Two-Flow Capacity Region Of The COPE Principle For Wireless Butterfly Networks with Broadcast Erasure Channels

Wei-Cheng Kuo, Chih-Chun Wang Member, IEEE
Center of Wireless Systems and Applications (CWSA)
School of Electrical and Computer Engineering, Purdue University, USA

Abstract—This paper characterizes the full capacity region of the COPE principle for 2-flow wireless butterfly networks with broadcast packet erasure channels (PECs). The capacity results in this work allow for random overhearing with arbitrary overhearing probabilities, arbitrary scheduling policies, network-wide channel state information (CSI) feedback after each transmission, and potential use of non-linear network codes. An information-theoretic outer bound is derived that takes into account the delayed CSI feedback of the underlying broadcast packet erasure channels. For the achievability, this paper proposes a new class of linear network codes, named as the Space-Based Linear Network Coding (SBLNC), that achieves the capacity outer bound. Further, the proposed outer and inner bounds are later generalized for the setting in which a transmission may be heard by its 2-hop neighbor(s), the so-called opportunistic routing scenario. When allowing the possibility of opportunistic routing, the proposed inner and outer bounds do not always meet. Numerical experiments, however, show that the relative gap of the two bounds is less than 0.08% in average. The proposed bounds thus tightly bracket the capacity region even when combining the COPE principle with opportunistic routing.

I. INTRODUCTION

The seminal work by Li et al. in 2003 [1] shows that, linear network coding (LNC) achieves the single-flow multicast network capacity. However, most flows are unicast. For the setting of multiple unicast flows, the capacity region of network coding (and/or LNC) remains largely unknown. In 2006, Katti et al. proposed a new multiple-unicast LNC protocol, called “COPE” [2], which realizes the network coding gain by exploiting the wireless network diversity created by overhearing packets of other coexisting flows. Take the 5-node wireless butterfly network in Fig. 1(a) for example. Suppose source \(s_1\) would like to send a packet \(X\) to destination \(d_1\); source \(s_2\) would like to send a packet \(Y\) to \(d_2\); and they share a common relay \(r\). Also suppose that when \(s_1\) (resp. \(s_2\)) sends \(X\) (resp. \(Y\)) to \(r\), destination \(d_2\) (resp. \(d_1\)) can overhear packet \(X\) (resp. \(Y\)). We further assume that after the first two transmissions, both \(d_1\) and \(d_2\) can use feedback\(^1\) to inform \(r\) the overhearing status at \(d_1\) and \(d_2\), respectively. Then instead of transmitting two packets \(X\) and \(Y\) separately, the relay node \(r\) can send the linear combination \([X + Y]\). Each destination \(d_i\) can then decode its desired packet by subtracting the overheard packet from the linear combination \([X + Y]\). In [2], it was shown empirically that the throughput improvement of the above COPE principle ranges from 40% to 200% in a multi-hop testbed environment. Throughout this paper, we use the term “COPE principle” to denote the above scenario that involves at least two sessions and periodic reception status feedback. The COPE principle exploits the following two types of coding gain to improve the overall throughput. The first one is “the side information” coding gain, for which the destination \(d_j\) overhears the transmission from \(s_i\) [3], [4]. The second one is “the feedback coding gain in broadcast channels,” which follows from the unmatched reception status when relay \(r\) is broadcasting packets to \(d_1\) and \(d_2\), respectively [5]–[7].

Despite its simple nature, the exact capacity region of the COPE principle remains an open problem even for the simplest case of two coexisting flows. Several attempts have since been made to quantify some suboptimal achievable rate regions of the COPE principle [8]–[17]. One difficulty of deriving the capacity region is due to the use of feedback in the COPE principle. It is shown in [5] that although feedback could strictly enhance the capacity in a multi-unicast environment, the

\(^1\)The use of feedback is critical in the COPE principle. Without feedback, the relay \(r\) will not know that sending the linear combination \([X + Y]\) can serve two flows simultaneously.
exact amount of throughput improvement is hard to quantify. Compared with the capacity-based approaches, the results in [4] proposes a queue-based approach for the general wireline and wireless networks while considering both inter-session and intra-session network coding. The proposed queue-based scheme explores the possible inter-session network coding chance by tracking the side information received from the opposite source. However, the results in [4] mainly focus on the side information benefits and on deciding whether to apply inter-session or intra-session network coding. Similarly, [3] circumvents the difficulty of feedback-based analysis by considering a special class of 2-staged coding schemes. Although the results in [3] fully quantify the benefits of message side information [4], [5], [18]–[22], they capture only partially the feedback benefits, which leads again to a strictly suboptimal achievable rate region.

Recently, [6] and [7] successfully characterized the full capacity region of the 1-hop broadcast packet erasure channel with $\leq 3$ coexisting flows. The results in [6] and [7] prompt the possibility of fully characterizing the capacity of the COPE principle over the wireless butterfly network in Fig. 1(a). For comparison, our work focuses on the wireless butterfly network in Fig. 1(a) while [6] and [7] focus on the 1-hop broadcast channel. For the wireless butterfly network in Fig. 1(a), the network designer faces both the scheduling problem: which node (out of the two source nodes $s_1$, $s_2$, and the relay node $r$) to transmit at the current time slot, and the network coding problem: how to combine the heard/overheard packets and generate the network coded packets. For the 1-hop broadcast channel considered in [6] and [7], there is no scheduling problem since there is only one base station and the base station transmits all the time. As will be shown shortly, for a wireless butterfly network, the feedback/control messages may propagate through the entire network and affect dynamically the scheduling and coding decisions for all three nodes $s_1$, $s_2$, and $r$, which further complicates the analysis.

In this paper, we first characterize the full capacity region of the above COPE principle for 2-flow wireless butterfly networks with broadcast PECs. The setting considered in this work allows for memoryless random overhearing with arbitrary overhearing probabilities, arbitrary scheduling policies, network-wide channel state information (CSI) feedback after each transmission, and potential use of non-linear network codes. An information-theoretic outer bound is derived that fully takes into account the delayed CSI feedback of the underlying broadcast packet erasure channels. This paper then proposes a new class of linear network codes, named as the Space-Based Linear Network Coding (SBLNC), for which the throughput can be quantified by solving a linear programming problem. We prove that the SBLNC scheme achieves the capacity region of the COPE principle.

Recently, new opportunistic routing (OpR) protocols, such as the MORE protocol [23]–[25], take advantage of the observation that in Fig. 1(a) the packet sent by source $s_i$ can sometimes be heard directly by its two-hop-away destination $d_i$. (The 2-hop overhearing probability is generally much less than the success probability of the 1-hop direct transmission from $s_i$ to $r$.) See Fig. 1(b) for illustration. [23] and [25] show that such 2-hop overhearing can also be used to significantly improve the throughput by the concept of opportunistic routing without relying on the COPE principle (i.e., no coding across multiple unicast flows). Throughout this paper, we will use “OpR” to denote the scenario in which $d_i$ can directly hear the transmission of $s_i$ with non-zero probability for $i = 1, 2$. An interesting question is thus what is the capacity region when we combine both the COPE principle and opportunistic routing. To that end, the proposed outer and inner bounds are further generalized for the setting in which a transmission may be heard by its 2-hop neighbor(s). In contrast with the case when only the COPE principle is considered, the proposed inner and outer bounds are no longer tight after taking into account the possibility of performing opportunistic routing.

Extensive numerical experiments are used to quantify the gap between the two bounds. The results show that for practical settings, the relative gap of the two bounds is less than 0.08% in average. The proposed bounds thus tightly bracket the capacity region even when combining the COPE principle with opportunistic routing.

The rest of the paper is organized as follows. Section II formulates the problem. Section III describes the main result of this paper, the full capacity region of the COPE principle for 2-flow wireless butterfly networks with broadcast PECs. Section IV introduces a new class of LNC, named as the Space-Based Linear Network Code (SBLNC), and provides some examples and motivations for designing the SBLNC code. Section V quantifies the performance of SBLNC and shows that when focusing exclusively on the COPE principle without considering opportunistic routing, SBLNC achieves the capacity. Section VI reports the results of several numerical experiments. Section VII concludes the paper.

II. Problem Formulation

A. Memoryless Broadcast Packet Erasure Channels

For any positive integer $M$, define $[M] \triangleq \{1, \cdots, M\}$. A 1-to-$M$ packet erasure channel takes an input $x$ from a

3The concept of “opportunistic routing” is different from the term “intra-session random linear network coding (RLNC).” For example, one way of realizing the OpR throughput improvement is by RLNC [23]. However, there are other ways of realizing the OpR benefits [26]. Also, when periodic feedback is not available [3], [4], [17], it is shown that even without direct links, one still needs to use intra-session network coding to fully realize the side information benefits of the COPE principle.
finite field GF(q) and outputs an M-dimensional vector \( y = (y_1, y_2, \ldots, y_M) \), where \( y_i \in \{x, *\} \) for all \( i \in [M] \). Here * denotes the erasure symbol. \( y_i = * \) means that the \( i \)-th receiver does not receive the input \( x \). We also assume that there is no other type of noise, i.e., the received \( y_i \) is either \( x \) or *.

We consider only memoryless PECs, i.e., the erasure pattern is independently and identically distributed (i.i.d.) for each channel usage. The characteristics of a memoryless 1-to-M PEC can be fully described by \( 2^M \) successful reception probabilities \( p_{s \rightarrow T(M)^T} \) indexed by any subset \( T \subseteq [M] \). That is, \( p_{s \rightarrow T(M)^T} \) denote the probability that a packet \( x \) sent from source \( s \) is heard by and only by the \( i \)-th destination for all \( i \in T \).

### B. The COPE Principle for 2-Flow Wireless Butterfly Networks with Broadcast PECs

The COPE principle for the 2-flow wireless butterfly networks with broadcast PECs is modeled as follows. We consider a 5-node 2-hop relay network with two source-destination pairs \((s_1, d_1)\) and \((s_2, d_2)\) and a common relay \( r \) interconnected by three broadcast PECs. See Fig. 2 for illustration. Specifically, source \( s_1 \) can use a 1-to-3 broadcast PEC to communicate with \( \{d_1, d_2, r\} \) for \( i = 1, 2 \), and relay \( r \) can use a 1-to-2 broadcast PEC to communicate with \( \{d_1, d_2\} \). To accommodate the discussion of opportunistic routing, we allow \( s_1 \) to directly communicate with \( d_1 \), see Fig. 2. When opportunistic routing is not permitted (as in the case when focusing exclusively on the COPE principle), we simply choose the PEC channel success probabilities \( p_{s_i \rightarrow r} \) such that the probability that \( d_i \) can hear the transmission from \( s_i \) is zero.

We assume slotted transmission. Within an overall time budget of \( n \) time slots, source \( s_i \) would like to convey \( nR_i \) packets \( W_i \equiv (W_{i,1}, \ldots, W_{i,nR_i}) \) to destination \( d_i \) for all \( i \in \{1, 2\} \) where \( R_i \) is the rate for flow \( i \). For all \( i \in \{1, 2\} \) and \( j \in [nR_i] \), the information packet \( W_{i,j} \) is assumed to be independently and uniformly distributed over GF(q).

For any time \( t \), we use an 8-dimensional channel status vector \( Z(t) \) to represent the channel reception status of the entire network:

\[
Z(t) = (Z_{s_1 \rightarrow d_1}(t), Z_{s_1 \rightarrow d_2}(t), Z_{s_1 \rightarrow r}(t), Z_{s_2 \rightarrow d_1}(t), Z_{s_2 \rightarrow d_2}(t), Z_{s_2 \rightarrow r}(t), Z_{r \rightarrow d_1}(t), Z_{r \rightarrow d_2}(t)) \in \{*, 1\}^8
\]

where “*” and “1” represent erasure and successful reception, respectively. For example, when \( s_1 \) transmits a packet \( X_{s_1}(t) \in GF(q) \) in time \( t \), relay \( r \) receives \( Y_{s_1 \rightarrow r}(t) = X_{s_1}(t) \) if \( Z_{s_1 \rightarrow r}(t) = 1 \) and receives \( Y_{s_1 \rightarrow r}(t) = * \) if \( Z_{s_1 \rightarrow r}(t) = * \). For simplicity, we use \( Y_{s_i \rightarrow r}(t) = X_{s_i}(t) \circ Z_{s_i \rightarrow r}(t) \) as shorthand.

In this work, we consider the node-exclusive interference model. That is, we allow only one node to be scheduled in each time slot. The scheduling decision at time \( t \) is denoted by \( \sigma(t) \), which takes value in the set \( \{s_1, s_2, r\} \). For example, \( \sigma(t) = s_1 \) means that node \( s_1 \) is scheduled for time slot \( t \). For convenience, when \( s_1 \) is not scheduled at time \( t \), we simply set \( Y_{s_1 \rightarrow r}(t) = * \). As a result, the scheduling decision can be incorporated into the following expression of \( Y_{s_i \rightarrow r}(t) \):

\[
Y_{s_i \rightarrow r}(t) = X_{s_i}(t) \circ Z_{s_i \rightarrow r}(t) \circ 1_{\{\sigma(t) = s_i\}}.
\]

Similar notation is used for all other received signals. For example, \( Y_{r \rightarrow d_2}(t) = X_r(t) \circ Z_{r \rightarrow d_2}(t) \circ 1_{\{\sigma(t) = r\}} \) is what \( d_2 \) receives from \( r \) in time \( t \), where \( X_r(t) \) is the packet sent by \( r \) in time \( t \).

We assume that the 3 PECs are memoryless and stationary. Namely, we allow arbitrary joint distribution for the 8 coordinates of \( Z(t) \) but assume that \( Z(t) \) is i.i.d. over the time axis \( t \). For example, the individual random variables \( Z_{s_2 \rightarrow d_1}(t) \) and \( Z_{s_2 \rightarrow r}(t) \) may be dependent but the two random vectors \( Z(t_1) \) and \( Z(t_2) \) are independent as long as \( t_1 \neq t_2 \). We also assume that the random process \( \{Z(t) : \forall t\} \) is independent of the information messages \( W_1 \) and \( W_2 \).

For simplicity, we use brackets \([i]_4 \) to denote the collection from time 1 to \( t \). For example, \( \{\sigma(r), Z_{s_1 \rightarrow d_1}(t), Z_{s_2 \rightarrow r}(t) : \forall t \in \{1, 2, \ldots, t\}\} \). Also, for any \( S \subseteq \{s_1, s_2, r\} \) and \( T \subseteq \{r, d_1, d_2\} \), we define

\[
Y_{S \rightarrow T}(t) \Deltaq \{Y_{s \rightarrow d}(t) : \forall s \in S, \forall d \in T\}.
\]

For example, \( Y_{\{s_1, r\} \rightarrow \{d_1, d_2\}}(t) \) is the collection of \( Y_{s_1 \rightarrow d_1}(t), Y_{s_1 \rightarrow d_2}(t), Y_{r \rightarrow d_1}(t) \), and \( Y_{r \rightarrow d_2}(t) \).

Given the rate vector \((R_1, R_2)\), a joint scheduling and network coding (NC) scheme is defined by \( n \) scheduling decision functions

\[
\forall t \in [n], \sigma(t) = f_{\sigma,t}([Z]_{t-1}^{t-1})
\]

3n encoding functions at \( s_1, s_2, \) and \( r \), respectively: For all \( t \in [n] \)

\[
X_{s_i}(t) = f_{s_i,t}([W_i]_{t-1}^{t-1}), \forall i \in \{1, 2\}
\]

\[
X_r(t) = f_{r,t}([Y_{\{s_1, s_2\} \rightarrow r}]_{t-1}^{t-1})
\]

and 2 decoding functions at \( d_1 \) and \( d_2 \), respectively:

\[
W_i = f_{d_i}([Y_{\{s_1, s_2, r\} \rightarrow d_i}]_{t-1}^{t-1}), \forall i \in \{1, 2\}.
\]

By (1), we allow \( \sigma(t) \), the scheduling decision at time \( t \), to be a function of the network-wide reception status vectors before time \( t \). By (2), the encoding decision at \( s_i \) is a function depending on the information messages and past channel status. Encoding at \( r \) depends on what \( r \) received in the past and the past channel status vector, see (3). In the end, \( d_i \) decodes \( W_i \) based on what \( d_i \) received and the past channel status of the entire network\(^4\). We allow the encoding and decoding functions \( f_{s_i,t}, f_{r,t}, \) and \( f_{d_i} \) to be linear or nonlinear.

This setting models the scenario in which there is a dedicated, error-free, low-rate control channel that can broadcast the previous network channel status \( Z(t - 1) \) causally to all network nodes. The total amount of control information is no larger than 8 bits per time slot, which is much smaller than the actual payload of each packet \( \approx 10^4 \) bits. As a result, the perfect feedback channel could be implemented by

\(^4\)Since the scheduling decision \( \sigma(t) \) is a function of \( [Z]_{t-1}^{t-1} \), all the encoding functions in (2) and (3), and the decoding functions in (4) also know implicitly the scheduling decision \( \sigma(t) \).
piggybacking\(^3\) on the data packets. The scheduling decision \(\sigma(t)\) can be computed centrally (by a central controller) or distributively by each individual node since we allow all nodes to have the knowledge of the reception status of the entire network. This work mainly focuses on the theoretical capacity analysis of the feedback-enabled butterfly networks. Hence we omit the discussion of any practical implementation issues like buffer management and time synchronization, the latter of which is critical when implementing the scheduler \(\sigma(t)\).

**Definition 1:** Fix the distribution of \(Z(t)\) and finite field \(GF(q)\). A rate vector \((R_1, R_2)\) is achievable if for any \(\epsilon > 0\), there exists a joint scheduling and NC scheme with sufficiently large \(n\) such that

\[
\max_{i \in \{1, 2\}} \text{Prob}(W_i \neq \hat{W}_i) < \epsilon.
\]

The capacity region is defined as the closure of all achievable rate vectors \((R_1, R_2)\).

**Remark:** In (1), the scheduling decision \(\sigma(t)\) does not depend on the information messages \(W_i\), which means that we prohibit the use of timing channels [27], [28]. Even when we allow the usage of timing channels, we conjecture that the overall capacity improvement with the timing channel techniques is negligible. A heuristic argument is that each successful packet transmission gives \(\log_2(q)\) bits of information while the timing information (to transmit or not) gives roughly 1 bit of information. When focusing on sufficiently large \(GF(q)\), additional gain of timing information is thus likely to be absorbed in our timing-information-free capacity characterization. In our setting, \(r\) is the only node that can mix packets from two different data flows. Further relaxation such that \(s_1\) and \(s_2\) can hear each other and perform coding accordingly is beyond the scope of this work.

