimensional row vector of exactly $\{1, 1, 0, \cdots, 0\}$. We then say that a pure packet is *not flagged* in the reception list, if the column of the corresponding entry contains all zeros. From the previous example, the pure session-2 packet Y_2 is not flagged in $\mathsf{RL}_{\{d_1\}}$, meaning that d_1 has neither received Y_2 nor any mixture involving this Y_2. Note that "not flagged" is a stronger definition than "unknown". From the previous example, the pure session-1 packet X_3 is unknown to d_1 but still flagged in $\mathsf{RL}_{\{d_1\}}$ as d_1 has received the mixture $[X_3+Y_1]$ involving this X_3. Another example is the pure X_2 that is flagged in $\mathsf{RL}_{\{d_1\}}$ but d_1 knows this X_2 as it can use the received X_1 and the mixture $[X_1+X_2]$ to extract X_2. We sometimes abuse the reception list notation to denote the collective reception list by RL_T for some non-empty subset $T \subseteq \{d_1, d_2, r\}$. For example, $\mathsf{RL}_{\{d_1,d_2,r\}}$ implies the vertical concatenation of all $\mathsf{RL}_{\{d_1\}}$, $\mathsf{RL}_{\{d_2\}}$, and $\mathsf{RL}_{\{r\}}$.

We now describe the properties of the queues.

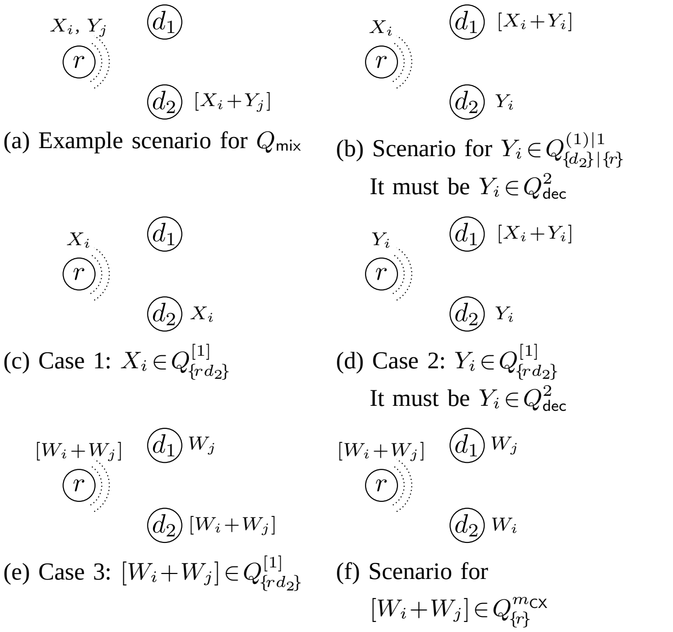
- Q^1_ϕ: Every packet in this queue is *of a pure session-*1 and *unknown* to any of $\{d_1, d_2, r\}$, even *not flagged* in $\mathsf{RL}_{\{d_1,d_2,r\}}$. Initially, this queue contains all the session-1 packets $\mathbf{W}_1$, and will be empty in the end.
- Q^1_{dec}: Every packet in this queue is *of a pure session*-1 and *known* to d_1. Initially, this queue is empty but will contain all the session-1 packets $\mathbf{W}_1$ in the end.
- $Q^1_{\{r\}}$: Every packet in this queue is *of a pure session*-1 and *known* by r but *unknown* to any of $\{d_1, d_2\}$, even *not flagged* in $\mathsf{RL}_{\{d_1,d_2\}}$.
- $Q^1_{\{d_2\}}$: Every packet in this queue is *of a pure session*-1 and *known* by d_2 but *unknown* to any of $\{d_1, r\}$, even *not flagged* in $\mathsf{RL}_{\{d_1,r\}}$.
- Q_{mix}: Every packet in this queue is *of a linear sum* $[X_i+Y_j]$ from a session-1 packet X_i and a session-2 packet Y_j such that at least one of the following conditions hold:

  (a) $[X_i + Y_j]$ is in $\mathsf{RL}_{\{d_1\}}$; X_i is *unknown* to d_1; and Y_j is *known* by r but *unknown* to d_2.

  (b) $[X_i + Y_j]$ is in $\mathsf{RL}_{\{d_2\}}$; X_i is *known* by r but *unknown* to d_1; and Y_j is *unknown* to d_2.

The detailed clarifications are as follows. For a NC designer, one important consideration is to generate as many "all-happy" scenarios as possible in an efficient manner so that single transmission benefits both destination simultaneously. One famous example is the *classic XOR* operation that a sender transmits a linear sum $[X_i+Y_j]$ when a session-1 packet X_i is not yet delivered to d_1 but overheard by d_2 and a session-2 packet Y_j is not yet delivered to d_2 but overheard by d_1. Namely, the source s can perform such classic butterfly-style operation of sending the linear mixture $[X_i + Y_j]$ whenever such pair of X_i and Y_j is available. Similarly, Q_{mix} represents such an "all-happy" scenario that the relay r can benefit both destinations simultaneously by sending either X_i or Y_j. For example, suppose that the source s has transmitted a packet mixture $[X_i + Y_j]$ and it is received by d_2 only. And assume that r already knows the individual X_i and Y_j but X_i is unknown to d_1, see Fig. 3(a). This example scenario falls into the second condition of Q_{mix} above. Then sending X_i from the relay r simultaneously enables d_1 to receive the desired X_i and d_2 to decode the desired Y_j by subtracting the received X_i from the known $[X_i + Y_j]$. Q_{mix} collects such all-happy mixtures $[X_i + Y_j]$ that has been received by either d_1 or d_2 or both. In the same scenario, however, notice that r cannot benefit both destinations simultaneously, if r sends Y_j, instead of X_i. As a result, we use the notation $[X_i + Y_j] : W$ to denote the specific packet W (known by r) that r can send to benefit both destinations. In this second condition scenario of Fig. 3(a), Q_{mix} is storing $[X_i + Y_j]\!:\!X_i$.

- $Q^{m|2}_{\{d_2\}|\{r\}}$: Every packet in this queue is *of a linear sum* $[X_i + Y_j]$ from a session-1 packet X_i and a session-2 packet



(a) Example scenario for Q_{mix}

(b) Scenario for $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ It must be $Y_i \in Q^2_{\mathsf{dec}}$

(c) Case 1: $X_i \in Q^{[1]}_{\{rd_2\}}$

(d) Case 2: $Y_i \in Q^{[1]}_{\{rd_2\}}$ It must be $Y_i \in Q^2_{\mathsf{dec}}$

(e) Case 3: $[W_i+W_j] \in Q^{[1]}_{\{rd_2\}}$

(f) Scenario for $[W_i+W_j] \in Q^{m\mathsf{cx}}_{\{r\}}$

Fig. 3: Illustrations of Scenarios of the Queues.

Y_j such that they jointly satisfy the following conditions simultaneously.

(a) $[X_i + Y_j]$ is in $\mathsf{RL}_{\{d_2\}}$.
(b) X_i is *unknown* to any of $\{d_1, d_2, r\}$, even *not flagged* in $\mathsf{RL}_{\{d_1,r\}}$.
(c) Y_j is *known* by r but *unknown* to any of $\{d_1, d_2\}$, even *not flagged* in $\mathsf{RL}_{\{d_1\}}$.

The scenario is the same as in Fig. 3(a) when r not having X_i. In this scenario, we have observed that r cannot benefit both destinations by sending the known Y_j. $Q^{m|2}_{\{d_2\}|\{r\}}$ collects such unpromising $[X_i + Y_j]$ mixtures.

• $Q^{(1)|1}_{\{d_2\}|\{r\}}$: Every packet in this queue is *of a pure session*-2 packet Y_i such that there exists a pure session-1 packet X_i that Y_i is information equivalent to, and they jointly satisfy the following conditions simultaneously.

(a) $[X_i + Y_i]$ is in $\mathsf{RL}_{\{d_1\}}$.
(b) X_i is *known* by r but *unknown* to any of $\{d_1, d_2\}$.
(c) Y_i is *known* by d_2 (i.e. already in Q^2_{dec}) but *unknown* to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{r\}}$.

The concrete explanations are as follows. The main purpose of this queue is basically the same as $Q^1_{\{d_2\}}$, i.e., to store session-1 packet overheard by d_2, so as to be used by the source s for the classic XOR operation with the session-2 counterparts (e.g., any packet in $Q^2_{\{d_1\}}$). Notice that any $X_i \in Q^1_{\{d_2\}}$ is unknown to r and thus r cannot generate the corresponding linear mixture with the counterpart. However, because X_i is unknown to the relay, r cannot even naively deliver X_i to the desired destination d_1. On the other hand, the queue $Q^{(1)|1}_{\{d_2\}|\{r\}}$ here not only allows s to perform the classic XOR operation but also admits naive delivery from r. To that end, consider the scenario in Fig. 3(b). Here, d_1 has received a linear sum $[X_i + Y_i]$. Whenever d_1 receives Y_i (session-2 packet), d_1 can use Y_i and the known $[X_i + Y_i]$ to decode the desired X_i. This Y_i is also known by d_2 (i.e., already in Q^2_{dec}), meaning that Y_i is no more different than a session-1 packet overheard by d_2 but not yet delivered to d_1. Namely, such Y_i can be treated as *information equivalent to* X_i. That is, using this session-2 packet Y_i for the sake of session-1 does not incur any information duplicity because Y_i is already received by the desired destination d_2.[4] For shorthand, we denote such Y_i as $Y_i \equiv X_i$. As a result, the source s can use this Y_i as for session-1 when performing the classic XOR operation with a session-2 counterpart. Moreover, r also knows the pure X_i and thus relay can perform naive delivery for d_1 as well.

• $Q^{[1]}_{\{rd_2\}}$: Every packet in this queue is *of either a pure or a mixed* packet $\overline{W}$ satisfying the following conditions simultaneously.

(a) $\overline{W}$ is *known* by both r and d_2 but *unknown* to d_1.
(b) d_1 can extract a desired session-1 packet when $\overline{W}$ is further received.

Specifically, there are three possible cases based on how the packet $\overline{W} \in Q^{[1]}_{\{rd_2\}}$ is constituted:

Case 1: $\overline{W}$ is a pure session-1 packet X_i. That is, X_i is known by both r and d_2 but unknown to d_1 as in Fig. 3(c). Obviously, d_1 acquires this new X_i when it is further delivered to d_1.

Case 2: $\overline{W}$ is a pure session-2 packet $Y_i \in Q^2_{\mathsf{dec}}$. That is, Y_i is already received by d_2 and known by r as well but unknown to d_1. For such Y_i, as similar to the discussions of $Q^{(1)|1}_{\{d_2\}|\{r\}}$, there exists a session-1 packet X_i still unknown to d_1 where $X_i \equiv Y_i$, and their mixture $[X_i + Y_i]$ is in $\mathsf{RL}_{\{d_1\}}$, see Fig. 3(d). One can easily see that when d_1 further receives this Y_i, d_1 can use the received Y_i and the known $[X_i + Y_i]$ to decode the desired X_i.

Case 3: $\overline{W}$ is a mixed packet of the form $[W_i + W_j]$ where W_i and W_j are pure but generic that can be either a session-1 or a session-2 packet. That is, the linear sum $[W_i + W_j]$ is known by both r and d_2 but unknown to d_1. In this case, W_i is still unknown to d_1 but W_j is already received by d_1 so that whenever $[W_i + W_j]$ is delivered to d_1, W_i can further be decoded. See Fig. 3(e) for details. Specifically, there are two possible subcases depending on whether W_i is of a pure session-1 or of a pure session-2:

– W_i is a session-1 packet X_i. As discussed above, X_i is unknown to d_1 and it is obvious that d_1 can decode the desired X_i whenever $[W_i + W_j]$ is delivered to d_1.
– W_i is a session-2 packet $Y_i \in Q^2_{\mathsf{dec}}$. In this subcase, there exists a session-1 packet X_i (other than W_j in the above Case 3 discussions) still unknown to d_1 where $X_i \equiv Y_i$. Moreover, $[X_i + Y_i]$ is already in $\mathsf{RL}_{\{d_1\}}$. As a result, d_1 can decode the desired X_i whenever $[W_i + W_j]$ is delivered to d_1.

