Capacity Region of Two Symmetric Nearby Erasure Channels With Channel State Feedback

Chih-Chun Wang; chihw@purdue.edu
School of Electrical and Computer Engineering, Purdue University, USA

***Abstract*—This work considers a commonly encountered wireless transmission scenario: Two nearby 1-hop flows $s_1 \to d_1$ and $s_2 \to d_2$ are within the transmission range of each other. The network nodes thus have to share the time resources, which limits the sum-rate performance. On the other hand, both s_i and d_i can (occasionally) overhear the transmission of the other pair (s_j, d_j) for all $i \neq j$, which opens up the opportunity of using *network coding* (NC) and ACK/NACK to improve the throughput. The key challenge, however, is that any dedicated communication between s_1 and s_2 also consumes the precious time resources. Hence NC coordination must be achieved through unreliable overhearing.**

In this work, the above scenario is modeled as four wireless nodes interconnected by *broadcast erasure channels* with channel state feedback. The corresponding capacity region (R_1, R_2) is fully characterized for the setting of symmetric, spatially independent erasure channels.

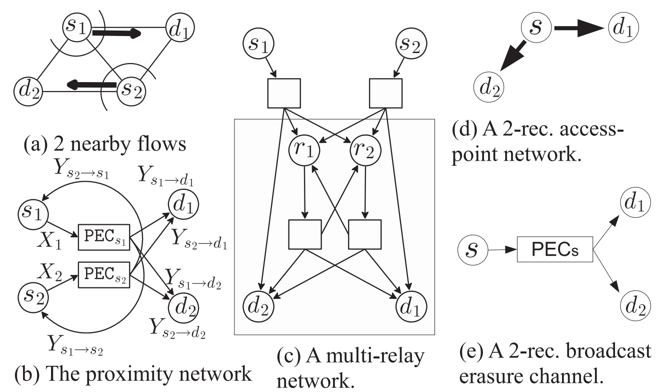

Fig. 1. (a) Two nearby 1-hop flows; (b) The corresponding wireless erasure network model; (c) The application to multi-relay networks. The rectangles represent the PECs; (d) A 2-receiver access-point network; (e) The corresponding 2-receiver broadcast erasure channel.

I. Introduction

This work considers a commonly encountered wireless transmission scenario: Two nearby 1-hop flows from s_1 to d_1 and from s_2 to d_2 are within the transmission range of each other, see Fig. 1(a). Assuming the node-exclusive interference model, commonly used in the network scheduling literature [7], the network nodes thus have to share the time resources, which limits the sum-rate performance. On the other hand, both s_i and d_i can (occasionally) overhear the transmission of the other pair (s_j, d_j), which opens up possibility of cooperative transmission. The above scenario can be modeled as four wireless nodes s_1, d_1, s_2, and d_2 interconnected by two broadcast packet erasure channels (PECs). That is, a packet X_1 sent by s_1 can be heard by a (random) subset of $\{s_2, d_1, d_2\}$ and a packet X_2 sent by s_2 can be heard by a (random) subset of $\{s_1, d_1, d_2\}$, see Fig. 1(b). We also assume causal *network-wide* channel state information (CSI) feedback. That is, after s_i sends a packet X_i, all four nodes $\{s_1, s_2, d_1, d_2\}$ know which of the three nodes $\{s_j, d_1, d_2\}$ has received X_i successfully. This can be achieved by s_j, d_1, and d_2 each broadcasting its ACK/NACK signals over the network via a very low-rate[1] control channel or via piggybacking the forward traffic. We term this setting (Fig. 1(b)) *the proximity network*. This work aims to characterize the corresponding capacity region under some symmetry conditions.