### C. A Useful Notation

In our network model, there are 3 broadcast PECs associated with \(s_1\), \(s_2\), and \(r\), respectively. We sometimes call those PECs the \(s_i\)-PEC, \(i = 1, 2\), and the \(r\)-PEC. Since only one node can be scheduled in each time slot, we can assume that the reception events of each PEC are independent from that of the other PECs. As a result, the distribution of the network-wide channel status vector \(Z(t)\) can be described by the probabilities \(p_{s_i \rightarrow r}^{T_{\{r, d_1, d_2\}}}\) and for all \(T \subseteq \{r, d_1, d_2\}\) and for all \(U \subseteq \{d_1, d_2\}\) of which is critical when implementing the scheduler \(\sigma(t)\).

\[
\max_{i \in \{1, 2\}} \text{Prob}(W_i \neq \hat{W}_i) < \epsilon.
\]

The capacity region is defined as the closure of all achievable rate vectors \((R_1, R_2)\).

**Remark:** In (1), the scheduling decision \(\sigma(t)\) does not depend on the information messages \(W_i\), which means that we prohibit the use of timing channels [27], [28]. Even when we allow the usage of timing channels, we conjecture that the overall capacity improvement with the timing channel techniques is negligible. A heuristic argument is that each successful packet transmission gives \(\log_2(q)\) bits of information while the timing information (to transmit or not) gives roughly 1 bit of information. When focusing on sufficiently large \(GF(q)\), additional gain of timing information is thus likely to be absorbed in our timing-information-free capacity characterization. In our setting, \(r\) is the only node that can mix packets from two different data flows. Further relaxation such that \(s_1\) and \(s_2\) can hear each other and perform coding accordingly is beyond the scope of this work.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Features in [3]</th>
<th>Features in this work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
<td>(1) Allow only sequential scheduling, (2) Allow only batch feedback, (3) Arbitrary number of flows.</td>
<td>(1) Allow dynamic scheduling, (2) Allow per-packet feedback, (3) 2 flows.</td>
</tr>
<tr>
<td>The outer bound</td>
<td>Side-information-based index coding analysis.</td>
<td>Full feedback and scheduling analysis.</td>
</tr>
<tr>
<td>The inner bound</td>
<td>(1) Sequential scheduling, (2) Linear coding functions.</td>
<td>(1) Sequential scheduling, (2) Linear coding functions.</td>
</tr>
<tr>
<td>Contribution</td>
<td>Suboptimal achievable scheme for an arbitrary number of flows.</td>
<td>Exact 2-flow capacity region of the COPE principle with per packet scheduling.</td>
</tr>
</tbody>
</table>

### D. Comparison to Existing Works and Practical COPE Protocol

In this subsection, we compare the network model of this paper with the existing message-side-information-based results [3] and the practical COPE-based schemes.

There are three major differences between the setting of this work and in [3]. First, the setting of the outer bound in [3] is restricted to considering only the deterministic sequential scheduling policy, which schedules nodes \(s_1\), \(s_2\), and \(r\) in a strict order. Namely, \(s_1\) transmits first. After \(s_1\) stops, \(s_2\) can begin to transmit. Only after \(s_2\) stops transmission can \(r\) start its own transmission. For comparison, the setting of our outer bound derivation allows for dynamically choosing

\[
p_{s_1}(d_2 r) = p_{s_1 \rightarrow d_2 r} + p_{s_1 \rightarrow d_2 r} \tag{5}
\]

which represents the probability that a packet sent by \(s_1\) is successfully received by \(d_2\) but not by \(r\). Herein, \(d_1\) is a don’t-care receiver and \(p_{s_1}(d_2 r)\) thus sums two joint probabilities together \((d_1\) receives it or not) as described in (5).

Another example is \(p_{r}(d_2) = p_{r \rightarrow d_1 d_2} + p_{r \rightarrow d_1 d_2}\), which is the probability that a packet sent by \(r\) is heard by \(d_2\). To slightly abuse the notation, we further allow \(p_{s}()\) to take multiple input arguments separated by the comma sign “,”. With this new notation, \(p_{s}()\) then represents the probability that the reception event is compatible to at least one of the input arguments. For example,

\[
p_{s_1}(d_1 d_2 r) = p_{s_1 \rightarrow d_1 d_2 r} + p_{s_1 \rightarrow d_1 d_2 r} + p_{s_1 \rightarrow d_1 d_2 r} + p_{s_1 \rightarrow d_1 d_2 r} + p_{s_1 \rightarrow d_1 d_2 r} + p_{s_1 \rightarrow d_1 d_2 r}.
\]

That is, \(p_{s_1}(d_1 d_2 r)\) represents the probability that \((Z_{s_1 \rightarrow d_1}, Z_{s_2 \rightarrow d_2}, Z_{s_1 \rightarrow r})\) equals one of the following 5 vectors \((1, *, *), (1, 1, 1), (%1, 1, 1), (1, * 1, 1)\), and \((*, *, 1)\). Note that these 5 vectors are compatible to either \(d_1 d_2\) or \(r\) or both. Another example of this \(p_{s}()\) notation is \(p_{s}(d_1, d_2, r)\), which represents the probability that a packet sent by \(s_1\) is received by \(d_1\) at least one of the three nodes \(d_1\), \(d_2\), and \(r\). In the following context, we slightly abuse the notation \(p_{s}(\cdot)\) is defined as \(p_{s}(\cdot)\) for \(i = 1, 2\) when there is no ambiguity.
the schedule $\sigma(t)$ for each time slot $t$ depending on the past reception status $[Z]_{t-1}$, which includes any store-\&-forward-based scheduling policies as special cases, such as the back-pressure and the maximal weighted matching schemes (see [25] for references). Our outer bound result thus quantifies the best possible achievable rates with jointly designed scheduling and coding policies.

Secondly, in [3] no feedback is allowed when $s_1$ and $s_2$ transmit. More specifically, suppose jointly $s_1$ and $s_2$ take $t_{s_1} + t_{s_2}$ time slots to finish transmission. Then only in the beginning of time $(t_{s_1} + t_{s_2} + 1)$ are we allowed to send the channel status $[Z]_{t_{s_1} + t_{s_2}}$ to $r$. No further feedback is allowed until time $n$, the end of overall transmission. For comparison, our setting allows constantly broadcasting network-wide channel status $[Z]_{t_{s_1}-1}$ to $s_1$, $s_2$, and $r$, as discussed in Section II-B. This setting thus includes the Automatic Repeat reQuest (ARQ) mechanism as a special case [3], [6]. Broadcasting the control information $[Z]_{t_{s_1}-1}$ to all the network nodes also eliminates the need of estimating/learning the reception status of the neighbors. Thirdly, [3] focuses on an arbitrary number of coexisting flows while this work focuses exclusively on the 2-flow scenario.

In addition to different settings, the outer bound analysis of [3] focuses on the side-information-based index-coding-like analysis, while this work focuses on full feedback and scheduling analysis. Although both the achievability schemes in [3] and in this work are based on linear network coding with sequential scheduling, the latter has to take into account the per-slot causal feedback in the LNC design. The differences between [3] and this work are summarized in Table I.

We also compare the analytical results in this work to the practical COPE implementation in [2]. The COPE protocol in [2] contains three major components: (i) Opportunistic listening: Each destination is in a promiscuous monitoring mode and stores all the overheard packets; (ii) Opportunistic coding: The relay node decides which packets to be coded together opportunistically, based on the overhearing patterns of its neighbors; and (iii) Learning the states of the neighbors: Although in the practical COPE implementation reception reports are periodically sent to advertise the overhearing patterns of the next-hop neighbors of the relay, the relay node still needs to extrapolate the overhearing status of its neighbors since there is always a time lag due to the infrequent periodic feedback.

Our setting closely captures the opportunistic listening component of COPE by modeling the wireless packet transmission as a random broadcast PEC. In (1)–(3), the channel status vector is used to make the coding and scheduling decisions, which captures the opportunistic coding component of COPE. In COPE, the reception reports are broadcast periodically, which is captured by the control information $[Z]_{t_{s_1}-1}$. In sum, our capacity region is a superset of any achievable rates of any COPE-principle-based schemes [2] when focusing on the 2-flow wireless butterfly networks with broadcast PECs in Fig. 1(a) and the node exclusive interference model.

**Remark:** The setting in Section II-B also includes the wireless erasure 2-way relay channel model (Fig. 3(a) and 3(b)) as a special case. Specifically, if we set the overhearing probabilities: $p_s(d_i) = 1$ for all $i \neq j$, then the capacity region of the setting in Section II-B is also the capacity region of the wireless erasure 2-way relay channel in Fig. 3.

### III. MAIN RESULTS

In this section, we provide our results based on two cases: The case of considering only the COPE principle and the case of combining COPE with the opportunistic routing technique. The main difference is that for the former setting, we assume that no transmission can be heard by its 2-hop neighbors, i.e., $p_s(d_i) = 0$ for all $i = 1, 2$. For the latter setting, we allow $p_s(d_i)$ to be non-zero.

For the case of using exclusively the COPE principle, the full capacity region has been characterized in Section III-A while for the case of COPE plus opportunistic routing, a pair of outer and inner bounds are provided in Sections III-B and III-C, respectively.

#### A. The COPE-Principle 2-Flow Wireless Butterfly Network Capacity

**Proposition 1:** Consider any 2-flow wireless butterfly network with broadcast PECs with $p_s(d_i) = 0$ for all $i = 1, 2$ and consider any finite field $\mathbb{F}_q$. The rate pair $(R_1, R_2)$ is in the capacity region if and only if there exist three non-negative time sharing parameters $t_{s_1}, t_{s_2}$ and $t_r$ such that jointly $(R_1, R_2)$ and $(t_{s_1}, t_{s_2}, t_r)$ satisfy

$$t_{s_1} + t_{s_2} + t_r \leq 1$$

$$\forall i \in \{1, 2\}, R_i \leq t_s p_i(r)$$

$$\frac{R_1}{p_r(d_1)} + \frac{(R_2 - t_{s_2} p_2(d_1))^+}{p_r(d_1, d_2)} \leq t_r$$

$$\frac{(R_1 - t_{s_1} p_1(d_2))^+}{p_r(d_1, d_2)} + \frac{R_2}{p_r(d_2)} \leq t_r$$

where $(\cdot)^+ \triangleq \max(0, \cdot)$ is the projection to non-negative reals.

The proof of the achievability part of Proposition 1 is relegated to Section V-B and the converse proof is relegated to Appendix A.

The intuition behind (6) to (9) is as follows. (6) is a time sharing bound, which follows from the total time budget being $n$ and the node-exclusive interference model.

Inequality (7) is a simple cut-set bound. That is, the message $W_i$ has to be sent from $s_i$ to the common relay $r$ first. Therefore, the rate is upper bounded by the link capacity from $s_i$ to $r$. 

![Fig. 3.](image-url)
Inequalities (8) and (9) combine the capacity results on message-side-information [3] and the capacity results on channel output feedback for broadcast channels [6, 7]. A very heuristic, not rigorous explanation of (8) is as follows. $R_t(d_1)$ represents how many slots it takes to send all the flow-1 packets to $d_1$ as if there is no flow-2. $t_{s_2-p_2}(d_1)$ characterizes how much flow-2 information can be “overheard” by $d_1$, and $(R_2 - t_{s_2-p_2}(d_1))^+$ thus represents how much flow-2 information that has not been heard by $d_1$ but still needs to be sent to $d_2$. Since those flow-2 packets cannot be “coded” together with any flow-1 packets, they need to be sent separately by themselves in addition to the $R_t(d_2)^+$ time slots used to send flow-1 packets. In general, it takes $(R_2 - t_{s_2-p_2}(d_1))_+^+$ for those packets to arrive at $d_2$. However, [6] shows that the use of feedback can further reduce the time to $(R_2 - t_{s_2-p_2}(d_1))^+$ We then have (8) since the total transmission time of relay $r$ is $nt_r$ time slots. (9) is symmetric to (8).

B. Capacity Outer Bound for COPE plus OpR

The capacity results in Proposition 1 can be generalized as an outer bound for the case when the destination $d_1$ may overhear directly the transmission of $s_i$, i.e., $p_i(d_1) > 0$.

**Proposition 2:** Consider any 2-flow wireless butterfly network with broadcast PECs in Fig. 2 with arbitrary channel characteristics and consider any finite field GF($q$). If a rate vector $(R_1, R_2)$ is achievable, there exist three non-negative scalars $s_{t_1}$, $t_{s_2}$, and $t_r$ satisfying

$$t_{s_1} + t_{s_2} + t_r \leq 1$$

(10)

$$\forall i \in \{1, 2\}, \quad R_i \leq t_{s_1}p_i(d_i, r)$$

(11)

$$\frac{(R_1 - t_{s_1}p_i(d_1))^+}{p_i(d_1)} + \frac{(R_2 - t_{s_2-p_2}(d_1, d_2))^+}{p_i(d_1, d_2)} \leq t_r$$

(12)

$$\frac{(R_1 - t_{s_2-p_2}(d_1, d_2))^+}{p_i(d_1, d_2)} + \frac{(R_2 - t_{s_2-p_2}(d_1, d_2))^+}{p_i(d_1, d_2)} \leq t_r.$$  

(13)

This outer bound considers channel status feedback, dynamic scheduling, and possibly non-linear encoding functions, and is derived by entropy-based analysis. One major challenge of the outer bound derivation is to incorporate the effects of dynamic scheduling, which was not presented in the existing sequential-scheme-based outer bound analysis [3]. To circumvent this issue, we further analyze the time average of the mutual information over all possible realizations. The detailed proof is relegated to Appendix A.

**Remark:** One can easily see that when the channel probabilities satisfy $p_i(d_i) = 0$ for all $i = 1, 2$, the outer bound in Proposition 2 collapses to the capacity region in Proposition 1. Proposition 2 is thus a strict generalization of the converse part of Proposition 1.

C. Capacity Inner Bound for COPE plus OpR

An inner bound for the general case of $p_i(d_i) \geq 0$ is described as follows.

**Proposition 3:** A rate vector $(R_1, R_2)$ is achievable by a linear network code if there exist 3 non-negative variables $t_{s_1}$, $t_{s_2}$, $t_r$, 10 non-negative variables, $\omega^k_{r, i}$, where $i \in \{1, 2\}$ and $k \in \{0, 1, 2, 3, 4\}$, 4 non-negative variables $\omega^k_{r, N\text{, } r}$, $\omega^k_{r, C}$ for $k = 1, 2$, such that jointly the 17 variables\(^6\) and $(R_1, R_2)$ satisfy the following four groups of inequalities:

- **Group 1** has 5 inequalities, named as the *time budget constraints.*

  $$\forall i = 1, 2, \quad \sum_{k=0}^4 \omega^k_{r, i} \leq t_{s_i}$$  

(14)

$$\forall i = 1, 2, \quad \omega^1_{r, N} + \omega^2_{r, N} + \omega^4_{r, C} \leq t_r$$  

(15)

$$t_{s_1} + t_{s_2} + t_r < 1$$  

(16)

- **Group 2** has 12 inequalities, named as the *packet conservation laws at the source nodes.* 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)), we have the following 6 inequalities.

  $$\omega^0_{r, p_i}(d_i, d_j, r) \leq R_i$$  

(17)

$$\omega^1_{r, p_i}(d_i, d_j, r) \leq \omega^0_{r, p_i}(d_i, d_j)$$  

(18)

$$\omega^2_{r, p_i}(d_i, d_j, r) \leq \omega^0_{r, p_i}(r, d_j)$$  

(19)

$$\omega^3_{r, p_i}(d_i, d_j, r) \leq \omega^0_{r, p_i}(r, d_i) - \omega^3_{r, p_i}(d_i, r)$$  

(20)

$$\omega^4_{r, p_i}(d_i, d_j, r) \leq \omega^0_{r, p_i}(d_i, r) + \omega^4_{r, p_i}(d_i, d_j)$$  

(21)

$$\omega^5_{r, p_i}(d_i) + \omega^5_{r, p_i}(d_i, d_j, r) + \omega^5_{r, p_i}(r, d_i) + \omega^5_{r, p_i}(d_i, d_j)$$  

(22)

- **Group 3** has 4 inequalities, named as the *packet conservation laws at the relay node.* For each $(i, j)$ pair with $i \neq j$, we have the following 2 inequalities.

  $$\omega^4_{r, p_{r_i}}(d_i, d_j) \leq \omega^0_{s_i, p_i}(r, d_j, d_i) - \omega^4_{s_i, p_i}(d_i, d_j)$$  

(23)

$$\omega^4_{r, p_{r_i}}(d_i) \leq \omega^0_{s_i, p_i}(d_i, r) + \omega^4_{s_i, p_i}(d_i, d_j)$$  

(24)

- **Group 4** has 2 inequalities, named as the *decodability conditions.* Consider $i = 1, 2$. For each $i$, we have the following inequality.

$$\sum_{k=0}^4 \omega^k_{r, i}(d_i) + (\omega^4_{r, N, i} + \omega^4_{r, C}) p_i(d_i) \geq R_i$$  

(25)

The inner bound inequalities are based on the SBLNC scheme constructed in Section IV. A heuristic but not rigorous explanation is as follows. The time budget constraints (14)–(16) describe the fact that each time slot can be assigned to one of the transmitting nodes $s_1, s_2$, or $r$ and the overall normalized time budget is one. The conservation laws (17)–(24) correspond to the fact that to select any one of the SBLNC policies, the corresponding *coding set* of the policy must be non-empty. The non-emptiness of the coding set can

\(^6\)In the achieving algorithm in Section V, the $t$ variables correspond to the numbers of time slots that each of the sources and the relay is used; and the $\omega$ variables correspond to the numbers of time slots each policy is used.
be described by (17)–(24), which determine the size of the coding sets. The decodability conditions describe how many packets need to be received at each destination before they can decode the desired messages.

Proposition 3 will be proved by explicit construction of an achievability scheme on the SBLNC scheme described in the next section. The detailed proof of Proposition 3 is relegated to Section V-A.

IV. A Space-Based Linear Network Code(SBLNC) Construction

In the existing network coding scheme works (e.g. [1], [3]), designing the encoding coefficients is always a challenging task. With per-time-slot causal feedback, the sources (including the relay) can continuously track what has been received by the destinations and the best network coding strategy might evolve overtime, which further exacerbates the problem of designing the right coding coefficients. In this section, we introduce a new class of network coding scheme named as the “Space-Based Linear Network Code (SBLNC)” scheme that significantly simplifies the design of the coding coefficients. The SBLNC scheme will later be used to prove the capacity inner bound in Proposition 3.