The concrete explanations are as follows. The main purpose of this queue is basically the same as $Q^{(1)|1}_{\{d_2\}|\{r\}}$ but the queue $Q^{[1]}_{\{rd_2\}}$ here allows not only the source s but also the relay r to perform the classic XOR operation. As elaborated above, we

[4]This means that d_2 does not require Y_i any more, and thus s or r can freely use this Y_i in the network to represent not-yet-decoded X_i instead.

have three possible cases depending on the form of the packet $\overline{W} \in Q^{[1]}_{\{rd_2\}}$. Specifically, either a pure session-1 packet $X_i \notin Q^1_{\mathsf{dec}}$ (Case 1) or a pure session-2 packet $Y_i \in Q^2_{\mathsf{dec}}$ (Case 2) or a mixture $[W_i + W_j]$ (Case 3) will be used when either s or r performs the classic XOR operation with a session-2 counterpart. For example, suppose that we have a packet $X \in Q^{[2]}_{\{rd_1\}}$ (Case 2) as a session-2 counterpart. Symmetrically following the Case 2 scenario of $Q^{[1]}_{\{rd_2\}}$ in Fig. 3(d), we know that X has been received by both r and d_1. There also exists a session-2 packet Y still unknown to d_2 where $Y \equiv X$, of which their mixture $[X + Y]$ is already in $\mathsf{RL}_{\{d_2\}}$. For this session-2 counterpart X, consider any packet $\overline{W}$ in $Q^{[1]}_{\{rd_2\}}$. Obviously, the relay r knows both $\overline{W}$ and X by assumption. As a result, either s or r can send their linear sum $[\overline{W} + X]$ as per the classic pairwise XOR operation. Since d_1 already knows X by assumption, such mixture $[\overline{W}+X]$, when received by d_1, can be used to decode $\overline{W}$ and further decode a desired session-1 packet as discussed above. Moreover, if d_2 receives $[\overline{W} + X]$, then d_2 can use the known $\overline{W}$ to extract X and further decode the desired Y since $[X+Y]$ is already in $\mathsf{RL}_{\{d_2\}}$ by assumption.

- $Q^{m_{\mathsf{CX}}}_{\{r\}}$: Every packet in this queue is *of a linear sum* $[W_i + W_j]$ that satisfies the following conditions simultaneously.

(a) $[W_i + W_j]$ is in $\mathsf{RL}_{\{r\}}$.
(b) W_i is *known* by d_2 but *unknown* to any of $\{d_1, r\}$.
(c) W_j is *known* by d_1 but *unknown* to any of $\{d_2, r\}$.

where W_i and W_j are pure but generic that can be either a session-1 or a session-2 packet. Specifically, there are four possible cases based on the types of W_i and W_j packets:

Case 1: W_i is a pure session-1 packet X_i and W_j is a pure session-2 packet Y_j.

Case 2: W_i is a pure session-1 packet X_i and W_j is a pure session-1 packet $X_j \in Q^1_{\mathsf{dec}}$. For the latter X_j packet, as similar to the discussions of $Q^{(1)|1}_{\{d_2\}|\{r\}}$, there also exists a pure session-2 packet Y_j still unknown to d_2 where $Y_j \equiv X_j$ and their mixture $[X_j+Y_j]$ is already in $\mathsf{RL}_{\{d_2\}}$. As a result, later when d_2 decodes this X_j, d_2 can use X_j and the known $[X_j + Y_j]$ to decode the desired Y_j.

Case 3: W_i is a pure session-2 packet $Y_i \in Q^2_{\mathsf{dec}}$ and W_j is a pure session-2 packet Y_j. For the former Y_i packet, there also exists a pure session-1 packet X_i still unknown to d_1 where $X_i \equiv Y_i$ and $[X_i + Y_i]$ is already in $\mathsf{RL}_{\{d_1\}}$. As a result, later when d_1 decodes this Y_i, d_1 can use Y_i and the known $[X_i + Y_i]$ to decode the desired X_i.

Case 3: W_i is a pure session-2 packet $Y_i \in Q^2_{\mathsf{dec}}$ and W_j is a pure session-1 packet $X_j \in Q^1_{\mathsf{dec}}$. For the former Y_i and the latter X_j packets, the discussions follow the Case 3 and Case 2 above, respectively.

The concrete explanations are as follows. This queue represents the "all-happy" scenario as similar to the butterfly-style operation by the relay r, i.e., sending a linear mixture $[W_i + W_j]$ using W_i heard by d_2 and W_j heard by d_1. Originally, r must have known both individuals packets W_i and W_j to generate their linear sum. However, the sender in fact does not need to know both individuals to perform this classic XOR operation. The sender can still do the same operation even though it knows the linear sum $[W_i + W_j]$ only. This possibility only applies to the relay r as all the messages including both individual packets are originated from the source s. As a result, this queue represents such scenario that the relay r only knows the linear sum instead of individuals, as in Fig. 3(f). More precisely, Cases 1 to 4 happen when the source s performed one of four classic XOR operations $s_{\mathsf{CX};1}$ to $s_{\mathsf{CX};4}$, respectively, and the corresponding linear sum is received only by r, see Appendix B for details.

TABLE II
Summary of the associated LNC operations that moves packets into and takes packets out of.

LNC operations $\mapsto$	**Queue**	**$\mapsto$ LNC operations**
	Q^1_ϕ	s^1_{UC}, s^1_{PM1}
s^1_{UC}, s^1_{PM1}	$Q^1_{\{r\}}$	s^2_{PM1}, s^1_{PM2}, r^1_{UC}
s^1_{PM1}	$Q^{m\|2}_{\{d_2\}\|\{r\}}$	s^1_{RC}
s^1_{UC}, s^1_{RC}	$Q^1_{\{d_2\}}$	s^2_{PM2}, s^1_{DX} $s_{\mathsf{CX};1}$, $s_{\mathsf{CX};2}$, $s_{\mathsf{CX};5}$
s^2_{RC}	$Q^{(1)\|1}_{\{d_2\}\|\{r\}}$	$s^{(1)}_{\mathsf{DX}}$, $s_{\mathsf{CX};3}$ $s_{\mathsf{CX};4}$, $s_{\mathsf{CX};7}$, $r^{(1)}_{\mathsf{DT}}$
s^1_{UC}, s^2_{PM2}, s^1_{RC}, s^1_{DX} $s_{\mathsf{CX};5}$, r^1_{UC}, $r^{(1)}_{\mathsf{DT}}$, r_{RC}	$Q^{[1]}_{\{rd_2\}}$ (Case 1)	$s_{\mathsf{CX};6}$, $s_{\mathsf{CX};8}$ $r^{[1]}_{\mathsf{DT}}$, r_{CX}
s^2_{PM2}, s^2_{RC}, $s^{(1)}_{\mathsf{DX}}$ $s_{\mathsf{CX};7}$, r_{RC}	$Q^{[1]}_{\{rd_2\}}$ (Case 2)	
$s_{\mathsf{CX};1}$, $s_{\mathsf{CX};2}$ $s_{\mathsf{CX};3}$, $s_{\mathsf{CX};4}$, r_{XT}	$Q^{[1]}_{\{rd_2\}}$ (Case 3)	
s^1_{UC}, s^1_{PM2}, s^1_{RC}, s^2_{RC} s^1_{DX}, $s^{(1)}_{\mathsf{DX}}$, $\{s_{\mathsf{CX};1}$ to $s_{\mathsf{CX};8}\}$ r^1_{UC}, $r^{(1)}_{\mathsf{DT}}$, $r^{[1]}_{\mathsf{DT}}$ r_{RC}, r_{XT}, r_{CX}	Q^1_{dec}	
s^1_{PM1}, s^2_{PM1}, s^1_{PM2}, s^2_{PM2} s^1_{RC}, s^2_{RC}	Q_{mix}	r_{RC}
$s_{\mathsf{CX};1}$, $s_{\mathsf{CX};2}$, $s_{\mathsf{CX};3}$, $s_{\mathsf{CX};4}$	$Q^{m_{\mathsf{CX}}}_{\{r\}}$	r_{XT}
	Q^2_ϕ	s^2_{UC}, s^2_{PM1}
s^2_{UC}, s^2_{PM1}	$Q^2_{\{r\}}$	s^1_{PM1}, s^2_{PM2}, r^2_{UC}
s^2_{PM1}	$Q^{m\|1}_{\{d_1\}\|\{r\}}$	s^2_{RC}
s^2_{UC}, s^2_{RC}	$Q^2_{\{d_1\}}$	s^1_{PM2}, s^2_{DX} $s_{\mathsf{CX};1}$, $s_{\mathsf{CX};3}$, $s_{\mathsf{CX};6}$
s^1_{RC}	$Q^{(2)\|2}_{\{d_1\}\|\{r\}}$	$s^{(2)}_{\mathsf{DX}}$, $s_{\mathsf{CX};2}$ $s_{\mathsf{CX};4}$, $s_{\mathsf{CX};8}$, $r^{(2)}_{\mathsf{DT}}$
s^2_{UC}, s^1_{PM2}, s^2_{RC}, s^2_{DX} $s_{\mathsf{CX};6}$, r^2_{UC}, $r^{(2)}_{\mathsf{DT}}$, r_{RC}	$Q^{[2]}_{\{rd_1\}}$ (Case 1)	$s_{\mathsf{CX};5}$, $s_{\mathsf{CX};7}$ $r^{[2]}_{\mathsf{DT}}$, r_{CX}
s^1_{PM2}, s^1_{RC}, $s^{(2)}_{\mathsf{DX}}$ $s_{\mathsf{CX};8}$, r_{RC}	$Q^{[2]}_{\{rd_1\}}$ (Case 2)	
$s_{\mathsf{CX};1}$, $s_{\mathsf{CX};2}$ $s_{\mathsf{CX};3}$, $s_{\mathsf{CX};4}$, r_{XT}	$Q^{[2]}_{\{rd_1\}}$ (Case 3)	
s^2_{UC}, s^2_{PM2}, s^1_{RC}, s^2_{RC} s^2_{DX}, $s^{(2)}_{\mathsf{DX}}$, $\{s_{\mathsf{CX};1}$ to $s_{\mathsf{CX};8}\}$ r^2_{UC}, $r^{(2)}_{\mathsf{DT}}$, $r^{[2]}_{\mathsf{DT}}$ r_{RC}, r_{XT}, r_{CX}	Q^2_{dec}	

Based on the properties of queues, we now describe the

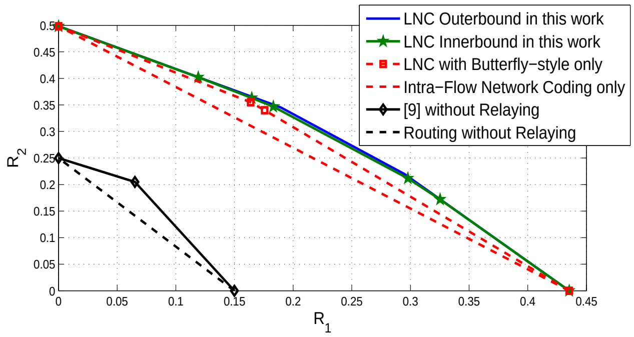


Fig. 4: Comparison of LNC regions with different achievable rates

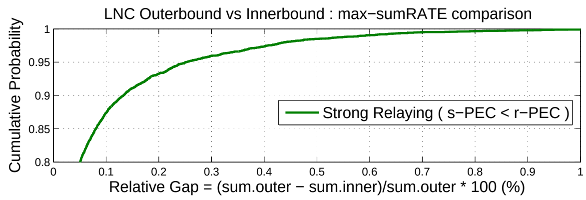


Fig. 5: The cumulative distribution of the relative gap between the outer and inner bounds.

correctness of Proposition 2, our LNC inner bound. To that end, we first investigate all the LNC operations involved in Proposition 2 and prove the "Queue Invariance", i.e., the queue properties explained above *remains invariant regardless of an LNC operation chosen*. Such long and tedious investigations are relegated to Appendix B. Then, the decodability condition (D), jointly with the Queue Invariance, imply that Q^1_{dec} and Q^2_{dec} will contain at least nR_1 and nR_2 number of pure session-1 and pure session-2 packets, respectively, in the end. This further means that, given a rate vector (R_1, R_2), any t-, s-, and r-variables that satisfy the inequalities (E) to (D) in Proposition 2 will be achievable. The correctness proof of Proposition 2 is thus complete.