The study of the proximity network is critical to the understanding of general wireless networks. For example, the XOR-in-the-air protocol [5] shows that if two coexisting 2-hop flows $s_1 \to r \to d_1$ and $s_2 \to r \to d_2$ share a common relay r, the so called wireless X network, then relay r can perform network coding (NC) and enhances the throughput by 40–200% [5]. However, in practice a more likely scenario is that the 2-hop flows $s_1 \to r_1 \to d_1$ and $s_2 \to r_2 \to d_2$ have two distinct relays r_1 and r_2 that are in close proximity, see Fig. 1(c). A nature question is thus how much NC throughput gain one can still have when the two flows do not share a common relay r. As can be seen in the boxed area of Fig. 1(c), a critical component of this problem is the proximity network.

One key motivation of this work is to study the throughput benefit of CSI feedback for erasure networks [1]. Existing results on the erasure-network CSI-feedback capacity include the K-receiver broadcast channel capacity under various settings [2]–[4], [9], the 2-receiver XOR-in-the-air capacity with CSI feedback [6], and the linear NC capacity of the 2-receiver MIMO broadcast erasure channel [10].

The capacity of Fig. 1(b) is highly related to that of the 2-receiver broadcast PEC with CSI feedback [4] since if the logical sources s_1 and s_2 are situated at the same physical node s, then Fig. 1(b) collapses to Fig. 1(e). Compared to the results on broadcast erasure channels [4], there are two main challenges when characterizing the capacity of the proximity network. Firstly, s_1 and s_2 can only be coordinated through random overhearing. Secondly, what s_1 transmitted in the past can affect the current NC decision of s_2, which in turn influences s_1's NC decisions in the future. Questions like "how many time slots one should schedule s_1 and s_2, respectively, and in what order" thus need to be carefully addressed. The challenges of *coordination through overhearing* and *transmission node scheduling* do not exist when there is only one transmission node s as in the case of broadcast PECs [4].

The cooperation between two nearby flows has been actively investigated under the subject of interference channels with

[1]The feedback rate is 3 bits per each forward packet transmission.

source cooperation, see [8], [11] and the references therein. In contrast with the interference channel results, several distinct features of this work include: (i) Interference is fully avoided through time-sharing. We can thus separate the benefits of CSI feedback from that of the interference mitigation techniques; (ii) In addition to overhearing the (corrupted) packets over the air, we assume network-wide CSI, which is a form of dedicated feedback that can be easily implemented in practice; and (iii) We characterize the full capacity region (R_1, R_2), which is stronger than the sum-rate capacity characterization.

II. Problem Formulation And The Main Result

Assume slotted transmission. Within a total time budget of n time slots, source s_i would like to send $n \cdot R_i$ packets $\mathbf{W}_i \triangleq (W_{i,1}, \cdots, W_{i,nR_i})$ to destination d_i for $i \in \{1,2\}$. Each packet $W_{i,l}$, $l = 1, \cdots, nR_i$, is chosen independently and uniformly randomly from a finite field $\mathsf{GF}(q)$. For any time $t \in [n] \triangleq \{1, \cdots, n\}$, consider a random 6-dimensional *channel state vector*:

$$\begin{aligned}\mathbf{Z}(t) = &(Z_{s_1\to s_2}(t), Z_{s_1\to d_1}(t), Z_{s_1\to d_2}(t),\\ &Z_{s_2\to s_1}(t), Z_{s_2\to d_1}(t), Z_{s_2\to d_2}(t)) \in \{*, 1\}^6\end{aligned}$$

where "$*$" and "1" represent erasure and successful reception, respectively. That is, for any $i = 1, 2$, when s_i transmits a packet $X_i(t) \in \mathsf{GF}(q)$ in time t, node v (v can be any of $\{s_j, d_1, d_2\}$ where $j \neq i$) receives $Y_{s_i\to v}(t) = X_i(t)$ if $Z_{s_i\to v}(t) = 1$ and receives $Y_{s_i\to v}(t) = *$ if $Z_{s_i\to v}(t) = *$. For simplicity, we use $Y_{s_i\to v}(t) = X_i(t) \circ Z_{s_i\to v}(t)$ as shorthand. We also assume $\mathbf{Z}(t)$ is independent of $\mathbf{W}_1$ and $\mathbf{W}_2$.