A. Linear-Space-Based Definitions

We first provide some basic definitions that will be used when describing an SBLNC scheme.

For \(i = 1, 2\), a flow-1 coding vector \(v^{(i)}\) is an \(nR_i\)-dimensional row vector with each coordinate being a scalar in \(\text{GF}(q)\). Any linear combination of the message symbols \(W_{i,1}\) to \(W_{i,nR_i}\) can thus be represented by \(v^{(i)}W_i^T\) where \(W_i^T\) is the transpose of \(W_i\). We use the superscript “\((i)\)” to emphasize that we are focusing on a flow-i vector.

We define the flow-i message space by \(\Omega_i \triangleq (\text{GF}(q))^{nR_i}\), an \(nR_i\)-dimensional linear space. In the following, we define the following 6 knowledge spaces \(S_r, S_{d_2}, S_{d_4}, T_r, T_{d_1}, T_{d_2}\) for the 5-node relay network in Fig. 2.

The knowledge spaces \(S_r, S_{d_2}, S_{d_4}\) are linear subspaces of \(\Omega_1\) and represent the knowledge about the flow-1 packets at nodes \(r, d_2\), and \(d_1\), respectively. Symmetrically, the knowledge spaces \(T_r, T_{d_1}\), and \(T_{d_2}\) are linear subspaces of \(\Omega_2\) and represent the knowledge about the flow-2 packets at nodes \(r, d_1\), and \(d_2\), respectively. In the following, we discuss the detailed construction of \(S_r, S_{d_2}, S_{d_4}\), and \(T_r\) and the construction of \(T_{d_1}\) to \(T_{d_2}\) follows symmetrically.\(^7\)

• In the end of any time \(t\), \(S_r(t) \subset \Omega_1\) is the linear span of a group of \(v^{(1)}\) vectors, denoted by \(V^{(1)}_{s_1}\). The group \(V^{(1)}_{s_1} \rightarrow T_r\) contains the \(v^{(1)}\) vectors sent by \(s_1\) during time \(1\) to \(t\) and have been received successfully by \(r\). Throughout the paper, we use the convention that the linear span of an empty set is a set containing the zero vector, i.e., \(\text{span}\{\emptyset\} = \{0\}\). For example, if \(r\) has not yet received any packet from \(s_1\), then by convention \(S_r(t) = \{0\}\).

• In the end of time \(t\), \(S_{d_2}(t) \subset \Omega_1\) is the linear span of two groups of \(v^{(1)}\) vectors, denoted by \(V^{(1)}_{s_1} \rightarrow d_2\) and \(V^{(1)}_{s_2} \rightarrow d_2\). The first group \(V^{(1)}_{s_1} \rightarrow d_2\) contains the \(v^{(1)}\) vectors corresponding to the packets sent by \(s_1\) during time \(1\) to \(t\) and have been received successfully by \(d_2\). The second group \(V^{(1)}_{s_2} \rightarrow d_2\) contains the \(v^{(1)}\) vectors corresponding to the packets sent by \(s_1\) during time \(1\) to \(t\) that are not mixed with any other flow-2 packets. The letter “\(N\)” in the subscript stands for Not-inter-flow-coded transmission.

• In the end of time \(t\), \(S_{d_4}(t) \subset \Omega_1\) is the linear span of three groups of \(v^{(1)}\) vectors, denoted by \(V^{(1)}_{s_1} \rightarrow d_1\), \(V^{(1)}_{s_2} \rightarrow d_1\), and \(V^{(1)}_{s_3} \rightarrow d_1\). The first group \(V^{(1)}_{s_1} \rightarrow d_1\) contains the \(v^{(1)}\) vectors corresponding to the packets sent by \(s_1\) during time \(1\) to \(t\) and have been received successfully by \(d_1\). The second group \(V^{(1)}_{s_2} \rightarrow d_1\) contains the \(v^{(1)}\) vectors corresponding to the packets sent by \(r\) during time \(1\) to \(t\) that are not mixed with any other flow-2 packets. The third group \(V^{(1)}_{s_3} \rightarrow d_1\) contains the \(v^{(1)}\) vectors that can be decoded from the inter-flow coded packets\(^8\) sent by \(r\) during time \(1\) to \(t\). The letter “\(C\)” in the subscript stands for inter-flow-Coded transmission.

In sum, we use \(S\) and \(T\) to distinguish whether we are focusing on flow-1 or flow-2 packets, respectively, and we use the subscripts to describe the node of interest. One can easily see that these six knowledge spaces evolve over time since each node may receive more and more packets that can be used to obtain/decode new information. We use the following example to illustrate the definitions of \(S_r\) to \(T_{d_2}\).

Example 1: Consider \(\text{GF}(3)\) and \(nR_1 = 3\) and \(nR_2 = 2\). That is, flow-1 contains 3 message symbols \(W_{1,1}\) to \(W_{1,3}\) and flow-2 contains 2 message symbols \(W_{2,1}\) and \(W_{2,2}\). \(\Omega_1\) and \(\Omega_2\) are thus 3-dimensional and 2-dimensional linear spaces in \(\text{GF}(3)\), respectively. Consider the first four time slots \(t = 1\) to \(4\) for our discussion.

When \(t = 1\), suppose that \(s_1\) is scheduled; an uncoded flow-1 message symbol \(W_{1,1}\) is transmitted; and the packet is heard by and only by \(d_2\) and \(r\). See Fig. 4(a) for illustration, for which we use the solid lines to represent that \(d_2\) and \(r\) have received the packet. We use the dashed line to denote that \(d_1\) does not receive the packet. When \(t = 2\), suppose that \(s_2\) is scheduled; an uncoded flow-2 message symbol \(W_{2,1}\) is transmitted; and the packet is heard by and only by \(d_2\), see Fig. 4(b). When \(t = 3\), suppose that \(s_3\) is scheduled; an uncoded flow-1 symbol \(W_{1,3}\) is transmitted; and the packet is heard by and only by \(r\). When \(t = 4\), suppose that \(r\) is scheduled; \(r\) sends a linear combination \([W_{1,1} + 2W_{1,3}]\) of the two flow-1 packets it has received thus far; and the packet \([W_{1,1} + 2W_{1,3}]\) is heard by both \(d_1\) and \(d_2\).

We now describe the six knowledge spaces \(S_r\) to \(T_{d_2}\) in the end of \(t = 4\). By Figs. 4(a) and 4(d), \(d_2\) has received two flow-1 packets \(W_{1,1}\) and \(W_{1,1} + 2W_{1,3}\), one from \(s_1\) and one from \(r\). Therefore, by the end of \(t = 4\), the flow-1 knowledge space at \(d_2\) becomes \(S_{d_2}(4) = \text{span}\{(1, 0, 0), (1, 0, 2)\}\). Also, neither \(r\) nor \(d_1\) has received any flow-2 packets by the end

\(^7\)The construction of \(T_{d_1}\) (resp. \(T_{d_2}\)) follows the construction of \(S_{d_2}\) (resp. \(S_{d_4}\)).

\(^8\)When the relay \(r\) sends a linear combination of both flow-1 and -2 packets.
Flow-2 the inclusion space/set the following, we will introduce a new class of net-
span Policy span that could be derived similarly and they are summarized in Table II.

The above definitions also lead to the following self-explanatory lemma. 

**Lemma 1:** The two destinations $d_1$ and $d_2$ can decode the desired message symbols $W_1$ and $W_2$, respectively, if and only if by the end of time $n$

$$S_{d_1}(n) = \Omega_1$$

and $T_{d_2}(n) = \Omega_2$.

For simplicity, we use $S_i(t)$ and $T_i(t)$ to denote the knowledge space $S_i(t)$ and $T_i(t)$ for $i = 1, 2$. We also omit the input argument "(t)" if the time index is clear from the context. To conclude this subsection, we introduce the notation of the sum space: $(A \oplus B) \equiv \text{span}\{v : \forall v \in A \cup B\}$. Notice that $A \oplus B$ and $A \cup B$ are different. For example, suppose we consider a 2-dimensional linear space with $\text{GF}(3)$ with two linear subspaces $A = \text{span}\{(1, 0)\}$ and $B = \text{span}\{(1, 1)\}$. Then $A \cup B = \{(0, 0), (1, 0), (2, 0), (1, 1), (2, 2)\}$ is not a linear subspace anymore, but $A \oplus B = \text{span}\{(1, 0), (1, 1)\} = \{(0, 0), (1, 0), (2, 0), (1, 1), (2, 2), (2, 1), (1, 2), (0, 1), (0, 2)\}$ is a linear subspace. By simple algebra, we have the following lemma.

**Lemma 2:** For any two linear subspaces $A$ and $B$ in $\Omega$, the following equality always holds.

$$\dim(A \oplus B) = \dim(A) + \dim(B) - \dim(A \cap B).$$

### TABLE II

**The resulting knowledge spaces at the end of Example 1**

<table>
<thead>
<tr>
<th></th>
<th>Flow-1</th>
<th>Flow-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{d_1}(4)$</td>
<td>$\text{span}{(1, 0, 2)}$</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>$S_{d_2}(4)$</td>
<td>$\text{span}{(1, 0, 0), (1, 0, 2)}$</td>
<td>$T_{d_2}(4)$ $\text{span}{(1, 0)}$</td>
</tr>
<tr>
<td>$S_{r}(4)$</td>
<td>$\text{span}{(1, 0, 0), (0, 0, 1)}$</td>
<td>$T_{r}(4)$ $(0, 0)$</td>
</tr>
</tbody>
</table>

of $t = 4$. Therefore, $T_r$ and $T_{d_1}$, the flow-2 knowledge spaces at $r$ and $d_1$, respectively, contain only the zero element. The other knowledge spaces $S_{d_1}$, $S_r$, and $T_{d_2}$ in the end of $t = 4$ can be derived similarly and they are summarized in Table II.

The above example also lead to the following self-explanatory lemma.

**Lemma 2:** For any two linear subspaces $A$ and $B$ in $\Omega$, the following equality always holds.

$$\dim(A \oplus B) = \dim(A) + \dim(B) - \dim(A \cap B).$$

### B. An Instance of The SBLNC Schemes

In the following, we will introduce a new class of network codes, named as the Space-Based Linear Network Code (SBLNC). An SBLNC scheme contains a finite number of policies. Each policy $\Gamma$ contains a linear subspace $A^{(\Gamma)}$, named as the inclusion space/set, and a finite collection of linear subspaces $B_{l}^{(\Gamma)}$ for $l = 1$ to $L^{(\Gamma)}$ named as the exclusion spaces/sets. For each time slot $t$, the SBLNC chooses one of the specified policies and uses it to generate the coded packet. For example, say node $s$ is scheduled for transmission and we decide to choose a policy $\Gamma$ for encoding. Then $s$ will first choose arbitrarily a coding vector $v^{(\Gamma)}$ from the set $A^{(\Gamma)} \setminus \left( \bigcup_{l=1}^{L^{(\Gamma)}} B_{l}^{(\Gamma)} \right)$, and then transmit a linearly encoded packet $X = v^{(\Gamma)} W_{1}^{T}$. That is, the coding vector must be in the inclusion set $A^{(\Gamma)}$ but not in any of the exclusion sets $B_{l}^{(\Gamma)}$.

Obviously, a policy can be used/choosen only when the corresponding set $A^{(\Gamma)} \setminus \left( \bigcup_{l=1}^{L^{(\Gamma)}} B_{l}^{(\Gamma)} \right)$ is non-empty. For notational simplicity, we say a policy is feasible if the corresponding $A^{(\Gamma)} \setminus \left( \bigcup_{l=1}^{L^{(\Gamma)}} B_{l}^{(\Gamma)} \right)$ is non-empty.

For illustration, consider the following policy for node $s_1$, named as Policy $\Gamma_{s_1,0}$. When Policy $\Gamma_{s_1,0}$ is used/choosen, we let source node $s_1$ choose arbitrarily a coding vector $v^{(1)}$ from $\Omega_1 \setminus (S_1 \oplus S_2 \oplus S_r)$ and send the corresponding coded packet $X_{s_1} = v^{(1)} W_{1}^{T}$. That is, the inclusion set is $A^{v^{(1)}} = \Omega_1$ and the exclusion set is $B_{1}^{v^{(1)}} = S_1 \oplus S_2 \oplus S_r$.

Continue the example in Section IV-A for which the knowledge spaces are summarized in Table II. In the beginning of the policy of $t = 5$ (or equivalently in the end of $t = 4$), we have $A^{v^{(1)}} = \Omega_1$ and $B^{v^{(1)}} = \Omega_1$, so $\Gamma_{s_1,0}$ is used/choosen but not in any of the exclusion sets $B_{l}^{(\Gamma)}$.

In the following, we define 13 policies that are relevant to the proof of the achievability part of Propositions 1 and 3. There are 5 policies governing the coding operations at source $s_1$, which are named as Policy $\Gamma_{s_1,j}$ for $j = 0$ to 4. When Policy $\Gamma_{s_1,j}$ is used, $s_1$ sends $X_{s_1} = v^{(1)} W_{1}^{T}$ for some $v^{(1)}$. That is, source $s_1$ only mixes/encodes flow-1 packets. In the following, we describe how to choose the vector $v^{(1)}$ for each individual policy.

**Policy $\Gamma_{s_1,0}$:** Choose $v^{(1)}$ arbitrarily from

$$\Omega_1 \setminus (S_1 \oplus S_2 \oplus S_r).$$

(26)
§ Policy $\Gamma_{s_1,1}$: Choose $v^{(1)}$ arbitrarily from 
\[(S_2 \oplus S_r) \setminus ((S_1 \oplus S_r) \cup (S_1 \oplus S_2)).\] (27)

§ Policy $\Gamma_{s_1,2}$: Choose $v^{(1)}$ arbitrarily from 
\[S_1 \setminus (S_1 \oplus S_r).\] (28)

§ Policy $\Gamma_{s_1,3}$: Choose $v^{(1)}$ arbitrarily from 
\[S_r \setminus (S_1 \oplus (S_2 \cap S_r)).\] (29)

§ Policy $\Gamma_{s_1,4}$: Choose $v^{(1)}$ arbitrarily from 
\[(S_2 \cap S_r) \setminus S_1.\] (30)

Policy $\Gamma_{s_2,j}$, $j = 0$ to 4 are symmetric versions of Policy $\Gamma_{s_1,j}$ that concern source $s_2$ and mix/encode flow-2 packets instead. More explicitly, source $s_2$ sends $X_{s_2} = v^{(2)}W^T_2$ for which the coding vector $v^{(2)}$ is chosen according to the following specification.

§ Policy $\Gamma_{s_2,0}$: Choose $v^{(2)}$ arbitrarily from 
\[\Omega_2 \setminus (T_1 \oplus T_2 \oplus T_r).\] (31)

§ Policy $\Gamma_{s_2,1}$: Choose $v^{(2)}$ arbitrarily from 
\[(T_1 \oplus T_r) \setminus ((T_2 \oplus T_r) \cup (T_1 \oplus T_2)).\] (32)

§ Policy $\Gamma_{s_2,2}$: Choose $v^{(2)}$ arbitrarily from 
\[T_1 \setminus (T_2 \oplus T_r).\] (33)

§ Policy $\Gamma_{s_2,3}$: Choose $v^{(2)}$ arbitrarily from 
\[T_r \setminus (T_2 \oplus (T_1 \cap T_r)).\] (34)

§ Policy $\Gamma_{s_2,4}$: Choose $v^{(2)}$ arbitrarily from 
\[(T_1 \cap T_r) \setminus T_2.\] (35)

There are 3 policies $\Gamma_{r,j}$, $j = 1, 2, 3$, governing the coding operations at the relay $r$, which are described as follows.

§ Policy $\Gamma_{r,1}$: The relay $r$ chooses arbitrarily a vector $v^{(1)}$ from 
\[S_r \setminus ((S_1 \oplus S_2) \oplus S_1)\] (36) and sends an intra-flow-coded flow-1 packet $X_r = v^{(1)}W^T_1$.

§ Policy $\Gamma_{r,2}$: The relay $r$ chooses arbitrarily a vector $v^{(2)}$ from 
\[T_r \setminus ((T_r \cap T_1) \oplus T_2)\] (37) and sends an intra-flow-coded flow-2 packet $X_r = v^{(2)}W^T_2$.

§ Policy $\Gamma_{r,3}$ is for the relay node $r$ to send an interflow-coded packet $X_r = (v^{(1)}W^T_1 + v^{(2)}W^T_2)$, with $v^{(1)}$ and $v^{(2)}$ chosen as follows: If $(S_2 \cap S_r) \setminus S_1$ is non-empty, choose $v^{(1)}$ arbitrarily from 
\[(S_2 \cap S_r) \setminus S_1,\] (38) otherwise choose $v^{(1)} = 0$, a zero vector. If $(T_1 \cap T_r) \setminus T_2$ is non-empty, choose $v^{(2)}$ arbitrarily from 
\[(T_1 \cap T_r) \setminus T_2,\] (39) otherwise choose $v^{(2)} = 0$.

Continue from Example 1 in Section IV-A with the knowledge spaces in the end of $t = 4$ described in Table II. Consider Policy $\Gamma_{s_1,3}$ as defined in (29). Since $S_2 \cap S_r = S_r$ in the end of $t = 4$, we have $S_r \setminus (S_1 \oplus (S_2 \cap S_r)) \subseteq S_r \setminus (S_2 \cap S_r) = \emptyset$ being an empty set. Thus, in contrast with the fact that Policy $\Gamma_{s_1,0}$ is feasible in the beginning of $t = 5$ as shown in our previous discussion, Policy $\Gamma_{s_1,3}$ is infeasible in the beginning of $t = 5$.

One can repeat the above analysis and verify that out of all 13 policies, only 4 of them are feasible in the beginning of $t = 5$, which are $\Gamma_{s_1,0}, \Gamma_{s_1,4}, \Gamma_{s_2,0}$, and $\Gamma_{r,3}$. The network code designer can thus apply one of the four policies in $t = 5$.