For readability, we also describe for each queue, the associated LNC operations that moves packet into and takes packets out of, see Table II.

D. Numerical Evaluation

Consider a smart repeater network with marginal channel success probabilities: (a) s-PEC: $p_s(d_1) = 0.15$, $p_s(d_2) = 0.25$, and $p_s(r) = 0.8$; and (b) r-PEC: $p_r(d_1) = 0.75$ and $p_r(d_2) = 0.85$. And we assume that all the erasure events are independent. We will use the results in Propositions 1 and 2 to find the largest (R_1, R_2) value for this example scenario.

Fig. 4 compares the LNC capacity outer bound (Proposition 1) and the LNC inner bound (Proposition 2) with different achievability schemes. The smallest rate region is achieved by simply performing uncoded direct transmission without using the relay r. The second achievability scheme is the 2-receiver broadcast channel LNC from the source s in [9] while still not exploiting r at all. The third scheme performs the time-shared transmission between s and r, while allowing only intra-flow network coding. The forth scheme is derived from using only the classic butterfly-style LNCs corresponding to $s_{\mathsf{CX};l}\,(l{=}1,\cdots,8)$, r_{CX}, and r_{XT}. One can see that the result is strictly suboptimal. This shows that the newly-identified LNC operations other than the classic butterfly-style LNCs are critical in approaching the LNC capacity. The detailed rate region description of each sub-optimal achievability scheme can be found in Appendix C.

Fig. 5 examines the relative gaps between the outer and inner bounds by choosing the channel parameters $p_s(\cdot)$ and $p_r(\cdot)$ uniformly randomly while obeying the strong-relaying condition in Definition 2. For any chosen parameter instance, we use a linear programming solver to find the largest sum rate $(R_1 + R_2)$ of the LNC outer and inner bounds of Propositions 1 and 2, which are denoted by $R_{\mathsf{sum.outer}}$ and $R_{\mathsf{sum.inner}}$, respectively. We then compute the relative gap per each experiment, $(R_{\mathsf{sum.outer}} - R_{\mathsf{sum.inner}})/R_{\mathsf{sum.outer}}$, and then repeat the experiment 10000 times, and plot the cumulative distribution function (cdf) in unit of percentage. We can see that with more than 85% of the experiments, the relative gap between the outer and inner bound is smaller than 0.08%.

IV. Conclusion

This work studies the LNC capacity of the smart repeater packet erasure network for two unicast flows. The capacity region has been effectively characterized by the proposed linear-subspace-based outer bound, and the capacity-approaching LNC scheme with newly identified LNC operations other than the previously well-known classic butterfly-style operations.

References

[1] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5g of terrestrial mobile telecommunication," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 36–43, May 2014.
[2] T. M. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, September 1979.
[3] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3037–3063, September 2005.
[4] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
[5] S. H. Lim, Y.-H. Kim, A. E. Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Trans. Inform. Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.
[6] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 789–804, March 2006.
[7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM*, Pisa, Italy, September 2006, pp. 243–254.
[8] D. Koutsonikolas, C.-C. Wang, and Y. Hu, "Efficient network coding based opportunistic routing through cumulative coded acknowledgment," *IEEE/ACM Trans. Networking*, vol. 19, no. 5, pp. 1368–1381, 2011.
[9] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback - capacity and algorithms," in *Proc. 5th Workshop on Network Coding, Theory, and Applications (NetCod)*, Lausanne, Switzerland, June 2009, pp. 54–61.
[10] W.-C. Kuo and C.-C. Wang, "Two-flow capacity region of the cope principle for two-hop wireless erasure networks," *IEEE Trans. Inform. Theory*, vol. 59, no. 11, pp. 7553–7575, November 2013.

APPENDIX A
LIST OF CODING TYPES FOR FTs AND rFTs

We enumerate the 154 *Feasible Types* (FTs) defined in (12) that the source s can transmit in the following way:

$$\begin{aligned}\mathsf{FTs} \triangleq \{&\mathtt{00000},\ \mathtt{00010},\ \mathtt{00020},\ \mathtt{00030},\ \mathtt{00070},\ \mathtt{00110},\\ &\mathtt{00130},\ \mathtt{00170},\ \mathtt{00220},\ \mathtt{00230},\ \mathtt{00270},\ \mathtt{00330},\\ &\mathtt{00370},\ \mathtt{00570},\ \mathtt{00770},\ \mathtt{00A70},\ \mathtt{00B70},\ \mathtt{00F70},\\ &\mathtt{00F71},\ \mathtt{01010},\ \mathtt{01030},\ \mathtt{01070},\ \mathtt{01110},\ \mathtt{01130},\\ &\mathtt{01170},\ \mathtt{01230},\ \mathtt{01270},\ \mathtt{01330},\ \mathtt{01370},\ \mathtt{01570},\\ &\mathtt{01770},\ \mathtt{01A70},\ \mathtt{01B70},\ \mathtt{01F70},\ \mathtt{01F71},\ \mathtt{02020},\\ &\mathtt{02030},\ \mathtt{02070},\ \mathtt{02130},\ \mathtt{02170},\ \mathtt{02220},\ \mathtt{02230},\\ &\mathtt{02270},\ \mathtt{02330},\ \mathtt{02370},\ \mathtt{02570},\ \mathtt{02770},\ \mathtt{02A70},\\ &\mathtt{02B70},\ \mathtt{02F70},\ \mathtt{02F71},\ \mathtt{03030},\ \mathtt{03070},\ \mathtt{03130},\\ &\mathtt{03170},\ \mathtt{03230},\ \mathtt{03270},\ \mathtt{03330},\ \mathtt{03370},\ \mathtt{03570},\\ &\mathtt{03770},\ \mathtt{03A70},\ \mathtt{03B70},\ \mathtt{03F70},\ \mathtt{03F71},\ \mathtt{07070},\\ &\mathtt{07170},\ \mathtt{07270},\ \mathtt{07370},\ \mathtt{07570},\ \mathtt{07770},\ \mathtt{07A70},\\ &\mathtt{07B70},\ \mathtt{07F70},\ \mathtt{07F71},\ \mathtt{11110},\ \mathtt{11130},\ \mathtt{11170},\\ &\mathtt{11330},\ \mathtt{11370},\ \mathtt{11570},\ \mathtt{11770},\ \mathtt{11B70},\ \mathtt{11F70},\\ &\mathtt{11F71},\ \mathtt{13130},\ \mathtt{13170},\ \mathtt{13330},\ \mathtt{13370},\ \mathtt{13570},\\ &\mathtt{13770},\ \mathtt{13B70},\ \mathtt{13F70},\ \mathtt{13F71},\ \mathtt{17170},\ \mathtt{17370},\\ &\mathtt{17570},\ \mathtt{17770},\ \mathtt{17B70},\ \mathtt{17F70},\ \mathtt{17F71},\ \mathtt{22220},\\ &\mathtt{22230},\ \mathtt{22270},\ \mathtt{22330},\ \mathtt{22370},\ \mathtt{22770},\ \mathtt{22A70},\\ &\mathtt{22B70},\ \mathtt{22F70},\ \mathtt{22F71},\ \mathtt{23230},\ \mathtt{23270},\ \mathtt{23330},\\ &\mathtt{23370},\ \mathtt{23770},\ \mathtt{23A70},\ \mathtt{23B70},\ \mathtt{23F70},\ \mathtt{23F71},\\ &\mathtt{27270},\ \mathtt{27370},\ \mathtt{27770},\ \mathtt{27A70},\ \mathtt{27B70},\ \mathtt{27F70},\\ &\mathtt{27F71},\ \mathtt{33330},\ \mathtt{33370},\ \mathtt{33770},\ \mathtt{33B70},\ \mathtt{33F70},\\ &\mathtt{33F71},\ \mathtt{37370},\ \mathtt{37770},\ \mathtt{37B70},\ \mathtt{37F70},\ \mathtt{37F71},\\ &\mathtt{57570},\ \mathtt{57770},\ \mathtt{57F70},\ \mathtt{57F71},\ \mathtt{77770},\ \mathtt{77F70},\\ &\mathtt{77F71},\ \mathtt{A7A70},\ \mathtt{A7B70},\ \mathtt{A7F70},\ \mathtt{A7F71},\ \mathtt{B7B70},\\ &\mathtt{B7F70},\ \mathtt{B7F71},\ \mathtt{F7F70},\ \mathtt{F7F71}\},\end{aligned}$$

where each 5-digit index $\overline{\mathbf{b}}_1\overline{\mathbf{b}}_2\overline{\mathbf{b}}_3\overline{\mathbf{b}}_4\overline{\mathbf{b}}_5$ represent a 15-bitstring $\mathbf{b}$ of which $\overline{\mathbf{b}}_1$ is a hexadecimal of first four bits, $\overline{\mathbf{b}}_2$ is a octal of the next three bits, $\overline{\mathbf{b}}_3$ is a hexadecimal of the next four bits, $\overline{\mathbf{b}}_4$ is a octal of the next three bits, and $\overline{\mathbf{b}}_5$ is binary of the last bit. The subset of FTs that the relay r can transmit, i.e., rFTs are listed separately in the following:

$$\begin{aligned}r\mathsf{FTs} \triangleq \{&\mathtt{00F71},\ \mathtt{01F71},\ \mathtt{02F71},\ \mathtt{03F71},\ \mathtt{07F71},\ \mathtt{11F71},\\ &\mathtt{13F71},\ \mathtt{17F71},\ \mathtt{22F71},\ \mathtt{23F71},\ \mathtt{27F71},\ \mathtt{33F71},\\ &\mathtt{37F71},\ \mathtt{57F71},\ \mathtt{77F71},\ \mathtt{A7F71},\ \mathtt{B7F71},\ \mathtt{F7F71}\},\end{aligned}$$

Recall that the b_{15} of a 15-bitstring $\mathbf{b}$ represents whether the coding subset belongs to $A_{15}(t)$ or not, and $A_{15}(t) \triangleq S_r(t-1)$ by definition (11). As a result, any coding type with $b_{15} = 1$ implies that it lies in the knowledge space of the relay r. The enumerated rFTs in the above is thus a collection of such coding subsets in FTs with $\overline{\mathbf{b}}_5 = 1$.

APPENDIX B
LNC ENCODING OPERATIONS, PACKET MOVEMENT PROCESS, AND QUEUE INVARIANCE IN PROPOSITION 2

In the following, we will describe all the LNC encoding operations and the corresponding packet movement process of Proposition 2 one by one, and then prove that the Queue Invariance explained in Section III-C always holds.