To model interference, we assume that only one node can be scheduled in each time slot. We use $\sigma(t) \in \{s_1, s_2\}$ to denote the *scheduling decision* at time t. For convenience, when s_i is not scheduled at time t, we simply set $Y_{s_i\to v}(t) = *$, i.e.,

$$Y_{s_i\to v}(t) = X_i(t) \circ Z_{s_i\to v}(t) \circ 1_{\{\sigma(t)=s_i\}}.$$

We assume the two PECs in Fig. 1(b) are memoryless, stationary, symmetric, and spatially independent. That is, for all $i \in \{1,2\}$ the success probability $\mathsf{Prob}(Z_{s_i\to d_k}(t)) = p$ for all $k \in \{1,2\}$ and $\mathsf{Prob}(Z_{s_i\to s_j}(t)) = q$ for all $j \in \{1,2\}\backslash i$; Each coordinate of $\mathbf{Z}(t)$ is independently distributed; and $\mathbf{Z}(t)$ is independently and identically distributed over the time axis t. We use brackets $[\cdot]_1^t$ to denote the collection from time 1 to t. For example, $[\mathbf{Z}, Y_{s_1\to d_2}]_1^t \triangleq \{\mathbf{Z}(\tau), Y_{s_1\to d_2}(\tau) : \forall \tau \in [1,t]\}$.

Given the traffic load (R_1, R_2), a network code is defined by n scheduling decision functions

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

$2n$ encoding functions at s_1 and s_2, respectively: For all $t \in [n]$, $i \in \{1,2\}$, and $j \in \{1,2\}\backslash i$,

$$X_i(t) = f_{s_i,t}(\sigma(t), \mathbf{W}_i, [Y_{s_j\to s_i}, \mathbf{Z}]_1^{t-1}), \tag{2}$$

and 2 decoding functions at d_1 and d_2, respectively:

$$\hat{\mathbf{W}}_i = f_{d_i}([\sigma, Y_{s_1\to d_i}, Y_{s_2\to d_i}, \mathbf{Z}]_1^n), \quad \forall i \in \{1,2\}. \tag{3}$$

Namely, the scheduling decision at time t is based on the network-wide CSI in time 1 to $(t-1)$. Encoding at s_i depends on the scheduling decision, the information messages, what s_i overhears from s_j, and the past CSI. In the end, d_i decodes $\mathbf{W}_i$ based on the past scheduling decisions, what d_i has received, and the past network-wide CSI. We can then define the capacity of the proximity network, Fig. 1(b), as follows.

Definition 1: Fix the p and q values. A rate vector (R_1, R_2) is achievable if for any $\epsilon > 0$, there exists a network code with sufficiently large n and $\mathsf{GF}(q)$ such that $\max_{\forall i\in\{1,2\}} \mathsf{Prob}(\mathbf{W}_i \neq \hat{\mathbf{W}}_i) < \epsilon$. The capacity region is the closure of all achievable (R_1, R_2).

The main result of this work is described as follows.

Proposition 1: A rate vector (R_1, R_2) is in the capacity region if and only if the following inequalities are satisfied.

$$\frac{R_1}{p} + \frac{R_2}{1-(1-p)^2} \leq 1 \tag{4}$$

$$\frac{R_1}{1-(1-p)^2} + \frac{R_2}{p} \leq 1 \tag{5}$$

$$R_1 + R_2 \leq 1 - (1-p)(1-q). \tag{6}$$

A detailed sketch of the proof is provided in the next section.