Suppose the network designer chooses policy $\Gamma_{s_1,0}$ for $t = 5$ and sends a flow-1 coded packet with the coding vector being $v^{(1)} = (2, 1, 0)$. Also suppose that the packet is received by $r$ but by neither $d_1$ nor $d_2$. Then in the end of time $t = 5$, the knowledge space $S_r$ evolves from the original span$\{(1, 0, 0), (0, 0, 1)\}$ to the new space span$\{(1, 0, 0), (0, 0, 1), (2, 1, 0)\}$. We now notice that the Policy $\Gamma_{s_1,0}$ is no longer feasible since with the new $S_r$, the exclusion space of $\Gamma_{s_1,0}$ becomes $S_1 \oplus S_2 \oplus S_r = \text{span}\{(1, 0, 0), (0, 0, 1), (2, 1, 0)\}$ and $T_1 \setminus (S_1 \oplus S_2 \oplus S_r)$ is now empty. On the other hand, the new $S_r$ also lets some previously infeasible policies become feasible. For example, consider Policy $\Gamma_{s_1,3}$. With the new $S_r$, we have $S_r = \text{span}\{(1, 0, 0), (0, 0, 1), (2, 1, 0)\}$ and $S_1 \cap (S_2 \cap S_r) = \text{span}\{(1, 0, 0), (1, 2, 0)\}$. Therefore, $S_r \setminus (S_1 \oplus (S_2 \cap S_r)) \neq \emptyset$. Policy $\Gamma_{s_1,3}$ is thus feasible and can be used for transmission in $t = 6$. With similar analysis, one can verify that in the beginning of $t = 6$, we have 5 feasible policies: $\Gamma_{s_1,3}, \Gamma_{s_1,4}, \Gamma_{s_2,0}, \Gamma_{r,1}$, and $\Gamma_{r,3}$. This example shows that due to the evolution of the knowledge spaces over time, each coding policy may become feasible or infeasible depending on the reception status until the present. Focusing on coding policies frees us from designing the value of each coordinate of the coding vector $v^{(i)}$ (resp. $v^{(2)}$). Instead, we only need to choose $v^{(i)}$ (resp. $v^{(2)}$) from one of the policies that are currently feasible, which significantly simplifies the corresponding analysis.

C. The Intuition Behind the Proposed SBLNC Policies

In Section V, we will prove that the proposed SBLNC scheme can achieve the capacity in Proposition 1 and the inner bound in Proposition 3 when we carefully decide which of the 13 policies to apply for each time instant. We conclude Section IV by discussing the intuition behind the proposed 13 policies. We first consider the relay policies $\Gamma_{r,1}$ to $\Gamma_{r,3}$ due to its conceptual simplicity. We then discuss the source policies $\Gamma_{s_i,0}$ to $\Gamma_{s_i,4}$.

1) The Relay Policies: We first notice that for all relay policies $\Gamma_{r,1}, \Gamma_{r,2}$, and $\Gamma_{r,3}$, the corresponding inclusion space is either a subspace of $S_r$ or a subspace of $T_r$. The reason is that for node $r$ to send a coded packet, the encoded packet must already be in $S_r$ or $T_r$, the knowledge spaces of $r$. As a result, the transmitted vector $v^{(1)}$ (or $v^{(2)}$) must be drawn from a subset of $S_r$ (or $T_r$).

It is clear that a good network code should try to serve two flows simultaneously in order to maximize the throughput. We
now focus on Policy \( \Gamma_{r,3} \). First notice that by (39), \( \mathbf{v}^{(2)} \) is drawn from \((T_1 \cap T_r)\). This means that the value of \( \mathbf{v}^{(2)} \mathbf{W}_2^T \) is already known by destination \( d_1 \) since \( \mathbf{v}^{(2)} \) is in the flow-2 knowledge space \( T_1 \) at \( d_1 \). Hence whenever \( d_1 \) receives the packet \( X_r(t) = \mathbf{v}^{(1)} \mathbf{W}_1^T + \mathbf{v}^{(2)} \mathbf{W}_2^T \), it can extract its desired information and recover \( \mathbf{v}^{(1)} \mathbf{W}_1^T \) by substracting \( \mathbf{v}^{(2)} \mathbf{W}_2^T \). We then note that Policy \( \Gamma_{r,3} \) ensures that whenever (38) is not empty the selected \( \mathbf{v}^{(1)} \) is not in \( S_1 \), the flow-1 knowledge space at \( d_1 \). Hence upon the reception of such a coded packet, \( \dim(S_1) \) will increase by one. By Lemma 1 destination \( d_1 \) is one step closer to fully decode its desired message \( W_1 \). Symmetrically, by (38) \( d_2 \) has already known the value of \( \mathbf{v}^{(1)} \mathbf{W}_1^T \) and thus \( d_2 \) can compute the value of \( \mathbf{v}^{(2)} \mathbf{W}_2^T \) upon the reception of the inter-flow coded packet generated by Policy \( \Gamma_{r,3} \). Since \( \mathbf{v}^{(2)} \) is not in \( T_2 \), \( d_2 \) can decode one extra linear combination of flow-2 packets. Policy \( \Gamma_{r,3} \) thus serves both \( d_1 \) and \( d_2 \) simultaneously.

Although Policy \( \Gamma_{r,3} \) can serve both destinations simultaneously, there is a limit on how much information can be sent by \( \Gamma_{r,3} \). That is, if we use only Policy \( \Gamma_{r,3} \) and nothing else, the information that can be received by \( d_1 \) through Policy \( \Gamma_{r,3} \) is at most \((S_1 \cap S_2)\) since all \( \mathbf{v}^{(1)} \) are drawn from \((S_r \cap S_2)\). The largest flow-1 knowledge space that \( d_1 \) can possibly attain is thus \( S_1 \oplus (S_r \cap S_2) \), where \( S_1 \) represents the flow-1 information that \( d_1 \) has accumulated by overhearing the transmission directly from its two-hop neighbor \( s_1 \), and \((S_r \cap S_2)\) represents the information that can be conveyed by \( \Gamma_{r,3} \). Note that it is possible that \( S_r \) is not a subspace of \( S_1 \oplus (S_r \cap S_2) \), which means that relay \( r \) still possesses some flow-1 information that cannot be conveyed to \( d_1 \) by \( \Gamma_{r,3} \) alone. \( \Gamma_{r,3} \) is devised to address this problem. That is, the \( \mathbf{v}^{(1)} \) vector chosen from (36) is (i) from the knowledge space of \( r \), and (ii) not in \( S_1 \oplus (S_r \cap S_2) \), the largest flow-1 knowledge space that \( d_1 \) can attain when using exclusively Policy \( \Gamma_{r,3} \). Such \( \mathbf{v}^{(1)} \) vector thus represents an information packet that is complementary to the inter-flow-coded Policy \( \Gamma_{r,3} \).

2) The Source Policies: Here without loss of generality we focus on source-1 policies.

Before explaining the source policies (26)–(30), we first discuss several network coding goals for the transmission of \( s_1 \). The highest priority is to enlarge \( S_1 \) and \( S_2 \) such that
\[
(S_1 \oplus S_2) = \Omega_1
\]
at the end of the source-1 transmission since \( s_1 \rightarrow r \rightarrow d_1 \) and \( s_1 \rightarrow d_3 \) are the only two routes from \( s_1 \) to \( d_1 \). By the cut-set bound, we must achieve
\[
(S_1 \oplus S_r) = \Omega_1
\]
at the end of the source-1 transmission otherwise \( d_1 \) cannot decode the complete flow-1 messages. Another priority is to maximize \( \dim(S_1 \oplus (S_2 \cap S_r)) \). As discussed in the previous paragraph, the largest amount of information that can be transmitted to \( d_1 \) through inter-session coded messages is \( \dim(S_1 \oplus (S_2 \cap S_r)) \). Therefore, it is important to maximize the inter-session coding benefits by maximizing \( \dim(S_1 \oplus (S_2 \cap S_r)) \) during the transmission of \( s_1 \) so that the relay \( r \) can harvest the largest amount of inter-session coding benefits during Policy \( \Gamma_{r,3} \). Another priority is to ensure that destination \( d_1 \) overhears directly from \( s_1 \) as much information as possible since that information heard directly by \( d_1 \) does not need to be sent by relay \( r \) anymore. This is equivalent to maximize \( \dim(S_1) \).

With the above three goals in mind, we now discuss the intuition of each source policy. Policy \( \Gamma_{s_1,j} \), \( j = 0, 1, 2 \), are designed to maximize \( \dim(S_1 \oplus S_r) \) since \( S_1 \oplus S_r \) is a subset of the exclusion sets for these policies. Therefore, every time one of \( d_1 \) and \( r \) receives a packet encoded by Policy \( \Gamma_{s_1,j} \), \( j = 0, 1, 2 \), the term \( \dim(S_1 \oplus S_r) \) will increase by one. Policy \( \Gamma_{s_1,j} \), \( j = 0 \) to 3, are designed to maximize \( \dim(S_1 \oplus (S_2 \cap S_r)) \) since \( S_1 \oplus (S_2 \cap S_r) \) is a subset of the exclusion sets for all these policies. Policies \( \Gamma_{s_1,j} \), \( j = 0 \) to 4, are designed to maximize \( \dim(S_1) \) since \( S_1 \) is a subset of the exclusion sets for all these policies. As one can see that the five \( s_1 \)-policies aim at simultaneously achieving the three goals.

To explain the heuristics why we choose the specific inclusion and exclusion sets for these policies, we use Policy \( \Gamma_{s_1,1} \) as an example. To that end, we notice that by Lemma 2, we have
\[
\begin{align*}
\dim(S_1 \oplus (S_2 \cap S_r)) &= \dim(S_1) + \dim(S_2) - \dim(S_1 \cap S_2 \cap S_r) \\
&= \dim(S_1) + \dim(S_2) + \dim(S_r) - \dim(S_2 \oplus S_r) - \dim(S_1 \cap S_2 \cap S_r).
\end{align*}
\]
(40)
Since \( S_2 \oplus S_r \) is the inclusion set for Policy \( \Gamma_{s_1,1} \); and \( S_1 \), \( S_2 \), and \( S_r \) are subsets of the exclusion sets for Policy \( \Gamma_{s_1,1} \), every time any one of the \{\( d_1, d_2, r \)\} nodes receives a policy-\( \Gamma_{s_1,1} \) packet, at least one of the three terms \( \dim(S_1) \), \( \dim(S_2) \), and \( \dim(S_r) \) will increase and the term \( \dim(S_2 \oplus S_r) \) remains unchanged. Assuming that \( \dim(S_1 \oplus S_r) \) does not change too much\(^9\), Policy \( \Gamma_{s_1,1} \) increases \( \dim(S_1 \oplus (S_2 \cap S_r)) \) in a very efficient way since all three positive terms in (40) can have a good chance of increase while one negative term in (40) remains constant and the other negative term in (40) only increases slightly.

To summarize, the proposed source-1 policies aim at simultaneously maximizing \( \dim(S_1 \oplus S_r) \), \( \dim(S_1 \oplus (S_2 \cap S_r)) \), and \( \dim(S_1) \). The specific design of the inclusion and exclusions sets is to maximize the above three different terms in an efficient way. Detailed description about how the three terms increase is relegated to the throughput analysis in Section V-A.

V. CAPACITY APPROACHING CODING SCHEME

In this section, we will first prove the capacity inner bound Proposition 3 for the 2-flow wireless butterfly network setting considering both the COPE principle and opportunistic routing. We will then prove that the inner bound coincides with the capacity characterization in Proposition 1 when considering only the COPE principle.

A. Achieving The Inner Bound of Proposition 3

We prove Proposition 3 by properly scheduling the 13 policies described in Section IV-B.

Consider any \( t_{s_1}, t_{s_2}, t_r, \omega^i_k, i \in \{1, 2\} \) and \( k \in \{0, 1, 2, 3, 4\} \), \( \omega^i_{r,c} \), and \( \omega^i_{r,c}, k = 1, 2 \), satisfying the inequalities (14) to (25) in Proposition 3. For any \( \epsilon > 0 \), we can always construct another set of \( t' \) and \( \omega' \) variables such that the new
\(^9\)\( \dim(S_1 \cap S_2 \cap S_r) \) usually changes only slightly since it is relatively difficult for all nodes to acquire the same common information.
we have equality in (41) instead of (25). Based on the above observation, we will assume that the given \( t_{s_1}, t_{s_2}, t_r, \omega_{s_1}^k, \omega_{s_1}^{k,N}, \) and \( \omega_{r,C}^k \) satisfy (14) to (24) with strict inequality. In the following, we will construct an SBLNC scheme such that the scheme “properly terminates” within the allocated \( n \) time slots with close-to-one probability and after the SBLNC scheme stops, each \( d_i \) has received at least \( n(R_i - \epsilon) \) number of its desired information packets.

We construct the SBLNC scheme as follows. We first schedule the \( s_1 \)-policies sequentially from \( \Gamma_{s_1,0} \) to \( \Gamma_{s_1,4} \). Each policy \( \Gamma_{s_1,k} \) lasts for \( n \cdot \omega_{s_1}^k \) time slots. After finishing \( \Gamma_{s_1,k} \) we move on to Policy \( \Gamma_{s_1,k+1} \) until finishing all 5 \( s_1 \)-policies. After finishing the \( s_1 \)-policies, we move on to the \( s_2 \)-policies. Again, we choose the \( s_2 \)-policies sequentially from \( \Gamma_{s_2,0} \) to \( \Gamma_{s_2,4} \) and each policy lasts for \( n \cdot \omega_{s_2}^k \) time slots. After the \( s_2 \)-policies, we schedule the \( r \)-policies sequentially from \( \Gamma_{r,1} \) to \( \Gamma_{r,3} \). Policies \( \Gamma_{r,1} \) and \( \Gamma_{r,2} \) last for \( n \cdot \omega_{r,1}^k \) and \( n \cdot \omega_{r,2}^k \) time slots, respectively. Policy \( \Gamma_{r,3} \) lasts for \( n \cdot \max\{\omega_{r,1}^k, \omega_{r,2}^k\} \) time slots. Feedback is critical for the SBLNC scheme as it is used to decide the evolution of the knowledge spaces \( S_1, S_2, \ldots, T_r \), which in turn decides the sets in (26)–(39).

To prove the correctness of the above construction, we need to show that the following two statements hold with close-to-one probability: (i) During each time slot, it is always possible to construct the desired coding vectors \( v^{(1)} \) (or \( v^{(2)} \)). That is, we never schedule an infeasible policy throughout the operation; (ii) Destination \( d_1 \) can decode \( n(R_1 - \epsilon) \) of the desired information packets when the scheme terminates. In addition to the above two statements, we will also prove that (iii) during the first \( n \cdot \omega_{r,C}^k \) (resp. \( n \cdot \omega_{r,C}^k \)) time slots of scheduling \( \Gamma_{r,3} \), the computed flow-1 vector \( v^{(1)} \) (resp. flow-2 vector \( v^{(2)} \)) is not zero with close-to-one probability.

We first prove (ii) while assuming both (i) and (iii) are true. We notice that all the exclusion spaces of policies \( \Gamma_{s_1,0} \) to \( \Gamma_{s_1,4} \), and \( \Gamma_{r,1} \) contain \( S_1 \) as a subset. As a result, all those packets carry some new flow-1 information that has not yet been received by \( d_1 \). If \( d_1 \) receives any of those packets, the dimension of \( S_1 \) will increase by one. Similarly, during the first \( n \omega_{r,C}^k \) time slots of Policy \( \Gamma_{r,3} \), the computed flow-1 vector does not belong to \( S_1 \), see (38). As a result, if \( d_1 \) receives any of those packets, the dimension of \( S_1 \) will increase by one. From the above reasoning, the expected value of \( \dim(S_1) \) in the end of the SBLNC scheme must satisfy

\[
\mathbb{E}\{\dim(S_1)\} = p_1(d_1) \left( \sum_{k=0}^{4} n \omega_{s_1}^k \right) + p_r(d_1) \left( n \omega_{s_1}^{k,N} + n \omega_{r,C}^k \right) > n(R_1 - \epsilon)
\]  

(42)
feasible if (27) is non-empty, which is equivalent to having

\[ q \dim(S_2 + S_r) = |S_2 + S_r| \]

\[ > \dim(S_2 + S_r) \cap ((S_1 + S_r) \cup (S_1 + S_2)) \]

\[ = \dim((S_2 + S_r) \cap (S_1 + S_r)) \cup ((S_2 + S_r) \cap (S_1 + S_2)) \]

\[ \Leftrightarrow \dim(S_2 + S_r) \]

\[ > \max \{ \dim((S_2 + S_r) \cap (S_1 + S_r)), \dim((S_2 + S_r) \cap (S_1 + S_2)) \} \]

\[ \text{(47)} \]

where “\( \Leftrightarrow \)" holds assuming the underlying finite field GF(q) satisfying \( q \geq 2 \). The reason is as follows. Let us temporarily define \( A = S_2 + S_r, B = S_1 + S_r, \) and \( C = S_1 + S_2 \). Then we have

\[ A \setminus (B \cup C) \neq \emptyset \Leftrightarrow |A| - |(A \cap B) \cup (A \cap C)| > 0. \]

\[ \text{(48)} \]

Also, we have the following inequality

\[ q \max \{ \dim(A \cap B), \dim(A \cap C) \} \leq \dim(A \cap B) + \dim(A \cap C) - 1 \]

\[ \text{(49)} \]

where the last inequality follows from the idea of the union bound and the observation that the all-zero vector is always in both \( A \cap B \) and \( A \cap C \). Based on (49), we then have

\[ (48) \Leftrightarrow \dim(A) > \max \{ \dim(A \cap B), \dim(A \cap C) \}, \]

which leads to (47).

When we choose Policy \( \Gamma_{s_1,1} \) as our coding policy, the coding vector \( v^{(1)} \) is chosen from (27). Therefore, \( v^{(1)} \) must belong to the inclusion space \( S_2 + S_r \), which implies that no matter how many nodes in \( \{ d_1, d_2, r \} \) receive the packet, \( \dim(S_2 + S_r) \) remains the same. Also note that similar to the case of \( \Gamma_{s_1,0} \), \( \dim((S_2 + S_r) \cap (S_1 + S_r)) \) and \( \dim((S_2 + S_r) \cap (S_1 + S_2)) \) increase monotonically over time. As a result, if we can prove that (47) holds in the end of the duration of Policy \( \Gamma_{s_1,1} \), then throughout the entire duration of \( \Gamma_{s_1,1} \), we can always find some \( v^{(1)} \) belong to (27). The remaining task is thus to quantify the three different terms \( \dim(S_2 + S_r), \dim((S_2 + S_r) \cap (S_1 + S_r)), \) and \( \dim((S_2 + S_r) \cap (S_1 + S_2)) \) at the end of (the duration of) \( \Gamma_{s_1,1} \). All the following discussions hold with close-to-one probability when focusing on the first order analysis of \( n \).