To simplify the analysis, we will ignore the null reception, i.e., none of $\{d_1, d_2, r\}$ receives a transmitted packet, because nothing will happen in the queueing network. Moreover, we exploit the following symmetry: For those variables whose superscript indicates the session information $k \in \{1, 2\}$ (either session-1 or session-2), here we describe session-1 ($k = 1$) only. Those variables with $k = 2$ in the superscript will be symmetrically explained by simultaneously swapping (a) session-1 and session-2 in the superscript; (b) X and Y; (c) i and j; and (d) d_1 and d_2, if applicable.

• s^1_{UC}: The source s transmits $X_i \in Q^1_\phi$. Depending on the reception status, the packet movement process following the inequalities in Proposition 2 is summarized as follows.

Departure	**Reception Status**	**Insertion**
$Q^1_\phi \xrightarrow{X_i}$	$\overline{d_1 d_2} r$	$\xrightarrow{X_i} Q^1_{\{r\}}$
	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\{d_2\}}$
	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
	$d_1 d_2 r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$

- **Departure**: One property for $X_i \in Q^1_\phi$ is that X_i must be unknown to any of $\{d_1, d_2, r\}$. As a result, whenever X_i is received by any of them, X_i must be removed from Q^1_ϕ for the Queue Invariance.
- **Insertion**: One can easily verify that the queue properties for $Q^1_{\{r\}}$, $Q^1_{\{d_2\}}$, Q^1_{dec}, and $Q^{[1]}_{\{rd_2\}}$ hold for the corresponding insertions.

• s^2_{UC}: s transmits $Y_j \in Q^2_\phi$. The movement process is symmetric to s^1_{UC}.

• s^1_{PM1}: s transmits a mixture $[X_i + Y_j]$ from $X_i \in Q^1_\phi$ and $Y_j \in Q^2_{\{r\}}$. The movement process is as follows.

$Q^1_\phi \xrightarrow{X_i}$	$\overline{d_1 d_2} r$	$\xrightarrow{X_i} Q^1_{\{r\}}$
$Q^1_\phi \xrightarrow{X_i}, Q^2_{\{r\}} \xrightarrow{Y_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{[X_i+Y_j]} Q^{m\|2}_{\{d_2\}\|\{r\}}$
	$d_1 \overline{d_2 r}$	$\xrightarrow{[X_i+Y_j]:Y_j} Q_{\mathsf{mix}}$
	$\overline{d_1} d_2 r$	$\xrightarrow{[X_i+Y_j]:X_i} Q_{\mathsf{mix}}$
	$d_1 \overline{d_2} r$ $d_1 d_2 \overline{r}$	$\xrightarrow{[X_i+Y_j]:Y_j} Q_{\mathsf{mix}}$
	$d_1 d_2 r$	$\xrightarrow{[X_i+Y_j]:\text{ either } X_i \text{ or } Y_j} Q_{\mathsf{mix}}$

- **Departure**: The property for $X_i \in Q^1_\phi$ is that X_i must be unknown to any of $\{d_1, d_2, r\}$, even not flagged in $\mathsf{RL}_{\{d_1,d_2,r\}}$. As a result, whenever the mixture $[X_i + Y_j]$ is received by any of $\{d_1, d_2, r\}$, X_i must be removed from Q^1_ϕ. Similarly, the property for $Y_j \in Q^2_{\{r\}}$ is that Y_j

must be unknown to any of $\{d_1, d_2\}$, even not flagged in $\mathsf{RL}_{\{d_1,d_2\}}$. Therefore, whenever the mixture is received by any of $\{d_1, d_2\}$, Y_j must be removed from $Q^2_{\{r\}}$.

- **Insertion**: When only r receives the mixture, r can use the known Y_j and the received $[X_i + Y_j]$ to extract the pure X_i. As a result, we can insert X_i to $Q^1_{\{r\}}$ as it is not flagged in $\mathsf{RL}_{\{d_1,d_2\}}$. The case when only d_2 receives the mixture satisfies the properties of $Q^{m|2}_{\{d_2\}|\{r\}}$ as r knows the pure Y_j only while d_2 knows the mixture $[X_i+Y_j]$ only. As a result, we can insert $[X_i + Y_j]$ to $Q^{m|2}_{\{d_2\}|\{r\}}$. The remaining reception cases fall into at least one of two conditions of Q_{mix}. For example when only d_1 receives the mixture, now $[X_i + Y_j]$ is in $\mathsf{RL}_{\{d_1\}}$ while Y_j is still known by r only. This corresponds to the first condition of Q_{mix}. One can easily verify that other cases satisfy either one of or both properties of Q_{mix}. Following the packet format for Q_{mix}, we insert $[X_i + Y_j] : W$ into Q_{mix} where W denotes the packet in r that can benefit both destinations when transmitted. From the previous example when only d_1 receives the mixture, we insert $[X_i+Y_j] : Y_j$ into Q_{mix} as sending the known Y_j from r simultaneously enables d_2 to receive the desired Y_j and d_1 to decode the desired X_i by subtracting Y_j from the received $[X_i + Y_j]$.

- s^2_{PM1}: s transmits a mixture $[X_i+Y_j]$ from $X_i \in Q^1_{\{r\}}$ and $Y_j \in Q^2_{\phi}$. The movement process is symmetric to s^1_{PM1}.
- s^1_{PM2}: s transmits a mixture $[X_i + Y_j]$ from $X_i \in Q^1_{\{r\}}$ and $Y_j \in Q^2_{\{d_1\}}$. The movement process is as follows.

$Q^2_{\{d_1\}}\xrightarrow{Y_j}$	$\overline{d_1 d_2} r$	$\xrightarrow[\text{Case 1}]{Y_j} Q^{[2]}_{\{rd_1\}}$
$Q^1_{\{r\}}\xrightarrow{X_i}$, $Q^2_{\{d_1\}}\xrightarrow{Y_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{[X_i+Y_j]:X_i} Q_{\mathsf{mix}}$
$Q^1_{\{r\}}\xrightarrow{X_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
$Q^1_{\{r\}}\xrightarrow{X_i}$, $Q^2_{\{d_1\}}\xrightarrow{Y_j}$	$\overline{d_1} d_2 r$	$\xrightarrow{[X_i+Y_j]:X_i} Q_{\mathsf{mix}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 1}]{Y_j} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 2}]{X_i(\equiv Y_j)} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 1}]{Y_j} Q^{[2]}_{\{rd_1\}}$

- **Departure**: The property for $X_i \in Q^1_{\{r\}}$ is that X_i must be unknown to any of $\{d_1, d_2\}$, even not flagged in $\mathsf{RL}_{\{d_1,d_2\}}$. As a result, whenever the mixture $[X_i+Y_j]$ is received by any of $\{d_1, d_2\}$, X_i must be removed from $Q^1_{\{r\}}$. Similarly, the property for $Y_j \in Q^2_{\{d_1\}}$ is that Y_j must be unknown to any of $\{d_2, r\}$, even not flagged in $\mathsf{RL}_{\{d_2,r\}}$. Therefore, whenever the mixture is received by any of $\{d_2, r\}$, Y_j must be removed from $Q^2_{\{d_1\}}$.
- **Insertion**: Whenever d_1 receives the mixture, d_1 can use the known Y_j and the received $[X_i + Y_j]$ to extract the pure/desired X_i. As a result, we can insert X_i into Q^1_{dec} whenever d_1 receives. The cases when d_2 receives but d_1 does not fall into the second condition of Q_{mix} as $[X_i+Y_j]$ is in $\mathsf{RL}_{\{d_2\}}$ and X_i is known by r only. Namely, r can benefit both destinations simultaneously by sending the known X_i. For those two reception status $\overline{d_1}d_2\overline{r}$ and $\overline{d_1}d_2r$, we can thus insert this mixture into Q_{mix} as $[X_i+Y_j]:X_i$. Whenever r receives the mixture, r can use the known X_i and the received $[X_i + Y_j]$ to extract the pure Y_j. Now Y_j is known by both r and d_1 but still unknown to d_2 even if d_2 receives this mixture $[X_i + Y_j]$ as well. As a result, Y_j can be moved to $Q^{[2]}_{\{rd_1\}}$ as the Case 1 insertion. But for the reception status of $\overline{d_1}d_2r$, note from the previous discussion that we can insert the mixture into Q_{mix} since d_2 receives the mixture but d_1 does not. In this case, we chose to use more efficient Q_{mix} that can handle both sessions simultaneously. Finally when the reception status is $d_1d_2\overline{r}$, we have that X_i is known by both r and d_1 while the mixture $[X_i+Y_j]$ is received by d_2. Namely, X_i is still unknown to d_2 but when it is delivered, d_2 can use X_i and the received $[X_i + Y_j]$ to extract a desired session-2 packet Y_j. Moreover, X_i is already in Q^1_{dec} and thus can be used as an information-equivalent packet for Y_j. This scenario is exactly the same as the Case 2 of $Q^{[2]}_{\{rd_1\}}$ and thus we can move X_i into $Q^{[2]}_{\{rd_1\}}$ as the Case 2 insertion.

- s^2_{PM2}: s transmits a mixture $[X_i + Y_j]$ from $X_i \in Q^1_{\{d_2\}}$ and $Y_j \in Q^2_{\{r\}}$. The movement process is symmetric to s^1_{PM2}.
- s^1_{RC}: s transmits X_i of the mixture $[X_i + Y_j]$ in $Q^{m|2}_{\{d_2\}|\{r\}}$. The movement process is as follows.

$Q^{m\|2}_{\{d_2\}\|\{r\}} \xrightarrow{[X_i+Y_j]}$	$\overline{d_1 d_2} r$	$\xrightarrow{[X_i+Y_j]:X_i} Q_{\mathsf{mix}}$
	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\{d_2\}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow{X_i} Q^{(2)\|2}_{\{d_1\}\|\{r\}}$
	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 2}]{X_i(\equiv Y_j)} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 \overline{r}$ $d_1 d_2 r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$

- **Departure**: One condition for $[X_i + Y_j] \in Q^{m|2}_{\{d_2\}|\{r\}}$ is that X_i is unknown to any of $\{d_1, d_2, r\}$. As a result, whenever X_i is received by any of $\{d_1, d_2, r\}$, the mixture $[X_i + Y_j]$ must be removed from $Q^{m|2}_{\{d_2\}|\{r\}}$.
- **Insertion**: From the conditions of $Q^{m|2}_{\{d_2\}|\{r\}}$, we know that X_i is unknown to d_1 and Y_j is known only by r. As a result, whenever d_1 receives X_i, d_1 receives the new session-1 packet and thus we can insert X_i into Q^1_{dec}. Whenever d_2 receives X_i, d_2 can use the known $[X_i + Y_j]$ and the received X_i to subtract the pure Y_j. We can thus insert Y_j into Q^2_{dec}. The case when only r receives X_i falls into the first condition of Q_{mix} as $[X_i + Y_j]$ is in $\mathsf{RL}_{\{d_2\}}$ and X_i is known by r only. In this case, r can benefit both destinations simultaneously by sending the received X_i. For this reception status of $\overline{d_1d_2}r$, we thus insert the mixture into Q_{mix} as $[X_i + Y_j] : X_i$. The remaining reception status to consider are $\overline{d_1}d_2\overline{r}$, $d_1\overline{d_2r}$, $\overline{d_1}d_2r$, and $d_1\overline{d_2}r$. The first when only d_2 receives X_i falls into the property of $Q^1_{\{d_2\}}$ as X_i is known only by d_2 and not flagged in $\mathsf{RL}_{\{d_1,r\}}$. Thus we can insert X_i into $Q^1_{\{d_2\}}$. Obviously, d_2 can decode Y_j from the previous discussion. For the second when only d_1 receives X_i, we first have $X_i \in Q^1_{\mathsf{dec}}$ while X_i is unknown to any of $\{d_2, r\}$. Moreover, Y_j is known by r only and $[X_i+Y_j]$ is in $\mathsf{RL}_{\{d_2\}}$. This scenario falls exactly into $Q^2_{\{d_1\}}$ and thus

we can insert X_i into $Q^2_{\{d_1\}}$. The third case when both d_2 and r receive X_i falls exactly into Case 1 of $Q^{[1]}_{\{rd_2\}}$ as X_i is now known by both d_2 and r but still unknown to d_1. And obviously, d_2 can decode Y_j from the previous discussion. For the fourth case when both d_1 and r receive X_i, we now have that r contains $\{X_i, Y_j\}$; d_1 contains X_i; and d_2 contains $[X_i + Y_j]$. That is, X_i is already in Q^1_{dec} and known by r as well but still unknown to d_2. Moreover, d_2 can decode the desired session-2 packet Y_j when it receives X_i further. As a result, X_i can be used as an information-equivalent packet for Y_j and can be moved into $Q^{[2]}_{\{rd_1\}}$ as the Case 2 insertion.