Remark: One can quickly see that the larger the q value is, the easier source s_i can overhear packets from the other source s_j, the stronger coordination s_1 and s_2 can achieve through random overhearing, and the closer the proximity network in Fig. 1(b) is to the 2-receiver broadcast PEC in Fig. 1(e). One implication of Proposition 1 is that even with relatively weak overhearing between s_1 and s_2, as long as $q \geq \frac{p}{3-p}$ (i.e., roughly 1/3 to 1/2 of the p value), nodes s_1 and s_2 can be fully coordinated and the capacity of the proximity network is identical to that of the 2-receiver broadcast PEC. Also see the discussion in Section III-A.

III. Proof Of The Main Result

A. The outer bound proof

One challenge of the problem is that s_1 and s_2 have to coordinate through unreliable overhearing. A straightforward outer bound is thus to connect s_1 and s_2 through an auxiliary information pipe with infinite capacity, which effectively merges s_1 and s_2 into a single source s and eliminates the coordination problem. The original setting is thus converted into a symmetric 2-receiver broadcast PEC with each sub-channel $s \to d_i$ having success probability p. The capacity of the latter is described in (4) and (5), see [4], which serves as an outer bound of the original problem.

The third inequality (6) is a cut-set bound and can be derived as follows. For any feasible network code, we notice that conditioning on $[\mathbf{Z}]_1^n$, we have $\mathbf{W}_i \to [Y_{s_i\to d_i}, Y_{s_i\to s_j}]_1^n \to \hat{\mathbf{W}}_i$, $j \neq i$, being a Markov chain. As a result,

$$\begin{aligned}I(\mathbf{W}_i; \hat{\mathbf{W}}_i) &\leq H\left([Y_{s_i\to d_i}, Y_{s_i\to s_j}]_1^n \,\middle|\, [\mathbf{Z}]_1^n\right)\\ &\leq (1-(1-p)(1-q)) \cdot t_i\end{aligned}$$

where $I(\cdot;\cdot)$ is the mutual information and t_i is the expected number of s_i's transmissions, i.e., $t_i \triangleq \mathsf{E}\left(\sum_{\tau=1}^n 1_{\{\sigma(\tau)=s_i\}}\right)$ with $1_{\{\cdot\}}$ being the indicator function. As a result, we have

$$\frac{I(\mathbf{W}_1; \hat{\mathbf{W}}_1) + I(\mathbf{W}_2; \hat{\mathbf{W}}_2)}{1-(1-p)(1-q)} \leq t_1 + t_2 = n. \tag{7}$$

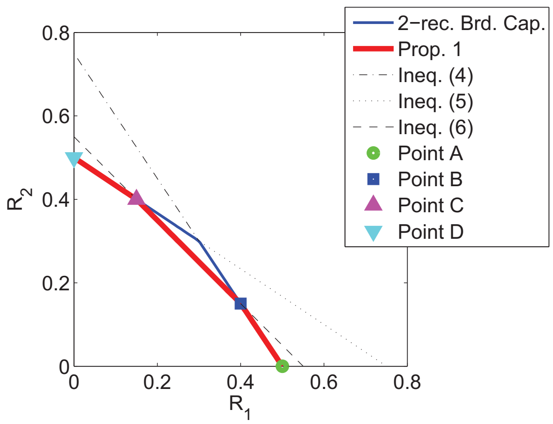

Fig. 2. The capacity region for the case of $p = 0.5$ and $q = 0.1$. The four corner points of the capacity region are denoted by A to D, which will be referred to in the achievability proof.

By Fano's inequality we can choose a feasible network code with $I(\mathbf{W}_i; \hat{\mathbf{W}}_i) \geq n(R_i - \epsilon)$ for any ϵ. (7) thus implies (6).

Fig. 2 illustrates the capacity region for the case of $p = 0.5$ and $q = 0.1$. Note that the larger the q value is, the less restrictive is the constraint in (6). When $q > \frac{p}{3-p}$ the line $R_1 + R_2 = 1 - (1-p)(1-q)$ no longer intersects with the capacity region of the 2-receiver broadcast erasure channel. Hence s_1 and s_2 are fully coordinated as long as $q \geq \frac{p}{3-p}$.