We will first decide the value of \( \dim(S_2 + S_r) \). We know that \( \dim(S_2 + S_r) \) remains the same during Policy \( \Gamma_{s_1,1} \). Therefore, the value of \( \dim(S_2 + S_r) \) is decided by how much it increases during \( \Gamma_{s_1,0} \). Since any \( v^{(1)} \) in Policy \( \Gamma_{s_1,0} \) does not belong to \( S_2 + S_r \) (see (26)), every time one of \( d_1 \) and \( r \) receives a packet of \( \Gamma_{s_1,0} \), \( \dim(S_2 + S_r) \) will increase by one. As a result, in the end of \( \Gamma_{s_1,1} \) we have

\[ E \{ \dim(S_2 + S_r) \} = n \omega_{s_1}^0 \cdot p_1(d_2, r) + n \omega_{s_1}^1 \cdot 0. \]

\[ \text{(50)} \]

We now consider the first term \( \dim((S_2 + S_r) \cap (S_1 + S_r)) \) in the max operation in (47). By Lemma 2, we can rewrite

\[ \dim((S_1 + S_r) \cap (S_2 + S_r)) \]

\[ = \dim(S_2 + S_r) + \dim(S_1 + S_r) - \dim(S_1 + S_2 + S_r). \]

\[ \text{(51)} \]

The value of \( \dim(S_2 + S_r) \) is quantified in (50). Since any \( v^{(1)} \) in Policy \( \Gamma_{s_1,0} \) does not belong to \( S_1 + S_r \) (see (26)) and any \( v^{(1)} \) in Policy \( \Gamma_{s_1,1} \) does not belong to \( S_1 + S_r \) either (see (27)), every time one of \( d_1 \) and \( r \) receives a packet of \( \Gamma_{s_1,0} \) or \( \Gamma_{s_1,1} \), \( \dim(S_1 + S_r) \) will increase by one. In the end of \( \Gamma_{s_1,1} \) we thus have

\[ E \{ \dim(S_1 + S_r) \} = n \omega_{s_1}^0 \cdot p_1(d_1, r) + n \omega_{s_1}^1 \cdot p_1(d_1, r). \]

\[ \text{(52)} \]

Similarly, since any \( v^{(1)} \) in Policy \( \Gamma_{s_1,0} \) does not belong to \( S_1 + S_2 + S_r \) (see (26)) and any \( v^{(1)} \) in Policy \( \Gamma_{s_1,1} \) belongs to \( S_1 + S_2 + S_r \) (see (27)), every time one of \( d_1, d_2, \) and \( r \) receives a packet of \( \Gamma_{s_1,0} \), \( \dim(S_1 + S_2 + S_r) \) will increase by one. In the end of \( \Gamma_{s_1,1} \) we thus have

\[ E \{ \dim(S_1 + S_2 + S_r) \} = n \omega_{s_1}^0 \cdot p_1(d_1, d_2, r) + n \omega_{s_1}^1 \cdot 0. \]

\[ \text{(53)} \]

By (50), (51), (52), and (53), we can verify that (18) implies that \( \dim(S_2 + S_r) > \dim((S_2 + S_r) \cap (S_1 + S_r)) \) in the end of Policy \( \Gamma_{s_1,1} \). By swapping the roles of \( d_2 \) and \( r \), symmetric arguments can be used to prove that (19) implies \( \dim(S_2 + S_r) > \dim((S_2 + S_r) \cap (S_1 + S_2)) \) in the end of Policy \( \Gamma_{s_1,1} \). Therefore, \( \Gamma_{s_1,1} \) is feasible throughout its duration of \( n \omega_{s_1}^1 \) time slots.

Similar dimension-comparison arguments can be used to complete the proof of (i) and (iii). The remaining derivation repeats similar steps described above, and is relegated to Appendix B. The proof of Proposition 3 is thus complete.

### B. Capacity of the COPE Principle On 2-Flow Wireless Butterfly Networks with Broadcast PECs

In this subsection we will prove that the capacity outer bound in Proposition 2 and the capacity inner bound in Proposition 3 coincide when destination \( d_i \) cannot directly hear from source \( s_i \) for \( i = 1, 2 \), which prohibits the use of opportunistic routing. Proposition 1 thus describes the exact capacity region of the COPE principle on 2-flow wireless networks with broadcast PECs. Recall that in Proposition 2, the outer bound considers channel status feedback, dynamic scheduling, and possibly non-linear encoding functions. Meanwhile, the proposed achieving scheme in Section V-A uses only linear encoding functions and a sequential-order scheduling. By proving that the outer and inner bounds match, we have shown that the proposed linear encoding and sequential scheduling scheme in Section V-A is as good as any other non-linear encoding and dynamic scheduling scheme when the use of opportunistic routing is prohibited.

To complete the proof, we note that when \( p_i(d_i) = 0 \), for \( i = 1, 2 \), (17)–(25) of the inner bound in Proposition 3 is
reduced to the following forms.\(^\text{11}\)

\[
\begin{align*}
\omega_s^0 p_t(d_j, r) &\leq R_i, \\
\omega_s^1 p_t(r) &\leq \omega_s^0 p_t(d_j, \tau), \\
\omega_s^2 p_t(d_j) &\leq \omega_s^0 p_t(r d_j), \\
\omega_s^3 p_t(r) &\leq \omega_s^0 p_t(d_j, \tau) - \omega_s^1 p_t(r), \\
\omega_s^3 p_t(d_j) &\leq \omega_s^0 p_t(r d_j) - \omega_s^2 p_t(d_j), \\
\omega_{r,N}^t p_t(d_i, d_j) &\leq \omega_s^0 p_t(r d_j) - \omega_s^1 p_t(d_j), \\
\omega_{r,C}^t p_t(d_i) &\leq \omega_s^0 p_t(r j d_r) + \omega_s^1 p_t(r j), \\
\omega_{r,C}^t p_t(d_i) &\leq \omega_s^0 p_t(r j d_r) + \omega_s^1 p_t(r j) + \omega_{r,N}^t p_t(d_j d_i), \\
\omega_{r,C}^t p_t(d_i) &\leq \omega_s^0 p_t(r j d_r) + \omega_s^1 p_t(r j) + \omega_{r,N}^t p_t(d_j d_i) + \omega_{r,C}^t p_t(d_i)
\end{align*}
\]  

(54) (55) (56) (57) (58) (59) (60)

and (25) becomes

\[
p_t(d_i)(\omega_{r,N}^t + \omega_{r,C}^t) \geq R_i. 
\]  

(61)

The following lemma proves the tightness of the bounds when there is no 2-hop overhearing, i.e., \(p_t(d_i) = 0\) for \(i = 1, 2\).

**Lemma 3:** For any 5-tuple \((R_1, R_2, t_1, t_2, t_r)\) satisfying the capacity outer bound (6)–(9), we can always find 14 coexisting variables \(\omega_s^0, \omega_s^1, \omega_s^2, \omega_s^3, \omega_r^0, \omega_r^1, \omega_r^2, \omega_r^3, \omega_{r,N}^t, \omega_{r,C}^t\) for \(i = 1, 2\) and \(j = 0, 1, 2, 3, 4\), such that jointly the 5 + 14 = 19 variables satisfy (14), (15), (54) to (61).

**Proof:** Given any \((R_1, R_2, t_1, t_2, t_r)\) satisfying (6)–(9), we construct \(\{\omega_s^0, \omega_s^1, \omega_s^2, \omega_s^3, \omega_r^0, \omega_r^1, \omega_r^2, \omega_r^3, \omega_{r,N}^t, \omega_{r,C}^t : i \in \{1, 2\}, j \in \{0, 1, 2, 3, 4\}\) in the following way. For each pair \((i, j) = (1, 2)\) or \((2, 1)\), we define

\[
\begin{align*}
\omega_s^0 & = \frac{R_i}{p_t(d_j, r)}, \\
\omega_s^1 & = R_i \left( \min \left\{ \frac{1}{p_t(r)}, \frac{1}{p_t(d_j)} \right\} - \frac{1}{p_t(d_j, r)} \right), \\
\omega_s^2 & = R_i \left( \frac{1}{p_t(r)} - \frac{1}{p_t(d_j)} \right), \\
\omega_s^3 & = \min \left\{ R_i \left( \frac{1}{p_t(d_j)} - \frac{1}{p_t(r)} \right), t_s - \frac{R_i}{p_t(r)} \right\}, \\
\omega_s^4 & = 0, \\
\omega_{r,N}^t & = \frac{(R_i - t_s p_t(d_j))^+}{p_t(d_i, d_j)}, \\
\omega_{r,C}^t & = \frac{(R_i - t_s p_t(d_j))^+}{p_t(d_i, d_j)}.
\end{align*}
\]  

(62) (63) (64) (65) (66) (67) (68)

One can verify that the above assignment \(\{R_1, R_2, t_1, t_2, t_r, \omega_s^0, \omega_s^1, \omega_s^2, \omega_s^3, \omega_r^0, \omega_r^1, \omega_r^2, \omega_r^3, \omega_{r,N}^t, \omega_{r,C}^t : i \in \{1, 2\}, j \in \{0, 1, 2, 3, 4\}\) is always non-negative and satisfies (14), (15), (54) to (61). The detailed verification is relegated to Appendix C. The proof of Lemma 3 is thus complete.

**VI. Numerical Results**

In this section, we apply the capacity results to some numerically generated scenarios so that we can explicitly quantify the throughput/capacity improvement of the SBLNC scheme under the COPE principle with and without opportunistic routing. The detailed simulation setting is described as follows.

![Fig. 5. An instance of the 2-flow wireless butterfly network with broadcast PECs with the success probabilities being indicated next to the corresponding arrows: (a) The COPE principle only; and (b) The COPE principle plus opportunistic routing. We also assume that the success events between different node pairs are independent.](image)

![Fig. 6. The achievable regions of the scenario in Fig. 5(a) and Fig. 5(b). The solid lines indicate the achievable regions of the SBLNC scheme under the scenarios of COPE only (the orange line) or COPE plus OpR (the red line). The dash lines indicate the achievable regions of the existing result [3] under the scenarios of COPE only (the green line) or COPE plus OpR (the blue line). The dotted lines indicate the achievable regions of intra-session network coding only (random linear network coding) under the scenarios of OpR (the gray line) or not (the black line).](image)

Consider one particular channel parameter assignment of the 2-flow wireless butterfly network with broadcast PECs. Fig. 5(a) describes the transmission success probability between each node pair as the number next to the corresponding arrow without opportunistic routing. And Fig. 5(b) describes the same set of channel parameter assignment, except for that now we allow OpR. We also assume that the success events between different node pairs are independent. For example, when allowing opportunistic routing in Fig. 5(b), the probability that a packet sent by \(s_1\) is heard by \(d_1\) is \(p_t(d_1) = .2\) and the probability that a packet sent by \(r\) is received by \(d_2\) is \(p_t(d_2) = .6\). We then compute 6 different achievable regions and plot them in Fig. 6.

The solid lines in Fig. 6 represent the achievable regions\(^{12}\) of

\(^{11}\)Inequality (22) becomes trivial since the left-hand side of (22) becomes zero and the right-hand side of (22) is always non-negative.

\(^{12}\)Our main results provide a pair of outer and inner bounds for this capacity region. Since the gap between the inner and outer bounds is negligible (with relative gap less than 0.08%), we plot only the inner bound (the achievable rate) in Fig. 6.
place four nodes \((s_1, s_2, d_1, d_2)\) inside the unit circle. To simulate the need of the relay for each session pair, we force the placement of each pair to be in the opposite 90 degree area. That is, \(d_1\) must be located in the opposite 90 degree area of \(s_i\)’s location for \(i = 1, 2\). See Fig. 7(a) for illustration. Fig. 7(b) illustrates one realization of our random node placement.

We use the Euclidean distance \(D\) between any two nodes to decide the overhearing probability when a packet is transmitted. More explicitly, we use the Rayleigh model

\[
\text{Prob(success)} = \int_{T^*}^{\infty} \frac{2x}{\gamma} e^{-\frac{x^2}{2\gamma}} dx \quad \text{where } \gamma = \frac{1}{(4\pi)^2D^2},
\]

where \(\alpha\) is the path loss factor, and \(T^*\) is the decodable SNR threshold. To reflect the packet delivery ratio measured in practical environments, we choose \(\alpha = 2.5\) and \(T^* = 0.006\) so that the overhearing probability for a 1-hop neighbor is around 0.7–0.8 while overhearing probability for a 2-hop neighbor is around 0.2–0.3. If no direct overhearing is allowed, we simply hardwire the probability that \(d_i\) overhears \(s_i\) to be zero. We again assume that the success events between different node pairs are independent.

We consider three different fairness requirements: (a) No fairness requirement; (b) Proportional fairness; and (c) Min-cut-based fairness requirement. When there is no fairness constraint, we use a linear programming solver to find the largest sum rate \(R_1 + R_2\) that satisfies the capacity outer bound in Proposition 2, which is denoted by \(R_{\text{sum.outer}}\). Similarly, we find the largest sum rate \(R_1 + R_2\) that satisfies the capacity inner bound in Proposition 3 and denote it by \(R_{\text{sum.inner}}\). After computing the sum rates \(R_{\text{sum.outer}}\) and \(R_{\text{sum.inner}}\), we repeat the above experiment with different randomly chosen node placements for 10000 times. For the setting of (b) proportional fairness, we replace the sum rate objective function \(R_1 + R_2\) by the logarithmic objective function \(\log(R_1) + \log(R_2)\). We again compute the \(R_{\text{sum.outer}}\) and \(R_{\text{sum.inner}}\) using the new objective function. For the setting of (c) min-cut-based requirement, we impose an additional constraint \(R_i = \beta \min(p_i(d_i), r), p_i(d_i) + p_i(d_i))\) for \(i = 1, 2\) with a common \(\beta\), which enforces the individual rate \(R_i\) being proportional to the min-cut value from \(s_i\) to \(d_i\) assuming no other sessions are transmitting and \(s_i\) and \(r\) are scheduled with the same frequency. The results are summarized in Table III and Fig. 8.

Table III lists the sum-rate averaged over 10000 simulations. When opportunistic routing is allowed \((p_i(d_i) > 0)\), then the inner and outer bounds do not always meet. Therefore, for the entries with \(\text{SBLNC}\) and \(\text{OpR}\), the number on the left is the average of \(R_{\text{sum.outer}}\) while the number on the right is the average of \(R_{\text{sum.inner}}\). When there is no \(\text{OpR}\) \((p_i(d_i) = 0)\),

<table>
<thead>
<tr>
<th>Fairness Constraints</th>
<th>(\text{OpR})</th>
<th>(\text{SBLNC})</th>
<th>([3])</th>
<th>(\text{RLNC})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{No allowed})</td>
<td>0.6599/0.6594</td>
<td>0.6472</td>
<td>0.6180</td>
<td></td>
</tr>
<tr>
<td>(\text{not allowed})</td>
<td>0.4820</td>
<td>0.4779</td>
<td>0.4116</td>
<td></td>
</tr>
<tr>
<td>(\text{Proportional allowed})</td>
<td>0.6294/0.6286</td>
<td>0.6101</td>
<td>0.5484</td>
<td></td>
</tr>
<tr>
<td>(\text{not allowed})</td>
<td>0.47/0.75</td>
<td>0.47/0.75</td>
<td>0.3854</td>
<td></td>
</tr>
<tr>
<td>(\text{Min-cut allowed})</td>
<td>0.6037/0.6026</td>
<td>0.3892</td>
<td>0.5406</td>
<td></td>
</tr>
<tr>
<td>(\text{not allowed})</td>
<td>0.4671</td>
<td>0.4671</td>
<td>0.3856</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7. (a) The relative location of \((s_i, d_i)\). (b) Topology of two \((s_i, d_i)\) pairs.

Fig. 8. The cumulative distribution of the relative gap between the outer and the inner bounds with proportional fairness. The outer and the inner bounds are described in Propositions 2 and 3, respectively.

the SBLNC scheme with and without \(\text{OpR}\), respectively. The dash lines represent the achievable regions of the existing side-information-based feedback-free result [3] with and without \(\text{OpR}\). The dotted lines represent the achievable regions of intra-session network coding while performing time-sharing between different sessions. When \(\text{OpR}\) is allowed, the dotted line corresponds to the intra-session RLNC discussed in [29]. When \(\text{OpR}\) is not allowed, the dotted line is the scheduling-based capacity (stability) of the store-&-forward solutions.

As can be seen, when there is only one flow in the network (say \(R_1 = 0\)), then \(\text{OpR}\) (the gray dotted line) is optimal as was first established in [29]. However, when there are two coexisting flows (when both \(R_1\) and \(R_2 > 0\)), the COPE principle without \(\text{OpR}\) (the orange solid line) can sometimes outperform \(\text{OpR}\) due to the stronger overhearing between \(s_2 \to d_1\) and \(s_1 \to d_2\), \(p_2(d_1) = 0.4\) and \(p_1(d_2) = 0.5\), than the two-hop direct overhearing from \(s_1 \to d_1\) and \(s_2 \to d_2\), \(p_1(d_1) = 0.2\) and \(p_2(d_2) = 0.15\). On the other hand, the throughput can be further enhanced by the proposed joint COPE and OpR solution (the red solid line). Last but not least, the proposed SBLNC scheme (the solid lines) always outperforms all the existing schemes since it achieves/approaches the capacity region.

We are also interested in quantifying the average throughput benefits of COPE and \(\text{OpR}\) in a randomly placed network. To generate a typical XOR-in-the-air scenario, we first place the relay node in the center of a unit circle. Then we randomly simulate the need of the relay for each session pair, we force the placement of each pair to be in the opposite 90 degree area.
as was proven Section V-B, the sum-rate outer and inner bounds always coincide and hence only one number is shown in each corresponding entry. The achievable rates in [3] for different scenarios are also listed for comparison. The capacity of pure routing and pure OpR is provided in [29] and we also compute the capacity of the intra-session-coding-only RLNC-based achievability scheme in [29] for comparison. In Table III, we first note that in terms of the averaged throughput, the difference between the outer and the inner bounds is around 0.08%. Among all 10000 instances, the largest absolute difference is with $R_{\text{sum,outer}} = 0.6409$ and $R_{\text{sum,inner}} = 0.6375$. The proposed bounds thus effectively bracket the capacity when combining the COPE and the OpR principles. With the proportional fairness constraint, the proposed SBLNC scheme that combines the benefits of COPE and OpR provides roughly 14.6% throughput improvement when compared to the scheme using only intra-session RLNC and OpR without the inter-session COPE-based network coding operations. Again assuming proportional fairness, if we compare the SBLNC scheme that combines COPE and OpR (sum rate = 0.6286) with the benchmark store-&-forward scheme without inter-session coding and without OpR (sum rate = 0.3854), it shows that COPE and OpR can provide in average 63% throughput improvement over the traditional TCP/IP solutions. We also observe that the throughput improvement of COPE is greater with the benchmark store-&-forward scheme without inter-

VII. CONCLUSION

This paper has introduced a new network coding architecture, named as the “Space-Based Linear Network Code (SBLNC).” The SBLNC scheme has been used to find the exact capacity region of the COPE principle 2-flow wireless butterfly networks with broadcast PECs. The result has also been extended to bracket the capacity when combining the COPE principle and the concept of opportunistic routing. Numerical results show that the proposed outer and inner bounds effectively quantify the capacity for almost all practical scenarios.