• s^2_{RC}: s transmits Y_j of $[X_i+Y_j] \in Q^{m|1}_{\{d_1\}|\{r\}}$. The movement process is symmetric to s^1_{RC}.

• s^1_{DX}: s transmits $X_i \in Q^1_{\{d_2\}}$. The movement process is as follows.

$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$\overline{d_1 d_2} r$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$
do nothing	$\overline{d_1} d_2 \overline{r}$	do nothing
$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
	$d_1 d_2 \overline{r}$	
	$d_1 d_2 r$	

- **Departure**: One condition for $X_i \in Q^1_{\{d_2\}}$ is that X_i must be unknown to any of $\{d_1, r\}$. As a result, X_i must be removed from $Q^1_{\{d_2\}}$ whenever it is received by any of $\{d_1, r\}$.
- **Insertion**: Whenever d_1 receives X_i, it receives a new session-1 packet and thus we can insert X_i into Q^1_{dec}. If X_i is received by r but not by d_1, then X_i will be known by both d_2 and r (since d_2 already knows X_i) but still unknown to d_1. This falls exactly into the first-case scenario of $Q^{[1]}_{\{rd_2\}}$ and thus we can move X_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 1 insertion.

• s^2_{DX}: s transmits $Y_j \in Q^2_{\{d_1\}}$. The movement process is symmetric to s^1_{DX}.

• $s^{(1)}_{\mathsf{DX}}$: s transmits $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$. The movement process is as follows.

$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$	$\overline{d_1 d_2} r$	$\xrightarrow[\text{Case 2}]{Y_i} Q^{[1]}_{\{rd_2\}}$
do nothing	$\overline{d_1} d_2 \overline{r}$	do nothing
$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i (\equiv Y_i)} Q^1_{\mathsf{dec}}$
	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 2}]{Y_i} Q^{[1]}_{\{rd_2\}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i (\equiv Y_i)} Q^1_{\mathsf{dec}}$
	$d_1 d_2 \overline{r}$	
	$d_1 d_2 r$	

- **Departure**: One property for $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ is that Y_i must be unknown to any of $\{d_1, r\}$. As a result, whenever Y_i is received by any of $\{d_1, r\}$, Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$.
- **Insertion**: From the property of $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$, we know that $Y_i \in Q^2_{\mathsf{dec}}$; there exists a session-1 packet X_i still unknown to d_1 where $X_i \equiv Y_i$; and $[X_i + Y_i]$ is in $\mathsf{RL}_{\{d_1\}}$. As a result, whenever d_1 receives Y_i, d_1 can use the received Y_i and the known $[X_i + Y_i]$ to extract X_i and thus we can insert X_i into Q^1_{dec}. If Y_i is received by r but not by d_1, then Y_i will be known by both d_2 and r but unknown to d_1, where $[X_i + Y_i]$ is in $\mathsf{RL}_{\{d_1\}}$. Thus when d_1 receives Y_i, d_1 can further decode the desired X_i. Moreover, Y_i is already in Q^2_{dec}. As a result, we can move Y_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 2 insertion.

• $s^{(2)}_{\mathsf{DX}}$: s transmits $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$. The movement process is symmetric to $s^{(1)}_{\mathsf{DX}}$.

• $s_{\mathsf{CX};1}$: s transmits $[X_i+Y_j]$ from $X_i \in Q^1_{\{d_2\}}$ and $Y_j \in Q^2_{\{d_1\}}$. The movement process is as follows.

$Q^1_{\{d_2\}} \xrightarrow{X_i}$, $Q^2_{\{d_1\}} \xrightarrow{Y_j}$	$\overline{d_1 d_2} r$	$\xrightarrow{[X_i+Y_j]} Q^{m_{\mathsf{CX}}}_{\{r\}}$
$Q^2_{\{d_1\}} \xrightarrow{Y_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$, $Q^2_{\{d_1\}} \xrightarrow{Y_j}$	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 3}]{[X_i+Y_j]} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 3}]{[X_i+Y_j]} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1 d_2 r$	

- **Departure**: One condition for $X_i \in Q^1_{\{d_2\}}$ is that X_i must be unknown to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{d_1,r\}}$. As a result, whenever the mixture is received by any of $\{d_1, r\}$, X_i must be removed from $Q^1_{\{d_2\}}$. Symmetrically for $Y_j \in Q^2_{\{d_1\}}$, whenever the mixture is received by any of $\{d_2, r\}$, Y_j must be removed from $Q^2_{\{d_1\}}$.
- **Insertion**: Whenever d_1 receives the mixture $[X_i+Y_j]$, d_1 can use the known $Y_j \in Q^2_{\{d_1\}}$ and the received $[X_i + Y_j]$ to extract the desired X_i and thus we can insert X_i into Q^1_{dec}. Similarly, whenever d_2 receives this mixture, d_2 can use the known $X_i \in Q^1_{\{d_2\}}$ and the received $[X_i + Y_j]$ to extract the desired Y_j and thus we can insert Y_j into Q^2_{dec}. The remaining reception status are $\overline{d_1 d_2} r$, $\overline{d_1} d_2 r$, and $d_2 \overline{d_2} r$. The first when only r receives the mixture exactly falls into the first-case scenario of $Q^{m_{\mathsf{CX}}}_{\{r\}}$ as $[X_i + Y_j]$ is in $\mathsf{RL}_{\{r\}}$; $X_i \in Q^1_{\{d_2\}}$ is known by d_2 only; and $Y_j \in Q^2_{\{d_1\}}$ is known by d_1 only. As a result, r can then send this mixture $[X_i+Y_j]$ to benefit both destinations. The second case when both d_2 and r receive the mixture, jointly with the assumption $Y_j \in Q^2_{\{d_1\}}$, falls exactly into the third-case scenario of $Q^{[1]}_{\{rd_2\}}$ where W_i is a pure session-1 packet. As a result, we can move $[X_i + Y_j]$ into $Q^{[1]}_{\{rd_2\}}$ as the Case 3 insertion. (And obviously, d_2 can decode Y_j from the previous discussion.) The third case when both d_1 and r receive the mixture follows symmetrically to the second case of $\overline{d_1} d_2 r$ and thus we can insert $[X_i + Y_j]$ into $Q^{[2]}_{\{rd_1\}}$ as the Case 3 insertion.

• $s_{\mathsf{CX};2}$: s transmits $[X_i + X_j]$ from $X_i \in Q^1_{\{d_2\}}$ and $X_j \in$

$Q^{(2)|2}_{\{d_1\}|\{r\}}$. The movement process is as follows.

$Q^1_{\{d_2\}} \xrightarrow{X_i}$, $Q^{(2)|2}_{\{d_1\}|\{r\}} \xrightarrow{X_j}$	$\overline{d_1 d_2} r$	$\xrightarrow{[X_i+X_j]} Q^{m_{\mathsf{CX}}}_{\{r\}}$
$Q^{(2)|2}_{\{d_1\}|\{r\}} \xrightarrow{X_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{Y_j(\equiv X_j)} Q^2_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$, $Q^{(2)|2}_{\{d_1\}|\{r\}} \xrightarrow{X_j}$	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 3}]{[X_i+X_j]} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j(\equiv X_j)} Q^2_{\mathsf{dec}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 3}]{[X_i+X_j]} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j(\equiv X_j)} Q^2_{\mathsf{dec}}$
	$d_1 d_2 r$	

- **Departure**: One condition for $X_i \in Q^1_{\{d_2\}}$ is that X_i must be unknown to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{d_1,r\}}$. As a result, whenever the mixture $[X_i + X_j]$ is received by any of $\{d_1, r\}$, X_i must be removed from $Q^1_{\{d_2\}}$. From the property for $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$, we know that X_j is unknown to any of $\{d_2, r\}$, even not flagged in $\mathsf{RL}_{\{r\}}$. As a result, whenever r receives the mixture $[X_i + X_j]$, X_j must be removed from $Q^{(2)|2}_{\{d_1\}|\{r\}}$. Moreover, whenever d_2 receives this mixture, d_2 can use the known $X_i \in Q^1_{\{d_2\}}$ and the received $[X_i + X_j]$ to decode X_j and thus X_j must be removed from $Q^{(2)|2}_{\{d_1\}|\{r\}}$.
- **Insertion**: From the properties of $X_i \in Q^1_{\{d_2\}}$ and $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$, we know that r contains Y_j (still unknown to d_2 and $Y_j \equiv X_j$); d_1 contains X_j; and d_2 contains $\{X_i, [Y_j + X_j]\}$ already. Therefore, whenever d_1 receives the mixture $[X_i + X_j]$, d_1 can use the known X_j and the received $[X_i + X_j]$ to extract the desired X_i and thus we can insert X_i into Q^1_{dec}. Similarly, whenever d_2 receives this mixture, d_2 can use the known $\{X_i, [Y_j + X_j]\}$ and the received $[X_i + X_j]$ to extract the desired Y_j, and thus we can insert Y_j into Q^2_{dec}. The remaining reception status are $\overline{d_1 d_2} r$, $\overline{d_1} d_2 r$, and $d_2 \overline{d_2} r$. One can see that the case when only r receives the mixture exactly falls into the Case 2 scenario of $Q^{m_{\mathsf{CX}}}_{\{r\}}$. For the second case when both d_2 and r receive the mixture, now r contains $\{Y_j, [X_i + X_j]\}$; d_1 contained X_j before; and d_2 contains $\{X_i, [Y_j + X_j], [X_i + X_j]\}$. This falls exactly into the third-case scenario of $Q^{[1]}_{\{rd_2\}}$ where W_i is a pure session-1 packet X_i. As a result, we can move $[X_i + X_j]$ into $Q^{[1]}_{\{rd_2\}}$ as the Case 3 insertion. (And obviously, d_2 can decode the desired Y_j from the previous discussion.) For the third case when both d_1 and r receive the mixture, now r contains $\{Y_j, [X_i + X_j]\}$; d_1 contains $\{X_j, [X_i + X_j]\}$; and d_2 contained $\{X_i, [Y_j + X_j]\}$ before, where we now have $X_i \in Q^1_{\mathsf{dec}}$ from the previous discussion. This falls exactly into the third-case scenario of $Q^{[2]}_{\{rd_1\}}$ where W_j is a pure session-1 packet $X_j \in Q^1_{\mathsf{dec}}$. Note that delivering $[X_i + X_j]$ will enable d_2 to further decode the desired Y_j. Thus we can move $[X_i + X_j]$ into $Q^{[2]}_{\{rd_1\}}$ as the Case 3 insertion.