B. Proof of the inner bound

Due to space limit, we focus on the more interesting case of $q < \frac{p}{3-p}$, for which the capacity region described in (4) to (6) is a pentagon with 4 corner points A to D, see Fig. 2. The coordinates of Point A are $(R_1, R_2) = (p, 0)$, which can be achieved by letting only s_1 transmit. The coordinates of Point B are $(R_1, R_2) = (p - q, q(2 - p))$. For the following, we will prove that Point B can be achieved asymptotically. By symmetry, Points C and D can also be approached. We can then achieve the entire capacity region through time sharing.

The following scheme approaches Point B, i.e., $(R_1, R_2) = (p - q, q(2 - p))$. The scheme consists of 4 major phases. The first two phases contains 2 sub-phases, one for the transmission of s_1 and one for s_2. The network code executes the following (sub-)phases in sequence.

In Phase 1.s_1, source s_1 keeps transmitting an uncoded packet $W_{1,l}$ until it is received by at least one of $\{s_2, d_1, d_2\}$. s_1 then moves on to the next packet $W_{1,l+1}$. Phase 1.s_1 stops when all $\mathbf{W}_1$ packets have been heard by at least one of $\{s_2, d_1, d_2\}$. Phase 1.s_2 is symmetric to Phase 1.s_1.

After Phase 1.s_1, we can partition all $\mathbf{W}_1$ packets into 7 groups A_1 to A_7 based on their *reception status*. For example, the reception status of $W_{1,l}$ being $s_2\overline{d_1}d_2$ means that $W_{1,l}$ is received by s_2 and d_2 but not by d_1. The reception status of each group, A_1 to A_7, is summarized as follows.

$$A_1\colon s_2d_1d_2,\ \ A_2\colon \overline{s_2}d_1d_2,\ \ A_3\colon s_2\overline{d_1}d_2,\ \ A_4\colon \overline{s_2d_1}d_2,$$
$$A_5\colon s_2d_1\overline{d_2},\ \ A_6\colon \overline{s_2}d_1\overline{d_2},\ \text{ and }\ A_7\colon s_2\overline{d_1d_2}. \tag{8}$$

Symmetrically, after Phase 1.s_2 we can partition all $\mathbf{W}_2$ packets into 7 groups: B_1 to B_7, for which the reception status are

$$B_1\colon s_1d_1d_2,\ \ B_2\colon \overline{s_1}d_1d_2,\ \ B_3\colon s_1d_1\overline{d_2},\ \ B_4\colon \overline{s_1}d_1\overline{d_2},$$
$$B_5\colon s_1\overline{d_1}d_2,\ \ B_6\colon \overline{s_1d_1}d_2,\ \text{ and }\ B_7\colon s_1\overline{d_1d_2}. \tag{9}$$

The purpose of Phases 1.s_1 and 1.s_2 is to populate the network with the information packets $\mathbf{W}_1$ and $\mathbf{W}_2$.

After Phase 1.s_2, we move on to Phases 2.s_1 and 2.s_2:

§ PHASE 2.s_1

1: Set $l_4 \leftarrow 1$ and $l_7 \leftarrow 1$.
2: **while** $l_7 \leq |B_7|$, the number of $\mathbf{W}_2$ packets in B_7 **do**
3: Source s_1 transmits a linear sum of two packets: the l_7-th packet in B_7 and the l_4-th packet in A_4.
4: **if** at least one of $\{d_1, d_2\}$ receives the packet **then**
5: Set $l_7 \leftarrow l_7 + 1$.
6: **end if**
7: **if** at least one of $\{s_2, d_1\}$ receives the packet **then**
8: Set $l_4 \leftarrow l_4 + 1$.
9: **end if**
10: **end while**

Phase 2.s_2 is a symmetric version of Phase 2.s_1. That is, source s_2 transmits a linear sum of a B_4 and an A_7 packets.