APPENDIX

A. Proof of Proposition 2

For any joint scheduling and NC scheme, we choose $t_{s_i}$ (resp. $t_r$) as the normalized expected number of time slots for which $s_i$ (resp. $r$) is scheduled. Namely,

$$t_{s_i} = \frac{1}{n} \mathbb{E}\left\{ \sum_{\tau=1}^{n} 1_{\{\sigma(\tau)=s_i\}} \right\}, \quad t_r = \frac{1}{n} \mathbb{E}\left\{ \sum_{\tau=1}^{n} 1_{\{\sigma(\tau)=r\}} \right\}.$$

By definition, $t_{s_1}$, $t_{s_2}$, and $t_r$ must satisfy (10).

In the subsequent proofs, the logarithm is taken with base $q$. We prove (11) first. To that end, we notice that

$$I(W_1; W_1) \leq I(W_1; [Y_{\{s_1, s_2, r\}} \rightarrow d_1], Z_1^n)$$

$$= I(W_1; [Z_1^n]_1^1) + I(W_1; [Y_{\{s_1, s_2, r\}} \rightarrow d_1]|_1^n) [Z_1^n]_1^1$$

$$\leq I(W_1; [Y_{\{s_1, s_2\}} \rightarrow d_1, r]|_1^n) [Z_1^n]_1^1$$

$$= I(W_1; [Y_{s_1} \rightarrow d_1, r]|_1^n) [Z_1^n]_1^1$$

$$\leq H([Y_{s_1} \rightarrow (d_1, r)]|_1^n) [Z_1^n]_1^1$$

$$= \sum_{t=1}^{n} H(Y_{s_1} \rightarrow (d_1, r)(t)) [Z_1^n]_1^1, [Y_{s_1} \rightarrow (d_1, r)|_1^n]^{-1}$$

$$\leq \sum_{t=1}^{n} \mathbb{E}\left\{ 1\{Z_{s_1} \rightarrow d_1(t)=1 \text{ or } Z_{s_1} \rightarrow r(t)=1 \} \right\} \leq n t_{s_1} p_1(d_1, r).$$

where (69) follow from (4); (70) follows from the chain rule; (71) follows from (3), the data processing inequality, and the fact that $Z$ is independent of $W_1$; (72) follows from that conditioning on $Z$ (and $\sigma$ since $\sigma$ is a function of $Z$) $Y_{s_2} \rightarrow (d_1, r)$ is a deterministic function of $W_2$ and is thus independent of $W_1$; (73) follows from the chain rule; (74) follows from that only when $1\{Z_{s_1} \rightarrow d_1(t)=1 \text{ or } Z_{s_1} \rightarrow r(t)=1 \} = 1 \text{ and } 1\{\sigma(t)=s_1\} = 1$ will we have a non-zero entropy value $H(Y_{s_1} \rightarrow (d_1, r)|_1^n) [Z_1^n]_1^1, [Y_{s_1} \rightarrow (d_1, r)|_1^n]^{-1}$, and when $H(Y_{s_1} \rightarrow (d_1, r)|_1^n) [Z_1^n]_1^1, [Y_{s_1} \rightarrow (d_1, r)|_1^n]^{-1} > 0$, it is upper bounded by 1 since the base of the logarithm is $q$; (75) follows from Wald’s lemma.

On the other hand, Fano’s inequality gives us

$$I(W_1; W_1) \geq n R_1(1 - \epsilon) - H(\epsilon).$$

Combining (75) and (76), we have

$$R_1(1 - \epsilon) - \frac{H(\epsilon)}{n} \leq t_{s_1} p_1(d_1, r).$$

Letting $\epsilon \to 0$, (77) implies (11) for the case of $i = 1$. With symmetric arguments, we can derive (11) for $i = 2$.

We prove (13) by similar techniques as used in [5], [30]. Specifically, we create a new network from the original network by adding an auxiliary pipe that sends all information available at $d_2$ directly to $d_1$. Later we will show that even with the additional information, the achievable rates $R_1$ and $R_2$ are still upper bounded by (13). As a result, the achievable $R_1$ and $R_2$ for the original network must satisfy (13) as well. (12) is a symmetric version of (13).

With the additional information at $d_1$, the decoding function (see (4)) at $d_1$ for the new network becomes

$$\hat{W}_1 = f_{d_1}([Y_{\{s_1, s_2, r\}} \rightarrow (d_1, d_2)], Z_1^n).$$

For any $t \in [n]$, define

$$U(t) = \frac{1}{n} \mathbb{E}\left\{ \sum_{\tau=1}^{n} 1_{\{\sigma(\tau)=t\}} \right\}.$$
We then have

\[
\begin{align*}
\eta R_1 &= H(W_1 | W_2) \\
&\leq I(W_1; Y_{s_1, s_2, r} \rightarrow \{d_1, d_2\}, Z|_n^n) + n \epsilon_1 \\
&\leq I(W_1; Y_{s_1, s_2, r} \rightarrow \{d_1, d_2\}, W_2) + n \epsilon_1 \\
&= I(W_1; W_2 | Z|_1^n) + I(W_1; Y_{s_1, s_2, r} \rightarrow \{d_1, d_2\}, |W_2, Z|_1^n) + n \epsilon_1 \\
&= \sum_{t=1}^{n} I(W_1; Y_{s_1, s_2, r} \rightarrow \{d_1, d_2\}(t) | W_2, Z|_1^n) \\
&\quad - \frac{1}{n} H(Y_{r \rightarrow \{d_1, d_2\}}(q_t) | U(q_t), Z|_1^n) \\
&\quad \leq \frac{1}{n} \sum_{q_1=1}^{n} H(Y_{r \rightarrow \{d_1, d_2\}}(q_t) | U(q_t), Z|_1^n).
\end{align*}
\]

(80) follows from Fano’s inequality where \( \epsilon_1 \) goes to 0 when \( \epsilon \) goes to 0; (81) follows from the data processing inequality and (78); (82), (83), and (84) follow from the chain rule and the fact that the distribution of \( Z \) is independent of \( W_1 \) and \( W_2 \); (85) follows from the observation that the second term of the summation can be upper bounded by Wald’s lemma (similar to (75)) and \( Y_{s_2, r} \rightarrow \{d_1, d_2\}(t) \) is independent of \( W_1 \) given \( Z \) (similar to (72)); and (86) follows from the data processing inequality.

To continue, we define the time sharing random variable \( Q_t \in \{1, 2, \ldots, n\} \) with \( \text{Prob}(Q_t = i) = \frac{1}{n} \) for all \( i \in \{1, 2, \ldots, n\} \) and \( Q_t \) being independent of \( |Z|_1^n, W_1, \) and \( W_2 \). Since the mutual information is always non-negative, we can rewrite (86) as

\[
\begin{align*}
&\quad (R_1 - t_s p_1(d_1, d_2) - \epsilon_1)^+ \\
&\leq \sum_{q_1=1}^{n} \frac{1}{n} I(X_r(t); Y_{r \rightarrow \{d_1, d_2\}(t)} | U(t), |Z|_1^n) \\
&\leq \sum_{q_1=1}^{n} \frac{1}{n} H(Y_{r \rightarrow \{d_1, d_2\}(t)} | U(t), |Z|_1^n) \\
&= \sum_{q_1=1}^{n} \text{Prob}(Q_t = q_t) \cdot H(Y_{r \rightarrow \{d_1, d_2\}(q_t)} | U(q_t), |Z|_1^n, Q_t = q_t).
\end{align*}
\]

(87) follows from the definition of the mutual information; (88) follows from replacing the time index \( t \) by the time sharing random variable \( Q_t \) and the distribution of \( U(q_t) \) and \( Y_{r \rightarrow \{d_1, d_2\}(q_t)} \) does not depend on the future channel realization \( |Z|_{q_t+1}^n \).

We define three binary random variables \( \Theta_\sigma \triangleq 1_{\{Q_t = \sigma\}} \), \( \Theta_{q_t} \triangleq 1_{\{Q_t = q_t\}} \), and \( \Theta_{Z_{q_t}} \triangleq 1_{\{Q_t = q_t\}} \), which are functions of \( Q_t \) and \( |Z|_{q_t+1}^n \). Then we can rewrite (88) as the following.

\[
\begin{align*}
&\quad (R_1 - t_s p_1(d_1, d_2) - \epsilon_1)^+ \\
&\leq \sum_{q_1=1}^{n} \frac{1}{n} H(Y_{r \rightarrow \{d_1, d_2\}}(q_t) | U(q_t), |Z|_1^n) \\
&= \sum_{q_1=1}^{n} \frac{1}{n} \sum_{\forall q_1, |z|_1^n} \begin{aligned}
&\quad p_{U(q_t)}(u_1, |z|_1^n, \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}} | \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}}) \\
&\quad \cdot H(Y_{r \rightarrow \{d_1, d_2\}(q_t)} | U(q_t) = u_1, |Z|_1^n = |z|_1^n, \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}}) \\
&\quad \Theta_{\sigma} = \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}} = \Theta_{q_t} = \Theta_{Z_{q_t}}, \Theta_{Z_{q_t}} = \Theta_{Z_{q_t}}) \\
&= \sum_{q_1=1}^{n} \sum_{\forall q_1, |z|_1^n} \begin{aligned}
&\quad p_{U(q_t)}(u_1, |z|_1^n, \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}} | \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}}) \\
&\quad \cdot H(X_r(q_t) | U(q_t) = u_1, |Z|_1^n = |z|_1^n, \Theta_{\sigma} = 1, \Theta_{q_t} = \Theta_{Z_{q_t}} = \Theta_{Z_{q_t}}).
\end{aligned}
\end{aligned}
\]

(89) follows from the fact that \( \Theta_{\sigma} \) is a function of \( Q_t \) and \( |Z|_1^n \); (90) follows from the definition of the conditional entropy; and (91) follows from the fact that \( Y_{r \rightarrow \{d_1, d_2\}}(q_t) \) is not erasure only if \( \sigma(q_t) = r \) and at least one of \( Z_{r \rightarrow d_1} \) and \( Z_{r \rightarrow d_2} \) equals to one and furthermore \( Y_{r \rightarrow \{d_1, d_2\}}(q_t) = X_r(q_t) \) under such a condition, where we use \( p(u, |z|_1^n, 1, \Theta_{q_t}, \Theta_{Z_{q_t}}) \) as the shorthand of \( p_{U(q_t)}(u, |z|_1^n, \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}} | \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}}) \).

We can further simplify (91) by the following steps. We first note that conditioning on \( U(q_t) = u_1, |Z|_1^n = |z|_1^n \), and \( \Theta_{\sigma} = 1 \), the random variable \( X_r(q_t) \) is independent of \( |Z|_1^n \), \( \Theta_{Z_{q_t}}, \) and \( \Theta_{Z_{q_t}} \). Notice that \( |Z|_1^n \) is a subset of \( U(q_t) \). Therefore, we have

\[
\begin{align*}
&\quad H(X_r(q_t) | U(q_t) = u_1, |Z|_1^n = |z|_1^n, \Theta_{\sigma} = 1, \Theta_{q_t} = \Theta_{Z_{q_t}} = \Theta_{Z_{q_t}}) \\
&= H(X_r(q_t) | U(q_t) = u_1, \Theta_{\sigma} = 1).
\end{align*}
\]

(92) Also the joint probability can be rewritten as

\[
\begin{align*}
&\quad \sum_{\forall q_1, |z|_1^n} \begin{aligned}
&\quad p_{U(q_t)}(u_1, |z|_1^n, \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}} | \Theta_{\sigma, \Theta_{q_t}, \Theta_{Z_{q_t}}}) \\
&\quad \cdot H(X_r(q_t) | U(q_t) = u_1, |z|_1^n = |z|_1^n, \Theta_{\sigma} = 1, \Theta_{q_t} = \Theta_{Z_{q_t}} = \Theta_{Z_{q_t}}).
\end{aligned}
\end{align*}
\]

(93)
where (93) follows from the basic probability definition, and (94) follows from the assumption that the channel is memoryless.

(92) and (94) help us rewrite (91) as

\[
(91) = t_p \cdot p_r(d_1, d_2) \cdot \sum_{q_t=1}^n \frac{1}{n} \sum_{u} p(u, 1) \cdot H(X_r(q_t)|u, 1),
\]

where \( p(u, 1) \) and \( H(X_r(q_t)|u, 1) \) are the shorthand for \( p_U(u), \Theta_r(u, 1) \) and \( H(X_r(q_t)|U(q_t) = u, \Theta_r = 1) \), respectively.

We now focus on flow 2. By Fano’s inequality, for some \( \epsilon_2 > 0 \) that goes to 0 as \( \epsilon \to 0 \), with similar steps as in (80)–(86), we can also show that

\[
nR_2 = H(W_2) \\
\leq I(W_2; Y_{s_1-s_2-r} \rightarrow d_2, Z_{1-n}^n) + n\epsilon_2 \\
= I(W_2; Z_{1-n}^n) + I(W_2; Y_{s_1-s_2-r} \rightarrow d_2 | Z_{1-n}^n) + n\epsilon_2 \tag{96}
\]

\[
= \sum_{t=1}^n I(W_2; Y_{s_1-s_2-r} \rightarrow d_2(t) | Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \\
+ n\epsilon_2
\]

\[
= n\epsilon_2 + \sum_{t=1}^n \left( I(W_2; Y_r \rightarrow d_2(t) | Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \\
+ I(W_2; Y_{s_1-s_2-r} \rightarrow d_2(t) | Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \\
+ I(W_2; Y_{s_1-s_2-r} \rightarrow d_2(t) | Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \right) \tag{97}
\]

\[
\leq n\epsilon_2 + \sum_{t=1}^n I(W_2; Y_r \rightarrow d_2(t) | Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \\
+ n_t s_2 p_2(d_2) + 0 \tag{98}
\]

where (96), (97), and (98) follows from the chain rule and the independence between \( W_2 \) and \( Z_{1-n}^n \); and (99) follows from similar derivation as in (85). We then have

\[
(99) = n\epsilon_2 + n_t s_2 p_2(d_2) \\
+ \sum_{t=1}^n \left( I(Y_r \rightarrow d_2(t) | Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \\
- I(Y_r \rightarrow d_2(t) | W_2, Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1}, [Z_{1-n}^n]) \right) \tag{100}
\]

\[
\leq n\epsilon_2 + n_t s_2 p_2(d_2) \\
+ \sum_{t=1}^n \left( H(Y_r \rightarrow d_2(t) | [Z_{1-n}^n]) - H(Y_r \rightarrow d_2(t) | U(t), [Z_{1-n}^n]) \right) \tag{101}
\]

\[
= n\epsilon_2 + n_t s_2 p_2(d_2) + \sum_{t=1}^n I(U(t); Y_r \rightarrow d_2(t) | Z_{1-n}^n) \tag{102}
\]

where (100) and (102) follows from the definition of the mutual information; and (101) follows from the fact that \( \text{conditioning does not increase the entropy} \) and \( Y_{s_1-s_2-r} \rightarrow d_2(t)_{1-1} \) is a subset of \( U(t) \). Since the mutual information is always non-negative, we now have

\[
(R_2 - t_s p_s(d_2) - \epsilon_2)^+ \\
\leq \frac{1}{n} \sum_{q_t=1}^n I(U(t); Y_r \rightarrow d_2(t) | [\sigma, Z_{1-n}^n]) \\
= \sum_{q_t=1}^n \text{Prob}(q_t = q_t) \cdot I(U(q_t); Y_r \rightarrow d_2(q_t) | [Z_{1-n}^n], q_t = q_t) \tag{103}
\]

\[
= \sum_{q_t=1}^n \frac{1}{n} \cdot H(Y_r \rightarrow d_2(q_t) | [Z_{1-n}^n], q_t = q_t) \\
- \sum_{q_t=1}^n \frac{1}{n} \cdot H(Y_r \rightarrow d_2(q_t) | U(q_t), [Z_{1-n}^n], q_t = q_t) \tag{104}
\]

where (103) follows from the definition of the conditional mutual information and the fact that the distribution of \( U(q_t) \) and \( Y_r \rightarrow d_2(q_t) \) does not depend on the future channel realization \( Z_{1-n}^n \); and (104) follows from the definition of the mutual information. We now discuss the first summation in (104)

\[
\sum_{q_t=1}^n \frac{1}{n} \cdot H(Y_r \rightarrow d_2(q_t) | [Z_{1-n}^n], q_t = q_t) \\
= \sum_{q_t=1}^n \frac{1}{n} \cdot H(Y_r \rightarrow d_2(q_t) | Z_{1-n}^n, q_t = q_t, \Theta_r, \Theta_Z) \tag{105}
\]

\[
= \sum_{q_t=1}^n \frac{1}{n} \cdot \sum_{\forall[z_{1-n}^n]} p_{[z_{1-n}^n]} | \Theta_r, \Theta_Z [z_{1-n}^n, \theta_r, \theta_Z] \\
\cdot H(Y_r \rightarrow d_2(q_t) | [z_{1-n}^n], \Theta_r = \theta_r, \Theta_Z = \theta_Z) \tag{106}
\]

\[
= \sum_{q_t=1}^n \frac{1}{n} \cdot \sum_{\forall[z_{1-n}^n]} p_{[z_{1-n}^n]} | \Theta_r, \Theta_Z [z_{1-n}^n, 1, 1] \\
\cdot H(X_r(q_t) | [z_{1-n}^n], \Theta_r = 1, \Theta_Z = 1) \tag{107}
\]

where (105) follows from the fact that \( \Theta_r \)'s are functions of \( Q_t \) and \( Z_{1-n}^n \); (106) follows from the definition of the conditional entropy; and (107) follows from the fact that \( Y_r \rightarrow d_2(q_t) \) is not erasure only if \( \sigma(q_t) = r \) and \( Z_r \rightarrow d_2(q_t) \) equals to one. Furthermore \( Y_r \rightarrow d_2(q_t) = X_r(q_t) \) under such a condition.