- $s_{\mathsf{CX};3}$: s transmits $[Y_i + Y_j]$ from $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and $Y_j \in Q^2_{\{d_1\}}$. The movement process is as follows.

$Q^{(1)|1}_{\{d_2\}|\{r\}} \xrightarrow{Y_i}$, $Q^2_{\{d_1\}} \xrightarrow{Y_j}$	$\overline{d_1 d_2} r$	$\xrightarrow{[Y_i+Y_j]} Q^{m_{\mathsf{CX}}}_{\{r\}}$
$Q^2_{\{d_1\}} \xrightarrow{Y_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
$Q^{(1)|1}_{\{d_2\}|\{r\}} \xrightarrow{Y_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i(\equiv Y_i)} Q^1_{\mathsf{dec}}$
$Q^{(1)|1}_{\{d_2\}|\{r\}} \xrightarrow{Y_i}$, $Q^2_{\{d_1\}} \xrightarrow{Y_j}$	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 3}]{[Y_i+Y_j]} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i(\equiv Y_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 3}]{[Y_i+Y_j]} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i(\equiv Y_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1 d_2 r$	

- **Departure**: From the property for $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$, we know that Y_i is unknown to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{r\}}$. As a result, whenever r receives the mixture $[Y_i + Y_j]$, Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$. Moreover, whenever d_1 receives this mixture, d_1 can use the known $Y_j \in Q^2_{\{d_1\}}$ and the received $[Y_i + Y_j]$ to decode Y_i and thus Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$. One condition for $Y_j \in Q^2_{\{d_1\}}$ is that Y_j must be unknown to any of $\{d_2, r\}$, even not flagged in $\mathsf{RL}_{\{d_2,r\}}$. As a result, whenever the mixture $[Y_i + Y_j]$ is received by any of $\{d_2, r\}$, Y_j must be removed from $Q^2_{\{d_1\}}$.
- **Insertion**: From the properties of $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and $Y_j \in Q^2_{\{d_1\}}$, we know that r contains X_i (still unknown to d_1 and $X_i \equiv Y_i$); d_1 contains $\{Y_j, [X_i + Y_i]\}$; and d_2 contains Y_i already. Therefore, whenever d_1 receives the mixture $[Y_i + Y_j]$, d_1 can use the known $\{Y_j, [X_i + Y_i]\}$ and the received $[Y_i + Y_j]$ to extract the desired X_i and thus we can insert X_i into Q^1_{dec}. Similarly, whenever d_2 receives this mixture, d_2 can use the known Y_i and the received $[Y_i + Y_j]$ to extract the desired Y_j, and thus we can insert Y_j into Q^2_{dec}. The remaining reception status are $\overline{d_1 d_2} r$, $\overline{d_1} d_2 r$, and $d_2 \overline{d_2} r$. One can see that the first case when only r receives the mixture exactly falls into the Case 3 scenario of $Q^{m_{\mathsf{CX}}}_{\{r\}}$. For the second case when both d_2 and r receive the mixture, now r contains $\{X_i, [Y_i + Y_j]\}$; d_1 contained $\{Y_j, [X_i + Y_i]\}$ before; and d_2 contains $\{Y_i, [Y_i + Y_j]\}$, where we now have $Y_j \in Q^2_{\mathsf{dec}}$ from the previous discussion. This falls exactly into the third-case scenario of $Q^{[1]}_{\{rd_2\}}$ where W_i is a pure session-2 packet Y_i. Note that delivering $[Y_i + Y_j]$ will enable d_1 to further decode the desired X_i. Thus we can move $[Y_i + Y_j]$ into $Q^{[1]}_{\{rd_2\}}$ as the Case 3 insertion. For the third case when both d_1 and r receive the mixture, now r contains $\{X_i, [Y_i + Y_j]\}$; d_1 contains $\{Y_j, [X_i + Y_i], [Y_i + Y_j]\}$; and d_2 contained Y_i before. This falls exactly into the third-case scenario of $Q^{[2]}_{\{rd_1\}}$ where W_j is a pure session-2 packet Y_j. As a result, we can move $[Y_i + Y_j]$ into $Q^{[2]}_{\{rd_1\}}$ as the Case 3 insertion. (And obviously, d_1 can decode the desired X_i from the previous discussion.)

- $s_{\mathsf{CX};4}$: s transmits $[Y_i + X_j]$ from $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and $X_j \in$

$Q^{(2)|2}_{\{d_1\}|\{r\}}$. The movement process is as follows.

$Q^{(1)\mid 1}_{\{d_2\}\mid\{r\}} \xrightarrow{Y_i}$, $Q^{(2)\mid 2}_{\{d_1\}\mid\{r\}} \xrightarrow{X_j}$	$\overline{d_1 d_2} r$	$\xrightarrow{[Y_i+X_j]} Q^{m_{\mathsf{CX}}}_{\{r\}}$
$Q^{(2)\mid 2}_{\{d_1\}\mid\{r\}} \xrightarrow{X_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{Y_j(\equiv X_j)} Q^2_{\mathsf{dec}}$
$Q^{(1)\mid 1}_{\{d_2\}\mid\{r\}} \xrightarrow{Y_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i(\equiv Y_i)} Q^1_{\mathsf{dec}}$
$Q^{(1)\mid 1}_{\{d_2\}\mid\{r\}} \xrightarrow{Y_i}$, $Q^{(2)\mid 2}_{\{d_1\}\mid\{r\}} \xrightarrow{X_j}$	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 3}]{[Y_i+X_j]} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j(\equiv X_j)} Q^2_{\mathsf{dec}}$
	$d_1 \overline{d_2} r$	$\xrightarrow{X_i(\equiv Y_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 3}]{[Y_i+X_j]} Q^{[2]}_{\{rd_1\}}$
	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i(\equiv Y_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j(\equiv X_j)} Q^2_{\mathsf{dec}}$
	$d_1 d_2 r$	

- **Departure**: From the property for $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$, we know that Y_i is unknown to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{r\}}$. As a result, whenever r receives the mixture $[Y_i + X_j]$, Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$. Moreover, $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$ is known by d_1. As a result, whenever d_1 receives the mixture, d_1 can use the known X_j and the received $[Y_i + X_j]$ to decode Y_i and thus Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$. Symmetrically for $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$, whenever the mixture is received by any of $\{d_2, r\}$, X_j must be removed from $Q^{(2)|2}_{\{d_1\}|\{r\}}$.
- **Insertion**: From the properties of $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$, we know that r contains $\{X_i, Y_j\}$ where X_i (resp. Y_j) is still unknown to d_1 (resp. d_2) and $X_i \equiv Y_i$ (resp. $Y_j \equiv X_j$); d_1 contains $\{[X_i + Y_i], X_j\}$; and d_2 contains $\{Y_i, [Y_j + X_j]\}$ already. Therefore, whenever d_1 receives the mixture $[Y_i + X_j]$, d_1 can use the known $\{[X_i + Y_i], X_j\}$ and the received $[Y_i + X_j]$ to extract the desired X_i and thus we can insert X_i into Q^1_{dec}. Similarly, whenever d_2 receives this mixture, d_2 can use the known $\{Y_i, [Y_j + X_j]\}$ and the received $[Y_i + X_j]$ to extract the desired Y_j, and thus we can insert Y_j into Q^2_{dec}. The remaining reception status are $\overline{d_1 d_2} r$, $\overline{d_1} d_2 r$, and $d_2 \overline{d_2} r$. One can see that the first case when only r receives the mixture exactly falls into the Case 4 scenario of $Q^{m_{\mathsf{CX}}}_{\{r\}}$. For the second case when both d_2 and r receive the mixture, now r contains $\{X_i, Y_j, [Y_i + X_j]\}$; d_1 contained $\{[X_i + Y_i], X_j\}$ before; and d_2 contains $\{Y_i, [Y_j + X_j], [Y_i + X_j]\}$ where we now have $X_j \in Q^1_{\mathsf{dec}}$ from the previous discussion. This falls exactly into the third-case scenario of $Q^{[1]}_{\{rd_2\}}$ where W_i is a pure session-2 packet Y_i. Note that delivering $[Y_i + X_j]$ will enable d_1 to further decode the desired X_i. Thus we can move $[Y_i + X_j]$ into $Q^{[1]}_{\{rd_2\}}$ as the Case 3 insertion. For the third case when both d_1 and r receive the mixture, now r contains $\{X_i, Y_j, [Y_i + X_j]\}$; d_1 contains $\{[X_i + Y_i], X_j, [Y_i + X_j]\}$; and d_2 contained $\{Y_i, [Y_j + X_j]\}$ before, where we now have $Y_i \in Q^2_{\mathsf{dec}}$ from the previous discussion. This falls exactly into the third-case scenario of $Q^{[2]}_{\{rd_1\}}$ where W_j is a pure session-2 packet X_j. Note that delivering $[Y_i + X_j]$ will enable d_2 to further decode the desired Y_j. Thus we can move $[Y_i + X_j]$ into $Q^{[2]}_{\{rd_1\}}$ as the Case 3 insertion.

- $s_{\mathsf{CX};5}$: s transmits $[X_i + \overline{W}_j]$ from $X_i \in Q^1_{\{d_2\}}$ and $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$. The movement process is as follows.

$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$\overline{d_1 d_2} r$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$
$Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{Y_j(\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$, $Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j(\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$	$d_1 \overline{d_2} r$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$
$Q^1_{\{d_2\}} \xrightarrow{X_i}$, $Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j(\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
	$d_1 d_2 r$	

- **Departure**: The property for $X_i \in Q^1_{\{d_2\}}$ is that X_i must be unknown to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{d_1,r\}}$. As a result, whenever the mixture $[X_i + \overline{W}_j]$ is received by any of $\{d_1, r\}$, X_i must be removed from $Q^1_{\{d_2\}}$. Similarly, one condition for $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$ is that $\overline{W}_j$ must be unknown to d_2. However when d_2 receives the mixture, d_2 can use the known $X_i \in Q^1_{\{d_2\}}$ and the received $[X_i + \overline{W}_j]$ to decode $\overline{W}_j$. Thus $\overline{W}_j$ must be removed from $Q^{[2]}_{\{rd_1\}}$ whenever d_2 receives.
- **Insertion**: From the properties of $X_i \in Q^1_{\{d_2\}}$ and $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$, we know that r contains $\overline{W}_j$; d_1 contains $\overline{W}_j$; and d_2 contains X_i already. Therefore, whenever d_1 receives this mixture, d_1 can use the known $\overline{W}_j$ and the received $[X_i + \overline{W}_j]$ to extract the desired X_i and thus we can insert X_i into Q^1_{dec}. Similarly, whenever d_2 receives this mixture, d_2 can use the known X_i and the received $[X_i + \overline{W}_j]$ to extract $\overline{W}_j$. We now need to consider case by case when $\overline{W}_j$ was inserted into $Q^{[2]}_{\{rd_1\}}$. If it was the Case 1 insertion, then $\overline{W}_j$ is a pure session-2 packet Y_j and thus we can simply insert Y_j into Q^2_{dec}. If it was the Case 2 insertion, then $\overline{W}_j$ is a pure session-2 packet $X_j \in Q^1_{\mathsf{dec}}$ and there exists a session-2 packet Y_j still unknown to d_2 where $Y_j \equiv X_j$. Moreover, d_2 has received $[Y_j + X_j]$. As a result, d_2 can further decode Y_j and thus we can insert Y_j into Q^2_{dec}. If it was the Case 3 insertion, then $\overline{W}_j$ is a mixed form of $[W_i + W_j]$ where W_i is already known by d_2 but W_j is not. As a result, d_2 can decode W_j upon receiving $\overline{W}_j = [W_i + W_j]$. Note that W_j in the Case 3 insertion $\overline{W}_j = [W_i + W_j] \in Q^{[2]}_{\{rd_1\}}$ comes from either $Q^2_{\{d_1\}}$ or $Q^{(2)|2}_{\{d_1\}|\{r\}}$. If W_j was coming from $Q^2_{\{d_1\}}$, then W_j is a session-2 packet Y_j and we can simply insert Y_j into Q^2_{dec}. If W_j was coming from $Q^{(2)|2}_{\{d_1\}|\{r\}}$, then W_j is a session-1 packet X_j and there also exists a session-2 packet Y_j still unknown to d_2 where $Y_j \equiv X_j$. Moreover, d_2 has received $[Y_j + X_j]$. As a result, d_2 can further use the known $[Y_j + X_j]$ and the extracted X_j to decode Y_j and thus we can insert Y_j into Q^2_{dec}. In a nutshell, whenever d_2 receives the mixture $[X_i + \overline{W}_j]$, a session-2 packet Y_j that was unknown to d_2 can be newly decoded. The remaining reception status are $\overline{d_1 d_2} r$ and $\overline{d_1} d_2 r$. For both cases when r receives the mixture but d_1 does not, r can use the known $\overline{W}_j$ and the received $[X_i + \overline{W}_j]$ to

extract X_i. Since X_i is now known by both r and d_2 but unknown to d_1, we can thus move X_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 1 insertion.