We can further simplify (107) by the following steps. We first note that conditioning on \( [z_{1-n}^n] \) is a subset of \( U(t) \). Since the mutual information is always non-negative, we now have

\[
H(X_r(q_t) | [z_{1-n}^n], \Theta_r = 1, \Theta_Z = 1) \\
= H(X_r(q_t) | [z_{1-n}^n], \Theta_r = 1, \Theta_Z = 1) \tag{108}
\]
Also the joint probability can be rewritten as
\[
\sum_{\forall z} p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}, \Theta_{Z_2}}(\left[z\right]^q_{t+1}, 1, 1) = \sum_{\forall z} p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}}(\left[z\right]^q_{t+1}, 1) + \sum_{\forall z} p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}}(z, 1|\left[z\right]^q_{t+1}, 1) (109)
\]
\[
= \left( \sum_{\forall z} p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}}(\left[z\right]^q_{t+1}, 1) \right) . p_{\cdot}(d_2). (110)
\]
where (109) follows from the basic probability definition, and (110) follows from the assumption that the channel is memoryless.
(108) and (110) help us rewrite (107) as
\[
(107) = t_r \cdot p_{\cdot}(d_2), \sum_{q_t=1}^n \sum_{\forall z} p_{\left[Z\right]^q_{t+1}}(\left[z\right]^q_{t+1}, 1) , H(X_r(q_t)|\left[z\right]^q_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2})
\]
where \( p_{\left[Z\right]^q_{t+1}}(\left[z\right]^q_{t+1}, 1) \) and \( H(X_r(q_t)|\left[z\right]^q_{t+1}, 1) \) are the shorthand for \( p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}}(\left[z\right]^q_{t+1}, 1) \) and \( H(X_r(q_t)|\left[Z\right]^q_{t+1}) \), respectively.

Similarly, for the second summation in (104),
\[
\sum_{q_t=1}^n \frac{1}{n} . H(Y_{r \rightarrow d_2}(q_t)|U(q_t), [Z]^{q}_{t+1}, Q_t = q_t)
\]
\[
= \sum_{q_t=1}^n \frac{1}{n} . H(Y_{r \rightarrow d_2}(q_t)|U(q_t), [Z]^{q}_{t+1}, Q_t = q_t, \Theta_{\sigma}, \Theta_{Z_2}) (112)
\]
\[
= \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall u, \forall z} p_{\cdot}(u, [z]^{q}_{t+1}, \Theta_{\sigma}, \Theta_{Z_2})
\]
\[
\cdot H(Y_{r \rightarrow d_2}(q_t)|U(q_t) = u, [Z]^{q}_{t+1} = [z]^{q}_{t+1}, \Theta_{\sigma} = \Theta_{\sigma}, \Theta_{Z_2} = \Theta_{Z_2}) (113)
\]
\[
= \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall u, \forall z} p_{\cdot}(u, [z]^{q}_{t+1}, 1, 1)
\]
\[
\cdot H(X_r(q_t)|U(q_t) = u, [Z]^{q}_{t+1} = [z]^{q}_{t+1}, \Theta_{\sigma} = 1, \Theta_{Z_2} = 1) (114)
\]
where (112) follows from the fact that \( \Theta_{\sigma} \)s are functions of \( Q \) and \( [Z]^{q}_{t+1} \); (113) follows from the definition of the conditional entropy; and (114) follows from the fact that \( Y_{r \rightarrow d_2}(q_t) \) is not erased only if \( \sigma(q_t) = r \) and \( Z_{r \rightarrow d_2} \) equals to one. Furthermore \( Y_{r \rightarrow d_2}(q_t) = X_r(q_t) \) under such a condition.

We can further simplify (114) by the following steps. We first note that conditioning on \( U(q_t) = u, [Z]^{q}_{t+1} = [z]^{q}_{t+1}, \) and \( \Theta_{\sigma} = 1, \) the random variable \( X_r(q_t) \) is independent of \( Z(q_t) \) and \( \Theta_{Z_2} \). Notice that \( [Z]^{q-1}_{t+1} \) is a subset of \( U(q_t) \). Therefore, we have
\[
H(X_r(q_t)|U(q_t) = u, [Z]^{q}_{t+1} = [z]^{q}_{t+1}, \Theta_{\sigma} = 1, \Theta_{Z_2} = 1) = H(X_r(q_t)|U(q_t) = u, \Theta_{\sigma} = 1). (115)
\]

Also the joint probability can be rewritten as
\[
\sum_{\forall u, \forall z} p_{\cdot}(u, [z]^{q}_{t+1}, \Theta_{\sigma}, \Theta_{Z_2}) \sum_{q_t=1}^n \sum_{\forall z} p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}}(\left[z\right]^q_{t+1}, 1, 1) (u, [z]^{q}_{t+1}, 1, 1) = \sum_{\forall u, \forall z} p_{\cdot}(u, 1) . H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2})
\]
\[
= \sum_{\forall u, \forall z} p_{\cdot}(u, 1) . H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2})
\]
\[
= \sum_{\forall u, \forall z} p_{\left[Z\right]^q_{t+1}, \Theta_{\sigma}}(\left[z\right]^q_{t+1}, 1) . p_{\cdot}(d_2). (116)
\]
where (116) follows from the basic probability definition, and (117) follows from the assumption that the channel is memoryless.
(115) and (117) help us rewrite (114) as
\[
(117) = t_r \cdot p_{\cdot}(d_2), \sum_{q_t=1}^n \sum_{\forall u} p_{\cdot}(u, 1) \cdot H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2})
\]
where \( p_{\cdot}(u, 1) \) and \( H(X_r(q_t)|[z]^{q}_{t+1}, 1) \) are the shorthand for \( p_{\cdot}(u, \Theta_{\sigma}), (u, 1) \) and \( H(X_r(q_t)|U(q_t) = u, \Theta_{\sigma} = 1) \), respectively.

Combining (111) and (118), we can rewrite (104) in the following form.
\[
(R_2 - t_s p_{s_{2}}(d_2) - \epsilon_2)^+ \leq t_r \cdot p_{\cdot}(d_2)
\]
\[
\left( \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall z} p_{\left[Z\right]^q_{t+1}}(\left[z\right]^q_{t+1}, 1) . H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2}) \right) - \frac{1}{t_r} \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall u} p_{\cdot}(u, 1) \cdot H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2})
\]
Summing up \( \frac{1}{p_{\cdot}(d_2)} \) and \( \frac{1}{p_{\cdot}(d_2)} \), we thus have
\[
(R_1 - t_s p_{s_{1}}(d_1, d_2) - \epsilon_1)^+ \leq t_r \cdot p_{\cdot}(d_2)
\]
\[
\left( \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall z} p_{\left[Z\right]^q_{t+1}}(\left[z\right]^q_{t+1}, 1) . H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2}) \right) - \frac{1}{t_r} \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall u} p_{\cdot}(u, 1) \cdot H(X_r(q_t)|[z]^{q}_{t+1}, 1) \leftarrow t_r \rightarrow \Theta_{\sigma}, \Theta_{Z_2})
\]
\[
\leq t_r, (120)
\]
where (121) is based on the following observations. We first note that by definition
\[
t_r = \sum_{q_t=1}^n \frac{1}{n} \cdot \text{Prob}(\sigma(q_t) = r)
\]
\[
= \sum_{q_t=1}^n \frac{1}{n} \sum_{\forall z} p_{\left[Z\right]^q_{t+1}}(\left[z\right]^q_{t+1}, 1).
\]
Therefore, the fraction term in (120) can be viewed as the normalization of the conditional entropy \( H(X_r(q_t)|[z]^{q-1}_{t+1}, 1) \). Since each conditional entropy is no larger than 1 (with the base of the logarithm being \( q \)), we thus have (121).
(121) holds for arbitrary $\epsilon > 0$. Letting $\epsilon \to 0^{13}$, we thus have the following final inequality.

$$\frac{(R_1 - t_s p_1(d_1, d_2))^+}{p_r(d_1, d_2)} + \frac{(R_2 - t_s p_2(d_2))^+}{p_r(d_2)} \leq t_r,$$

which gives us (13). (12) can be proven by symmetry. The proof of the outer bound is thus complete.

B. Detailed Achievability Analysis

In this appendix, we finish the discussion about the policy feasibility in Section V-A. The feasibility for Policy $\Gamma_{s_1, 0}$ and Policy $\Gamma_{s_1, 1}$ has been proven in Section V-A. In the following discussion about the space dimensions, we again rely on the first order, expectation-based analysis and assume the application of the law of large numbers implicitly.

**Policy $\Gamma_{s_1, 2}$:** Similar to the analysis for Policy $\Gamma_{s_1, 1}$, assuming $q \geq 2$, the condition that (28) being non-empty is equivalent to whether the following dimension-based inequality is satisfied.

$$\dim(S_2) - \dim((S_2 \cap (S_1 \oplus S_r)) = \dim(S_1 \oplus S_2 \oplus S_r) - \dim(S_1 \oplus S_r) > 0, \quad (122)$$

where (122) follows from Lemma 2.

Similar to the discussion in $\Gamma_{s_1, 0}$ and $\Gamma_{s_1, 1}$, we will quantify individual dimension at the end of $\Gamma_{s_1, 2}$, the policy of interest, and prove that even in the end of $\Gamma_{s_1, 2}$, the dimension difference in (122) is strictly larger than 0. Therefore, throughout the entire duration of $\Gamma_{s_1, 2}$, (122) is larger than 0 and $\Gamma_{s_1, 2}$ is always feasible.

We first focus on $\dim(S_1 \oplus S_2 \oplus S_r)$. Since $S_1 \oplus S_2 \oplus S_r$ is a subset of the exclusion set in $\Gamma_{s_1, 0}$, every time a $\Gamma_{s_1, 0}$ packet is received by one of $d_1$, $d_2$, and $r$, $\dim(S_1 \oplus S_2 \oplus S_r)$ will increase by one. On the other hand, notice that $S_1 \oplus S_2 \oplus S_r$ is a superset of the inclusion set in $\Gamma_{s_1, 1}$ and $\Gamma_{s_1, 2}$. Hence $\dim(S_1 \oplus S_2 \oplus S_r)$ remains the same throughout $\Gamma_{s_1, 1}$ and $\Gamma_{s_1, 2}$. As a result, in the end of policy $\Gamma_{s_1, 2}$, we have

$$E\{\dim(S_1 \oplus S_2 \oplus S_r)\} = n\omega_1^0 + n\omega_1^1 + n\omega_1^2 p_1(d_1, d_2, r). \quad (123)$$

We now focus on $\dim(S_1 \oplus S_r)$. Since $S_1 \oplus S_r$ is a subset of the exclusion sets of $\Gamma_{s_1, 0}$, $\Gamma_{s_1, 1}$ and $\Gamma_{s_1, 2}$, every time a packet of $\Gamma_{s_1, 0}$, $\Gamma_{s_1, 1}$, or $\Gamma_{s_1, 2}$ is received by one of $d_1$ and $r$, $\dim(S_1 \oplus S_r)$ will increase by one. As a result, in the end of policy $\Gamma_{s_1, 2}$, we have

$$E\{\dim(S_1 \oplus S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2 p_1(d_1, r). \quad (124)$$

Jointly, (123), (124), and (20) imply (122) in the end of $\Gamma_{s_1, 2}$.

**Policy $\Gamma_{s_1, 3}$:** Similar to the analysis of the previous policies, assuming $q \geq 2$, the condition that (29) being non-empty is equivalent to whether the following dimension-based inequality is satisfied.

$$\dim(S_r) - \dim((S_2 \cap S_r) \oplus S_r) = \dim((S_2 \cap S_r) \oplus S_1 \oplus S_r) - \dim((S_2 \cap S_r) \oplus S_1) \quad (125)$$

$$= \dim(S_1 \oplus S_r) - (\dim(S_1) + \dim(S_2 \cap S_r) - \dim(S_1 \oplus S_r)) \quad (126)$$

$$= \dim(S_1 \oplus S_r) - \dim(S_1) - (\dim(S_2) + \dim(S_r) - \dim(S_1 \oplus S_r) + \dim(S_1 \oplus S_2 \oplus S_r) + \dim(S_1 \oplus S_2 \oplus S_r) > 0, \quad (127)$$

where (125) follows from Lemma 2; (126) follows from simple set operations and from Lemma 2; and (127) follows from Lemma 2.

Similar to the previous discussion, we will quantify individual dimension at the end of $\Gamma_{s_1, 3}$, the policy of interest and prove that even in the end of $\Gamma_{s_1, 3}$, the dimension difference in (127) is strictly larger than 0. Therefore, throughout the entire duration of $\Gamma_{s_1, 3}$, (127) is larger than 0 and $\Gamma_{s_1, 3}$ is always feasible.

By similar analysis,\(^{14}\) in the end of $\Gamma_{s_1, 3}$ we have

$$E\{\dim(S_1)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2 + \omega_1^3) p_1(d_1), \quad (128)$$

$$E\{\dim(S_2)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^3) p_1(d_2), \quad (129)$$

$$E\{\dim(S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2) p_1(r), \quad (130)$$

$$E\{\dim(S_1 \oplus S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2) p_1(d_1, r), \quad (131)$$

$$E\{\dim(S_2 \oplus S_r)\} = n(\omega_1^0 + \omega_1^2) p_1(d_2, r). \quad (132)$$

What remains to be decided is the value of $\dim(S_1 \cap S_2 \cap S_r)$ at the end of Policy $\Gamma_{s_1, 3}$. To proceed, we introduce an auxiliary node $a$ in the following way. Whenever a vector $v$ sent by $s_1$ is received by both $d_1$ and $r$, we let the auxiliary node $a$ observe such $v$ as well. The knowledge space of $a$, denoted by $S_a$, is thus the linear span of all vectors received by both $d_1$ and $r$.

We first argue that $S_a = S_1 \cap S_r$ in the end of policy $\Gamma_{s_1, 2}$. Since $a$ only observes those vectors commonly available at both $d_1$ and $r$, the knowledge space of $S_a$ is a subset of $S_1 \cap S_r$. Knowing $S_a \subseteq S_1 \cap S_r$, we can quickly check that $S_a$ is a subset of the exclusion sets in Policies $\Gamma_{s_1, 0}$, $\Gamma_{s_1, 1}$, and $\Gamma_{s_1, 2}$. Therefore, every time node $a$ receives a packet during policies $\Gamma_{s_1, 0}$, $\Gamma_{s_1, 1}$, and $\Gamma_{s_1, 2}$, the dimension of $S_a$ will increase by one. Therefore, we have

$$E\{\dim(S_a)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2) p_1(d_1 r) \quad (133)$$

in the end of $\Gamma_{s_1, 2}$. On the other hand, by similar analysis as before, we have

$$E\{\dim(S_1)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2) p_1(d_1),$$

$$E\{\dim(S_2)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2) p_1(r),$$

$$E\{\dim(S_1 \oplus S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_1^2) p_1(d_1, r),$$

\(^{14}\)The derivation of (129) for the case of Policy $\Gamma_{s_1, 3}$ uses the following inequality as well.

$$\dim(S_1) \subseteq (S_r \cap (S_2 \cap S_r)) = (S_r \cap S_2).$$
in the end of policy $\Gamma_{s1,2}$. By Lemma 2, we thus have $\dim(S_a) = \dim(S_1 \cap S_r)$. As a result, we have proven $S_a = (S_1 \cap S_r)$ in the end of $\Gamma_{s1,2}$.

By the above analysis, we thus have $(S_1 \cap S_2 \cap S_r) = S_a \cup S_2$. By similarly dimension-based analysis, in the end of $\Gamma_{s1,2}$ we have

$$E\{\dim(S_2)\} = n(\omega_{s1}^0 + \omega_{s1}^1)p_1(d_2), \quad (134)$$
$$E\{\dim(S_2 \oplus S_0)\} = n(\omega_{s1}^0 + \omega_{s1}^1)p_1(d_2, d_1r), \quad (135)$$

where $p_1(d_2, d_1r)$ in (135) is the probability that at least one of node $d_2$ and node $a$ receives the packet and (135) follows from the observation that $S_2 \oplus S_0$ is a subset of the exclusion sets of $\Gamma_{s1,0}$, $\Gamma_{s1,1}$ and is a superset of the inclusion set of $\Gamma_{s1,2}$. By (133), (134), and (135), we have thus proven that

$$E\{\dim(S_1 \cap S_2 \cap S_r)\} = E\{\dim(S_a \cap S_2)\} = n(\omega_{s1}^0 + \omega_{s1}^1)p_1(d_1d_2r) + n\omega_{s1}^2p_1(d_1r) \quad (136)$$

in the end of $\Gamma_{s1,2}$.

In the following, we will quantify the increment of $\dim(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s1,3}$. To that end, we introduce two more auxiliary nodes $b$ and $c$. In the beginning of $\Gamma_{s1,3}$, we let node $b$ (resp. $c$) be aware of the knowledge space $S_1 \cap S_r$ (resp. $S_2 \cap S_r$). During $\Gamma_{s1,3}$, whenever a packet is received by $d_1$ (resp. $d_2$), we let the auxiliary node $b$ (resp. $c$) observe such a packet as well. From the construction, it is clear that the following equalities hold in the beginning of $\Gamma_{s1,3}$.

$$S_b = S_1 \cap S_r, \quad (137)$$
$$S_c = S_2 \cap S_r. \quad (138)$$

We will prove that (137) and (138) hold even in the end of $\Gamma_{s1,3}$ as well.

In the following, we will prove that (137) holds in the end of $\Gamma_{s1,3}$. We first note that by our construction, we always have $S_1 \supset S_b \supset (S_1 \cap S_r)$. Knowing that $S_b$ is always a subset of $S_1$, and $S_1$ is a subset of the exclusion sets in $\Gamma_{s1,3}$, we can see that every time $d_1$ receives a packet during policy $\Gamma_{s1,3}$, $\dim(S_b)$ will increase by one. Moreover, only when $d_1$ receives a packet during policy $\Gamma_{s1,3}$ will $\dim(S_b)$ increase. As a result, the increment of $\dim(S_b)$ during $\Gamma_{s1,3}$ equals the number of times $d_1$ receives a packet during $\Gamma_{s1,3}$. On the other hand, $\dim(S_1 \cap S_r) = \dim(S_1) + \dim(S_r) - \dim(S_1 \oplus S_r)$. Since both $S_1$ and $S_1 \oplus S_r$ are supersets of the inclusion set of $\Gamma_{s1,3}$, both $\dim(S_1)$ and $\dim(S_1 \oplus S_r)$ remain identical during $\Gamma_{s1,3}$. Therefore, the increment of $\dim(S_1 \cap S_r)$ is identical to the increment of $\dim(S_1)$ during $\Gamma_{s1,3}$. As a result, the increment of $\dim(S_1 \cap S_r)$ during $\Gamma_{s1,3}$ equals the number of times $d_1$ receives a packet during $\Gamma_{s1,3}$. We have thus proven $\dim(S_b) = \dim(S_1 \cap S_r)$ in the end of $\Gamma_{s1,3}$, which implies (137). (138) can be proven by symmetry.