- $s_{\mathsf{CX};6}$: s transmits $[\overline{W}_i + Y_j]$ from $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$ and $Y_j \in Q^2_{\{d_1\}}$. The movement process is symmetric to $s_{\mathsf{CX};5}$.
- $s_{\mathsf{CX};7}$: s transmits $[Y_i + \overline{W}_j]$ from $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$. The movement process is as follows.

$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$	$\overline{d_1 d_2 r}$	$\xrightarrow[\text{Case 2}]{Y_i} Q^{[1]}_{\{rd_2\}}$
$Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$\overline{d_1} d_2 \overline{r}$	$\xrightarrow{Y_j (\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$	$d_1 \overline{d_2 r}$	$\xrightarrow{X_i (\equiv Y_i)} Q^1_{\mathsf{dec}}$
$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$, $Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$\overline{d_1} d_2 r$	$\xrightarrow[\text{Case 2}]{Y_i} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j (\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$	$d_1 \overline{d_2} r$	$\xrightarrow{X_i (\equiv Y_i)} Q^1_{\mathsf{dec}}$
$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$, $Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$d_1 d_2 \overline{r}$	$\xrightarrow{X_i (\equiv Y_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j (\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
	$d_1 d_2 r$	

  - **Departure**: From the property for $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$, we know that Y_i is unknown to any of $\{d_1, r\}$, even not flagged in $\mathsf{RL}_{\{r\}}$. As a result, whenever r receives the mixture $[Y_i + \overline{W}_j]$, Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$. Moreover, $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$ is known by d_1. As a result, whenever d_1 receives the mixture, d_1 can use the known $\overline{W}_j$ and the received $[Y_i + \overline{W}_j]$ to decode Y_i and thus Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$. Similarly, one condition for $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$ is that $\overline{W}_j$ must be unknown to d_2. However when d_2 receives the mixture, d_2 can use the known $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and the received $[Y_i + \overline{W}_j]$ to decode $\overline{W}_j$. Thus $\overline{W}_j$ must be removed from $Q^{[2]}_{\{rd_1\}}$ whenever d_2 receives.
  - **Insertion**: From the properties of $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$ and $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$, we know that r contains $\{X_i, \overline{W}_j\}$; d_1 contains $\{[X_i + Y_i], \overline{W}_j\}$; and d_2 contains Y_i already. Therefore, whenever d_1 receives this mixture, d_1 can use the known $\{[X_i + Y_i], \overline{W}_j\}$ and the received $[Y_i + \overline{W}_j]$ to extract the desired X_i and thus we can insert X_i into Q^1_{dec}. Similarly, whenever d_2 receives this mixture, d_2 can use the known Y_i and the received $[Y_i + \overline{W}_j]$ to extract $\overline{W}_j$. We now need to consider case by case when $\overline{W}_j$ was inserted into $Q^{[2]}_{\{rd_1\}}$. If it was the Case 1 insertion, then $\overline{W}_j$ is a pure session-2 packet Y_j and thus we can simply insert Y_j into Q^2_{dec}. If it was the Case 2 insertion, then $\overline{W}_j$ is a pure session-1 packet $X_j \in Q^1_{\mathsf{dec}}$ and there exists a session-2 packet Y_j still unknown to d_2 where $Y_j \equiv X_j$. Moreover, d_2 has received $[Y_j + X_j]$. As a result, d_2 can further decode Y_j and thus we can insert Y_j into Q^2_{dec}. If it was the Case 3 insertion, then $\overline{W}_j$ is a mixed form of $[W_i + W_j]$ where W_i is already known by d_2 but W_j is not. As a result, d_2 can decode W_j upon receiving $\overline{W}_j = [W_i + W_j]$. Note that W_j in the Case 3 insertion $\overline{W}_j = [W_i + W_j] \in Q^{[2]}_{\{rd_1\}}$ comes from either $Q^2_{\{d_1\}}$ or $Q^{(2)|2}_{\{d_1\}|\{r\}}$. If W_j was coming from $Q^2_{\{d_1\}}$, then W_j is a session-2 packet Y_j and we can simply insert Y_j into Q^2_{dec}. If W_j was coming from $Q^{(2)|2}_{\{d_1\}|\{r\}}$, then W_j is a session-1 packet X_j and there also exists a session-2 packet Y_j still unknown to d_2 where $Y_j \equiv X_j$. Moreover, d_2 has received $[Y_j + X_j]$. As a result, d_2 can further use the known $[Y_j + X_j]$ and the extracted X_j to decode Y_j and thus we can insert Y_j into Q^2_{dec}. In a nutshell, whenever d_2 receives the mixture $[Y_i + \overline{W}_j]$, a session-2 packet Y_j that was unknown to d_2 can be newly decoded. The remaining reception status are $\overline{d_1 d_2} r$ and $\overline{d_1} d_2 r$. For both cases when r receives the mixture but d_1 does not, r can use the known $\overline{W}_j$ and the received $[Y_i + \overline{W}_j]$ to extract Y_i. Since Y_i is now known by both r and d_2 but $[X_i + Y_i]$ is in $\mathsf{RL}_{\{d_1\}}$, we can thus move Y_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 2 insertion.

- $s_{\mathsf{CX};8}$: s transmits $[\overline{W}_i + X_j]$ from $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$ and $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$. The movement process is symmetric to $s_{\mathsf{CX};7}$.
- r^1_{UC}: r transmits X_i from $X_i \in Q^1_{\{r\}}$. The movement process is as follows.

$Q^1_{\{r\}} \xrightarrow{X_i}$	$\overline{d_1} d_2$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$
	$d_1 \overline{d_2}$, $d_1 d_2$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$

  - **Departure**: One condition for $X_i \in Q^1_{\{r\}}$ is that X_i must be unknown to any of $\{d_1, d_2\}$. As a result, whenever X_i is received by any of $\{d_1, d_2\}$, X_i must be removed from $Q^1_{\{r\}}$.
  - **Insertion**: From the above discussion, we know that X_i is unknown to d_1. As a result, whenever X_i is received by d_1, we can insert X_i to Q^1_{dec}. If X_i is received by d_2 but not by d_1, then X_i is now known by both d_2 and r but still unknown to d_1. This exactly falls into the first-case scenario of $Q^{[1]}_{\{rd_2\}}$ and thus we can move X_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 1 insertion.

- r^2_{UC}: r transmits Y_j from $Y_j \in Q^2_{\{r\}}$. The movement process is symmetric to r^1_{UC}.
- $r^{(1)}_{\mathsf{DT}}$: r transmits X_i that is known by r only and information equivalent from $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$. The movement process is as follows.

$Q^{(1)\|1}_{\{d_2\}\|\{r\}} \xrightarrow{Y_i}$	$\overline{d_1} d_2$	$\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$
	$d_1 \overline{d_2}$, $d_1 d_2$	$\xrightarrow{X_i (\equiv Y_i)} Q^1_{\mathsf{dec}}$

  - **Departure**: From the property for $Y_i \in Q^{(1)|1}_{\{d_2\}|\{r\}}$, we know that there exists an information-equivalent session-1 packet X_i that is known by r but unknown to any of $\{d_1, d_2\}$. As a result, whenever X_i is received by any of $\{d_1, d_2\}$, Y_i must be removed from $Q^{(1)|1}_{\{d_2\}|\{r\}}$.
  - **Insertion**: From the above discussion, we know that X_i is unknown to d_1 and thus we can insert X_i to Q^1_{dec} whenever X_i is received by d_1. If X_i is received by d_2 but not by d_1, then X_i is now known by both d_2 and r but still unknown to d_1. This exactly falls into the first-case scenario of $Q^{[1]}_{\{rd_2\}}$ and thus we can move X_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 1 insertion.

• $r^{(2)}_{\mathsf{DT}}$: r transmits Y_j that is known by r only and information equivalent from $X_j \in Q^{(2)|2}_{\{d_1\}|\{r\}}$. The movement process is symmetric to $r^{(1)}_{\mathsf{DT}}$.

• r_{RC}: r transmits W known by r for the packet of the form $[X_i + Y_j] : W \in Q_{\mathsf{mix}}$. The movement process is as follows.

$Q_{\mathsf{mix}} \xrightarrow{[X_i+Y_j]:W}$	$\overline{d_1}d_2$	either $\xrightarrow[\text{Case 1}]{X_i} Q^{[1]}_{\{rd_2\}}$ or $\xrightarrow[\text{Case 2}]{Y_j} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$
	$d_1\overline{d_2}$	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, either $\xrightarrow[\text{Case 1}]{Y_j} Q^{[2]}_{\{rd_1\}}$ or $\xrightarrow[\text{Case 2}]{X_i} Q^{[2]}_{\{rd_1\}}$
	d_1d_2	$\xrightarrow{X_i} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j} Q^2_{\mathsf{dec}}$

- **Departure**: From the conditions of $[X_i + Y_j] : W \in Q_{\mathsf{mix}}$, we know that Q_{mix} is designed to benefit both destinations simultaneously when r transmits W. That is, whenever d_1 (resp. d_2) receives W, d_1 (resp. d_2) can decode the desired X_i (resp. Y_j), regardless whether the packet W is of a session-1 or of a session-2. However from the conditions of Q_{mix}, we know that X_i is unknown to d_1 and Y_j is unknown to d_2. Therefore, whenever W is received by any of $\{d_1, d_2\}$, $[X_i + Y_j] : W$ must be removed from $Q^{[1]}_{\{rd_2\}}$.
- **Insertion**: From the above discussions, we know that d_1 (resp. d_2) can decode the desired X_i (resp. Y_j) when W is received by d_1 (resp. d_2). As a result, we can insert X_i into Q^1_{dec} (resp. Y_j into Q^2_{dec}) when d_1 (resp. d_2) receives W. We now consider two reception status $\overline{d_1}d_2$ and $d_1\overline{d_2}$. From the conditions of Q_{mix}, note that W is always known by r and can be either X_i or Y_j. Moreover, X_i (resp. Y_j) is unknown to d_1 (resp. d_2). For the first reception case $\overline{d_1}d_2$, if X_i was chosen as W to benefit both destinations, then X_i is now known by both d_2 and r but still unknown to d_1. This exactly falls into the first-case scenario of $Q^{[1]}_{\{rd_2\}}$ and thus we move X_i into $Q^{[1]}_{\{rd_2\}}$ as the Case 1 insertion. On the other hand, if Y_j was chosen as W to benefit both destinations, then we know that Y_j is now known by both d_2 and r, and that $[X_i + Y_j]$ is already in $\mathsf{RL}_{\{d_1\}}$. This exactly falls into the second-case scenario of $Q^{[1]}_{\{rd_2\}}$ and thus we can move $Y_j \in Q^2_{\mathsf{dec}}$ into $Q^{[1]}_{\{rd_2\}}$ as the Case 2 insertion. The second reception case $d_1\overline{d_2}$ will follow the the previous arguments symmetrically.

• r_{XT}: r transmits $[W_i + W_j] \in Q^{m\mathsf{CX}}_{\{r\}}$. The movement process is as follows.

$Q^{m\mathsf{CX}}_{\{r\}} \xrightarrow{[W_i+W_j]}$	$\overline{d_1}d_2$	$\xrightarrow[\text{Case 3}]{[W_i+W_j]} Q^{[1]}_{\{rd_2\}}$, $\xrightarrow{Y_j(\equiv W_j)} Q^2_{\mathsf{dec}}$
	$d_1\overline{d_2}$	$\xrightarrow{X_i(\equiv W_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow[\text{Case 3}]{[W_i+W_j]} Q^{[2]}_{\{rd_1\}}$
	d_1d_2	$\xrightarrow{X_i(\equiv W_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j(\equiv W_j)} Q^2_{\mathsf{dec}}$

- **Departure**: From the property for $[W_i + W_j] \in Q^{m\mathsf{CX}}_{\{r\}}$, we know that W_i is known only by d_2 and that W_j is only known by d_1. As a result, whenever d_1 receives this mixture, d_1 can use the known W_j and the received $[W_i + W_j]$ to extract W_i and thus the mixture must be removed from $Q^{m\mathsf{CX}}_{\{r\}}$. Similarly, whenever d_2 receives this mixture, d_2 can use the known W_i and the received $[W_i + W_j]$ to extract W_j and thus the mixture must be removed from $Q^{m\mathsf{CX}}_{\{r\}}$.
- **Insertion**: From the above discussions, we have observed that whenever d_1 (resp. d_2) receives the mixture, d_1 (resp. d_2) can extract W_i (resp. W_j). From the four cases study of $Q^{m\mathsf{CX}}_{\{r\}}$, we know that d_1 (resp. d_2) can decode a desired session-1 packet X_i (resp. session-2 packet Y_j) whenever d_1 (resp. d_2) receives the mixture, and thus we can insert X_i (resp. Y_j) into Q^1_{dec} (resp. Q^2_{dec}). We now consider the reception status $\overline{d_1}d_2$ and $d_1\overline{d_2}$. If d_2 receives the mixture but d_1 does not, then d_1 contained W_j and d_2 now contains $[W_i + W_j]$. Moreover, $[W_i + W_j]$ was transmitted from r. This falls exactly into the third-case scenario of $Q^{[1]}_{\{rd_2\}}$. As a result, we can move $[W_i + W_j]$ into $Q^{[1]}_{\{rd_2\}}$ as the Case 3 insertion. The case when the reception status is $d_1\overline{d_2}$ can be symmetrically followed such that we can move $[W_i + W_j]$ into $Q^{[2]}_{\{rd_1\}}$ as the Case 3 insertion.

• $r^{[1]}_{\mathsf{DT}}$: r transmits $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$. The movement process is as follows.

do nothing	$\overline{d_1}d_2$	do nothing
$Q^{[1]}_{\{rd_2\}} \xrightarrow{\overline{W}_i}$	$d_1\overline{d_2}$	$\xrightarrow{X_i(\equiv \overline{W}_i)} Q^1_{\mathsf{dec}}$
	d_1d_2	

- **Departure**: One condition for $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$ is that $\overline{W}_i$ is known by d_2 unknown to d_1. As a result, whenever d_1 receives, $\overline{W}_i$ must be removed from $Q^{[1]}_{\{rd_2\}}$. Since $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$ is already known by d_2, nothing happens if it is received by d_2.
- **Insertion**: From the previous observation, we only need to consider the reception status when d_1 receives $\overline{W}_i$. For those $d_1\overline{d_2}$ and d_1d_2, we need to consider case by case when $\overline{W}_i$ was inserted into $Q^{[1]}_{\{rd_2\}}$. If it was the Case 1 insertion, then $\overline{W}_i$ is a pure session-1 packet X_i and thus we can simply insert X_i into Q^1_{dec}. If it was the Case 2 insertion, then $\overline{W}_i$ is a pure session-2 packet $Y_i \in Q^2_{\mathsf{dec}}$ and there exists a session-1 packet X_i still unknown to d_1 where $X_i \equiv Y_i$. Moreover, d_1 has received $[X_i + Y_i]$. As a result, d_1 can further decode X_i and thus we can insert X_i into Q^1_{dec}. If it was the Case 3 insertion, then $\overline{W}_i$ is a mixed form of $[W_i + W_j]$ where W_j is already known by d_1 but W_i is not. As a result, d_1 can decode W_i upon receiving $\overline{W}_i = [W_i + W_j]$. Note that W_i in the Case 3 insertion $\overline{W}_i = [W_i + W_j] \in Q^{[1]}_{\{rd_2\}}$ comes from either $Q^1_{\{d_2\}}$ or $Q^{(1)|1}_{\{d_2\}|\{r\}}$. If W_i was coming from $Q^1_{\{d_2\}}$, then W_i is a session-1 packet X_i and we can simply insert X_i into Q^1_{dec}. If W_i was coming from $Q^{(1)|1}_{\{d_2\}|\{r\}}$, then W_i is a session-2 packet Y_i and there also exists a session-1 packet X_i still unknown to d_1 where $X_i \equiv Y_i$. Moreover, d_1 has received $[X_i + Y_i]$. As a result, d_1 can further use the known $[X_i + Y_i]$ and the extracted Y_i to decode X_i and thus we can insert X_i into Q^1_{dec}. In a nutshell, whenever

d_1 receives $\overline{W}_i$, a session-1 packet X_i that was unknown to d_1 can be newly decoded.

• $r^{[2]}_{\mathsf{DT}}$: r transmits $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$. The movement process is symmetric to $r^{[1]}_{\mathsf{DT}}$.

• r_{CX}: r transmits $[\overline{W}_i + \overline{W}_j]$ from $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$ and $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$. The movement process is as follows.

$Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$\overline{d_1} d_2$	$\xrightarrow{Y_j (\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$
$Q^{[1]}_{\{rd_2\}} \xrightarrow{\overline{W}_i}$	$d_1 \overline{d_2}$	$\xrightarrow{X_i (\equiv \overline{W}_i)} Q^1_{\mathsf{dec}}$
$Q^{[1]}_{\{rd_2\}} \xrightarrow{\overline{W}_i}$, $Q^{[2]}_{\{rd_1\}} \xrightarrow{\overline{W}_j}$	$d_1 d_2$	$\xrightarrow{X_i (\equiv \overline{W}_i)} Q^1_{\mathsf{dec}}$, $\xrightarrow{Y_j (\equiv \overline{W}_j)} Q^2_{\mathsf{dec}}$

- **Departure**: From the property for $\overline{W}_i \in Q^{[1]}_{\{rd_2\}}$, we know that $\overline{W}_i$ is known by d_2 but unknown to d_1. Symmetrically, $\overline{W}_j \in Q^{[2]}_{\{rd_1\}}$ is known by d_1 but unknown to d_2. As result, whenever d_1 (resp. d_2) receives the mixture, d_1 (resp. d_2) can use the known $\overline{W}_j$ (resp. $\overline{W}_i$) and the received $[\overline{W}_i + \overline{W}_j]$ to extract $\overline{W}_i$ (resp. $\overline{W}_j$). Therefore, we must remove $\overline{W}_i$ from $Q^{[1]}_{\{rd_2\}}$ whenever d_1 the mixture and remove $\overline{W}_j$ from $Q^{[2]}_{\{rd_1\}}$ whenever d_2 receives.
- **Insertion**: From the above discussions, we have observed that whenever d_1 (resp. d_2) receives the mixture, d_1 (resp. d_2) can extract $\overline{W}_i$ (resp. $\overline{W}_j$). We first focus on the case when d_1 receives the mixture. For those $d_1\overline{d_2}$ and $d_1 d_2$, we can use the same arguments for $\overline{W}_i$ as described in the Insertion process of $r^{[1]}_{\mathsf{DT}}$. Following these case studies, one can see that a session-1 packet X_i that was unknown to d_1 can be newly decoded whenever d_1 receives $\overline{W}_i$. The reception status when d_2 receives the mixture can be followed symmetrically such that d_2 can always decode a new session-2 packet Y_j that was unknown before.

APPENDIX C
DETAILED DESCRIPTION OF ACHIEVABILITY SCHEMES IN FIG. 4

In the following, we describe (R_1, R_2) rate regions of each suboptimal achievability scheme used for the numerical evaluation in Section III-D.

• **Intra-Flow Network Coding only:** The rate regions can be described by Proposition 2, if the variables $\{s^k_{\mathsf{PM1}}, s^k_{\mathsf{PM2}}, s^k_{\mathsf{RC}} :$ for all $k \in \{1,2\}\}$, $\{s_{\mathsf{CX};l}\,(l = 1, \cdots, 8)\}$, $\{r_{\mathsf{RC}}, r_{\mathsf{XT}}, r_{\mathsf{CX}}\}$ are all hardwired to 0. Namely, we completely shut down all the variables dealing with cross-packet-mixtures. After such hardwirings, Proposition 2 is further reduced to the following form:

$$1 \geq \sum_{k \in \{1,2\}} \left(s^k_{\mathsf{UC}} + s^k_{\mathsf{DX}} + r^k_{\mathsf{UC}} + r^{[k]}_{\mathsf{DT}} \right),$$

and consider any $i, j \in (1,2)$ satisfying $i \neq j$. For each (i,j) pair (out of the two choices $(1,2)$ and $(2,1)$),

$$R_i \geq s^i_{\mathsf{UC}} \cdot p_s(d_i, d_j, r),$$
$$s^i_{\mathsf{UC}} \cdot p_{s \to \overline{d_i d_j} r} \geq r^i_{\mathsf{UC}} \cdot p_r(d_i, d_j),$$
$$s^i_{\mathsf{UC}} \cdot p_{s \to \overline{d_i} d_j \overline{r}} \geq s^i_{\mathsf{DX}} \cdot p_s(d_i, r),$$
$$s^i_{\mathsf{UC}} \cdot p_{s \to \overline{d_i} d_j r} + s^i_{\mathsf{DX}} \cdot p_s(\overline{d_i} r) + r^i_{\mathsf{UC}} \cdot p_{r \to \overline{d_i} d_j} \geq r^{[i]}_{\mathsf{DT}} \cdot p_r(d_i),$$
$$\left(s^i_{\mathsf{UC}} + s^i_{\mathsf{DX}} \right) \cdot p_s(d_i) + \left(r^i_{\mathsf{UC}} + r^{[i]}_{\mathsf{DT}} \right) \cdot p_r(d_i) \geq R_i.$$

• **[9] without Relaying:** This scheme completely ignores the relay r in the middle, and s just performs 2-user broadcast channel LNC of [9]. The corresponding rate regions can be described as follows:

$$\frac{R_1}{p_s(d_1)} + \frac{R_2}{p_s(d_1, d_2)} \leq 1,$$
$$\frac{R_1}{p_s(d_1, d_2)} + \frac{R_2}{p_s(d_2)} \leq 1.$$

• **Routing without Relaying:** This scheme completely ignores the relay r in the middle, and s just performs uncoded routing. The corresponding rate regions can be described as follows:

$$\frac{R_1}{p_s(d_1)} + \frac{R_2}{p_s(d_2)} \leq 1.$$