To quantify the increment of $\dim(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s1,3}$, we notice that $\dim(S_1 \cap S_2 \cap S_r) = \dim(S_1 \cap S_2) = \dim(S_1) + \dim(S_2) - \dim(S_1 \oplus S_2)$. As a result, the increment of $\dim(S_1 \cap S_2 \cap S_r)$ during policy $\Gamma_{s1,3}$ is the summation of the increments of $\dim(S_1)$ and $\dim(S_2)$ minus the increment of $\dim(S_1 \oplus S_2)$ during $\Gamma_{s1,3}$. By our construction, the increments of $S_b$, $S_c$, and $S_b \oplus S_c$ during $\Gamma_{s1,3}$ are simply $n\omega_{s1}^3p_1(d_1)$, $n\omega_{s1}^3p_1(d_2)$, and $n\omega_{s1}^3p_1(d_1, d_2)$, respectively. As a result, the increment of $\dim(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s1,3}$ is simply $n\omega_{s1}^3p_1(d_1d_2)$.

Combining (136), we have thus proven that

$$E\{\dim(S_1 \cap S_2 \cap S_r)\} = n(\omega_{s1}^0 + \omega_{s1}^1)p_1(d_1d_2r) + n\omega_{s1}^2p_1(d_1r) + n\omega_{s1}^3p_1(d_1d_2) \quad (139)$$

in the end of Policy $\Gamma_{s1,3}$.

Jointly, (128) to (132), (139), and (21) imply (127) in the end of $\Gamma_{s1,3}$.

**Policy $\Gamma_{s1,4}$:** Similar to the analysis of the previous policies, the condition that (30) being non-empty is equivalent to whether the following dimension-based inequality is satisfied in the end of $\Gamma_{s1,4}$.

$$\dim(S_2 \cap S_r) - \dim(S_1 \cap S_2 \cap S_r) = (\dim(S_2) + \dim(S_r) - \dim(S_2 \oplus S_r)) - \dim(S_1 \cap S_2 \cap S_r) > 0. \quad (140)$$

Similar to the previous discussion, we will quantify individual dimension at the end of $\Gamma_{s1,4}$ and prove that (140) holds in the end of $\Gamma_{s1,4}$.

By similar analysis, we have

$$E\{\dim(S_2)\} = n(\omega_{s1}^0 + \omega_{s1}^1)p_1(d_2), \quad (141)$$
$$E\{\dim(S_1)\} = n(\omega_{s1}^0 + \omega_{s1}^1 + \omega_{s1}^2)p_1(r), \quad (142)$$
$$E\{\dim(S_2 \oplus S_r)\} = n\omega_{s1}^2p_1(d_2, r), \quad (143)$$

in the end of $\Gamma_{s1,4}$. What remains to be decided is the value of $\dim(S_1 \cap S_2 \cap S_r)$ at the end of Policy $\Gamma_{s1,4}$. In (139), we have already quantified $\dim(S_1 \cap S_2 \cap S_r)$ in the end of $\Gamma_{s1,3}$. In the following, we will quantify the increment of $\dim(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s1,4}$. By (30), we can see that every time $d_1$ receives a packet during $\Gamma_{s1,4}$, $\dim(S_1 \cap S_2 \cap S_r)$ will increase by one. As a result, the increment of $\dim(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s1,4}$ is $n\omega_{s1}^3p_1(d_1)$. Together with (139), we have proven that

$$E\{\dim(S_1 \cap S_2 \cap S_r)\} = n(\omega_{s1}^0 + \omega_{s1}^1)p_1(d_1d_2r) + n\omega_{s1}^2p_1(d_1r) + n\omega_{s1}^3p_1(d_1) \quad (144)$$

in the end of $\Gamma_{s1,4}$. Jointly, (141) to (144) and (22) imply that (140) holds in the end of $\Gamma_{s1,4}$.

The feasibility of policy $\Gamma_{s_k, k}$, $k = 0, 1, 2, 3, 4$, can be proven by symmetry.

**Policy $\Gamma_{r,1}$:** We first notice that the inclusion space and exclusion space of Policy $\Gamma_{r,1}$ are the same as of Policy $\Gamma_{s3,3}$. Hence to prove the feasibility of Policy $\Gamma_{r,1}$, we need to prove that (127) holds in the end of $\Gamma_{r,1}$. By similar analysis, we
have
\[ E\{\dim(S_1)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(d_1) + n\omega_{r,N}p_r(d_1), \quad (145) \]
\[ E\{\dim(S_2)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(d_2) + n\omega_{r,N}p_r(d_2), \quad (146) \]
\[ E\{\dim(S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(r), \quad (147) \]
\[ E\{\dim(S_1 \oplus S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(d_1, r), \quad (148) \]
\[ E\{\dim(S_2 \oplus S_r)\} = n\omega_1^0p_1(d_2, r), \quad (149) \]
in the end of the first Policy \(\Gamma_{r,1}\).

What remains to be decided is the value of \(\dim(S_1 \cap S_2 \cap S_r)\) at the end of Policy \(\Gamma_{r,1}\). In (144) we have computed the value of \(\dim(S_1 \cap S_2 \cap S_r)\) in the end of \(\Gamma_{r,1}\). As a result, we only need to quantify the increment of \(\dim(S_1 \cap S_2 \cap S_r)\) during \(\Gamma_{r,1}\). By the same analysis as used when we quantify the increment of \(\dim(S_1 \cap S_2 \cap S_r)\) during \(\Gamma_{r,3}\), the increment of \(\dim(S_1 \cap S_2 \cap S_r)\) during \(\Gamma_{r,3}\) is \(n\omega_{r,N}p_r(d_1d_2)\). By (144), we have shown that
\[ E\{\dim(S_1 \cap S_2 \cap S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(d_2d_2) + n\omega_1^2p_1(d_1r) + n\omega_1^3p_1(d_2r) + n\omega_{r,N}p_r(d_1d_2) \quad (150) \]
in the end of the first Policy \(\Gamma_{r,1}\). Jointly, (145) to (150) and (23) imply that (127) holds in the end of the first Policy \(\Gamma_{r,1}\).

The discussion of Policy \(\Gamma_{r,2}\) follows symmetrically.

**Policy \(\Gamma_{r,3}\) for v(1):** We will prove that for the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\), we can always choose \(v(1)\) according to (38). To that end, we first notice that the inclusion space and exclusion space in (38) are the same as those of Policy \(\Gamma_{s,4}\). Hence to prove that (38) remains non-empty during the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\), we need to prove that (140) holds in the end of the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\). By the same analysis as used in the previous policies, we have
\[ E\{\dim(S_2)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(d_2) + n\omega_{r,N}p_r(d_2), \quad (151) \]
\[ E\{\dim(S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(r), \quad (152) \]
\[ E\{\dim(S_2 \oplus S_r)\} = n\omega_1^0p_1(d_2, r), \quad (153) \]
in the end of the first Policy \(\Gamma_{r,3}\). What remains to be decided is the value of \(\dim(S_1 \cap S_2 \cap S_r)\) at the end of the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\). In (150) we have computed the value of \(\dim(S_1 \cap S_2 \cap S_r)\) in the end of \(\Gamma_{r,1}\). As a result, we only need to quantify the increment of \(\dim(S_1 \cap S_2 \cap S_r)\) during \(\Gamma_{r,3}\). By the same analysis as used when we quantify the increment of \(\dim(S_1 \cap S_2 \cap S_r)\) during \(\Gamma_{s,4}\), the increment of \(\dim(S_1 \cap S_2 \cap S_r)\) during the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\) is \(n\omega_{r,N}p_r(d_1)\). By (150), we have shown that
\[ E\{\dim(S_1 \cap S_2 \cap S_r)\} = n(\omega_1^0 + \omega_1^1 + \omega_2^1 + \omega_3^1 + \omega_4^1)p_1(d_1) + n\omega_{r,N}p_r(d_1) \quad (154) \]
in the end of the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\). Jointly, (151) to (154) and (24) imply that (140) holds in the end of the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\).

The discussion of the first \(n\omega_{r}^1\) time slots of Policy \(\Gamma_{r,3}\) follows symmetrically.

The above analysis completes the achievability proof stated in Section V-A.

**C. The Proof of Lemma 3**

In this appendix, we will show that the assignment of \(\omega_0^0\) to \(\omega_r^i\) in (62) to (68) is non-negative and satisfies (14), (15), (54) to (61).

We first show that the \(\omega_0^0\) to \(\omega_r^{i}\) are non-negative. By the definitions, \(\omega_0^0, \omega_1^1, \omega_2^1, \omega_r^1, \omega_r^2\) are non-negative. Since \(\max\{p_r(r), p_1(d_j)\} \leq p_r(d_j), \omega_r^1\) is non-negative. By (7), we have \(R_i \leq t_s, p_r(r)\). Therefore, \(\omega_r^3\) is non-negative. There are two terms in the definition of \(\omega_r^3\) in (68). The numerator of the first term is no less than the numerator of the second term. The denominator of the first term is no larger than the denominator of the second term. As a result, \(\omega_r^i\) is non-negative.

By (62) to (65), we have
\[ \sum_{k=0}^{3} \omega_0^k = \frac{R_i}{p_r(d_j, r)} + \frac{R_i}{p_r(r)} - \frac{R_i}{p_r(d_j, r)} + \frac{R_i}{p_r(r)} - \frac{R_i}{p_r(r)} \quad (155) \]
\[ = \frac{R_i}{p_r(r)} + \frac{R_i}{p_r(r)} - \frac{R_i}{p_r(r)} - \frac{R_i}{p_r(r)} \quad (156) \]
\[ \leq t_s, \quad (157) \]
where (156) is based on the fact that \(\min(x, y) + (x - y)^+ = x\) for arbitrary real values \(x\) and \(y\); and (157) follows from the fact that the minimum of two values is no more than any of two. (157) thus shows that our assignment satisfies (14).

To prove that (15) holds, we observe that our assignment leads to
\[ \omega_r^3 + \omega_r^2 + \omega_r^1 = \frac{R_i}{p_r(d_j, d_j)} + \frac{R_i}{p_r(d_j, d_j)} - \frac{R_i}{p_r(d_j, d_j)} - \frac{R_i}{p_r(d_j, d_j)} \leq t_r. \]
Hence the assignment satisfies (15). By noticing that
\[ \omega_{s_i}^0 p_i(d_j, r) = \frac{R_i}{p_i(d_j, r)} p_i(d_j, r) = R_i, \]
we have shown that the assignment satisfies (54). We now consider (55) and notice that

LHS of (55) = \[ R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(r) \]
\[ = R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{p_i(r)}{p_i(d_j, r)} \right) \]
\[ \leq R_i - R_i \frac{p_i(r)}{p_i(d_j, r)}, \quad (158) \]

RHS of (55) = \[ R_i \frac{\overline{p_i(d_j r)}}{p_i(d_j, r)} = R_i - R_i \frac{p_i(r)}{p_i(d_j, r)}, \quad (159) \]

where (158) follows from the definition of the minimum, and (159) follows from \( p_i(d_j, r) - p_i(r) = \overline{p_i(d_j r)} \). Since with the assignment, the left hand side (LHS) of (55) is no larger than the right hand side (RHS) of (55), the assignment satisfies (55). We now consider (56) and notice that

LHS of (56) = \[ R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \]
\[ = R_i \left( \min \left\{ \frac{p_i(d_j)}{p_i(r)}, 1 \right\} - \frac{p_i(d_j)}{p_i(d_j, r)} \right) \]
\[ \leq R_i - R_i \frac{p_i(d_j)}{p_i(d_j, r)}, \quad (160) \]

RHS of (56) = \[ R_i \frac{\overline{p_i(d_j r)}}{p_i(d_j, r)} = R_i - R_i \frac{p_i(d_j)}{p_i(d_j, r)}, \quad (161) \]

where (160) follows from the definition of the minimum, and (161) follows from \( p_i(d_j, r) - p_i(d_j) = \overline{p_i(d_j r)} \). Since with the assignment, the LHS of (56) is no larger than the RHS of (56), the assignment satisfies (56). We now consider (57) and notice that

LHS of (57) = \[ R_i \left( \frac{1}{p_i(r)} - \frac{1}{p_i(d_j)} \right) + p_i(r) \]
\[ = R_i \left( 1 - \frac{p_i(r)}{p_i(d_j)} \right)^+, \]

RHS of (57) = \[ R_i \frac{\overline{p_i(d_j r)}}{p_i(d_j, r)} \]
\[ - R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \]
\[ = R_i \left( 1 - \frac{p_i(r)}{p_i(d_j)} \right)^+, \]
\[ - R_i \left( \min \left\{ \frac{p_i(d_j)}{p_i(r)}, 1 \right\} - \frac{p_i(d_j)}{p_i(d_j, r)} \right) \]
\[ \geq (R_i - t_s, p_i(d_j))^+, \quad (168) \]

where (162) follows from \( p_i(d_j, r) - p_i(r) = \overline{p_i(d_j r)} \), and (163) follows from \( 1 - \min \left\{ \frac{p_i(r)}{p_i(d_j)}, 1 \right\} = \left( 1 - \frac{p_i(r)}{p_i(d_j)} \right)^+ \). Since with the assignment, the LHS of (57) is no larger than the RHS of (57), the assignment satisfies (57). We now consider (58) and notice that

LHS of (58) = \[ \min \left\{ R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, \right. \]
\[ \left. t_s, p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} , \]

RHS of (58) = \[ R_i \frac{\overline{p_i(r d_j)}}{p_i(d_j, r)} \]
\[ - R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \]
\[ = R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, \]
\[ - R_i \left( \min \left\{ p_i(d_j), 1 \right\} - \frac{p_i(d_j)}{p_i(r)} \right) \]
\[ = R_i - R_i \min \left\{ p_i(d_j) \frac{1}{p_i(r)} \right\}, \quad (164) \]

where (164) and (165) follow similar reasons as in (162) and (163). Hence the assignment satisfies (58). We now consider (59) and we notice that

LHS = \[ (R_i - t_s, p_i(d_j))^+, \]

RHS = \[ R_i \frac{\overline{p_i(r d_j)}}{p_i(d_j, r)} \]
\[ - R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \]
\[ - \min \left\{ R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, t_s, p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \]
\[ = R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, \]
\[ - R_i \left( \min \left\{ p_i(d_j), 1 \right\} - \frac{p_i(d_j)}{p_i(r)} \right) \]
\[ - \min \left\{ R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, t_s, p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \]
\[ = R_i - R_i \min \left\{ p_i(d_j) \frac{1}{p_i(r)} \right\}, \quad (166) \]

where (166) follows from \( p_i(d_j, r) - p_i(d_j) = \overline{p_i(r d_j)} \); and
(167) follows from the definition of $(\cdot)^{+}$. (168) follows from the following arguments.

\[
(167) = \max \left\{ 0, R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^{+} - t_s p_i(d_j) + R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
\geq \max \left\{ 0, R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^{+} - t_s p_i(d_j) + R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
= (168).
\]

Since with the assignment, the LHS of (59) is no greater than the RHS of (59), the assignment satisfies (59). We now consider (60) and notice that

\[
\text{LHS} = R_i - (R_i - t_s p_i(d_j))^{+} \frac{p_r(d_i)}{p_i(d_i, d_j)}, \\
\text{RHS} = R_i \left\{ \frac{p_i(d_j, r)}{p_i(d_j, r)} \right\}^{+} + (p_i(d_j) + p_i(r)) \cdot R_i \left( \min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) \\
+ R_i \left( 1 - \frac{p_i(r)}{p_i(d_j)} \right)^{+} + \min \left\{ R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^{+} + t_s p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
+ \frac{(R_i - t_s p_i(d_j))^{+}}{p_r(d_i, d_j)} (p_r(d_i, d_j) - p_r(d_i)) \\
= R_i \left\{ \frac{p_i(d_j, r) - p_i(d_j) - p_i(r)}{p_i(d_j, r)} \right\}^{+} + R_i \left( \min \left\{ 1 + \frac{p_i(d_j)}{p_i(r)}, 1 + \frac{p_i(r)}{p_i(d_j)} \right\} \right) \\
+ R_i \left( 1 - \frac{p_i(r)}{p_i(d_j)} \right)^{+} + \min \left\{ R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^{+} + t_s p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
+ \frac{(R_i - t_s p_i(d_j))^{+}}{p_r(d_i, d_j)} (p_r(d_i, d_j) - p_r(d_i)) \\
= R_i \left\{ \frac{p_i(d_j)}{p_i(d_j)} \cdot \frac{p_i(r)}{p_i(d_j)} \right\}^{+} + R_i \left( 1 - \frac{p_i(r)}{p_i(d_j)} \right)^{+} \\
+ \min \left\{ R_i \left( 1 - \frac{p_i(d_j)}{p_i(r)} \right)^{+} + t_s p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
+ \frac{(R_i - t_s p_i(d_j))^{+}}{p_r(d_i, d_j)} \left( 1 - \frac{p_r(d_i)}{p_r(d_i, d_j)} \right), \\
\text{Case 2: } p_i(d_j) < p_i(r).
\]

(169) follows from the equality \(\min\{x, y\} + (x - y)^{+} = x\) for any real valued \(x, y\). Plugging (171) and (170) into (169), we thus see that the LHS of (60) is no larger than the RHS of (60) with the assignment. Hence the assignment satisfies (60). We now consider (61) and notice that

\[
p_r(d_i)(\omega_{i,N}^x + \omega_{i,C}^y) = R_i
\]

which satisfies (61).

The above proof shows that the proposed assignment satisfies (14), (15), (54) to (61).

References


Wei-Cheng Kuo received the B.S. degree in electrical engineering from Nation Chiao Tung University, Hsinchu, Taiwan, in 2008. He is currently working toward the Ph.D. degree in electrical and computer engineering at Purdue University, Indiana, USA. His research interests are in Shannon capacity characterization of wireless network coding and scheduling algorithms for network coded traffic.

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

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