The purpose of Phase 2.s_1 is two-fold. Firstly, B_7 contains the $\mathbf{W}_2$ packets that have been heard by neither d_1 nor d_2, see (9). Therefore, to facilitate NC in the subsequent phases, s_1 would like to transmit the B_7 packets. Secondly, A_4 contains the $\mathbf{W}_1$ packets that have been heard by d_2 but not by d_1, see (8). Such a mismatched reception is a good candidate for NC. However, A_4 is not heard by s_2, which means that if we would like to perform NC that involves the A_4 packets, only s_1 is up to such a task. To broaden the possibility so that s_2 can also perform NC that involves the A_4 packets, s_1 would also like to transmit the A_4 packets. Instead of sending the A_4 and B_7 packets separately, s_1 transmits a linear sum instead. The purpose of Phase 2.s_2 is symmetric to that of Phase 2.s_1.

After Phase 2.s_1, each of the B_7 packets, say the l_7-th packet of B_7, can be classified as one of the 3 groups: $B_{7.1}$ to $B_{7.3}$, depending on the reception status when sending the linear sum of the l_7-th packet of B_7 and an A_4 packets. The reception status of $B_{7.1}$ to $B_{7.3}$ is summarized as follows.

$$B_{7.1}\colon \overline{d_1}d_2,\ \ B_{7.2}\colon d_1\overline{d_2},\ \ B_{7.3}\colon d_1d_2. \tag{10}$$

We can partition the A_4 packets similarly. However, since the termination condition of Phase 2.s_1 is based on using up the B_7 packets (Line 2 of Phase 2.s_1), there may be some A_4 packets that have not been received by any of $\{s_2, d_1\}$ yet (those with indices beyond the last l_4 value when Phase 2.s_1 ends). As a result, A_4 can be partitioned into 4 groups $A_{4.1}$ to $A_{4.4}$ with reception status being

$$A_{4.1}\colon s_2\overline{d_1},\ \ A_{4.2}\colon \overline{s_2}d_1,\ \ A_{4.3}\colon s_2d_1,\ \ A_{4.4}\colon \overline{s_2d_1}. \tag{11}$$

Symmetrically after Phase 2.s_2, we can partition the A_7 packets into 3 groups $A_{7.1}$ to $A_{7.3}$ and the B_4 packets into 4 groups $B_{4.1}$ to $B_{4.4}$ with the reception status being

$$A_{7.1}\colon d_1\overline{d_2},\ \ A_{7.2}\colon \overline{d_1}d_2,\ \ A_{7.3}\colon d_1d_2 \tag{12}$$
$$B_{4.1}\colon s_1\overline{d_2},\ \ B_{4.2}\colon \overline{s_1}d_2,\ \ B_{4.3}\colon s_1d_2,\ \ B_{4.4}\colon \overline{s_1d_2}. \tag{13}$$

For throughput enhancement, we would like to mix some $\mathbf{W}_1$ packets that have been heard by d_2 but not d_1 with some $\mathbf{W}_2$ packets that have been heard by d_1 but not d_2. However, *since s_1 and s_2 are not located in the same physical node, those NC-eligible $\mathbf{W}_1$ and $\mathbf{W}_2$ packets are spread between s_1 and s_2*. We first focus on s_1. Recall that s_1 has all the $\mathbf{W}_1$ packets to begin with. Among them, the ones that are heard by d_2 but not by d_1 are in A_3 and A_4 (equivalently $A_{4.1}$ to $A_{4.4}$), see (8).[2] Note that s_1 can decode $B_{4.1}$ and $B_{4.3}$ during Phase 2.s_2 by subtracting the A_7 packets it received in Phase 1.s_1. Therefore, among all the $\mathbf{W}_2$ packets that s_1 has, the ones that are heard by d_1 but not by d_2 are B_3, $B_{4.1}$ and $B_{4.3}$. As a result, if s_1 sends a linear sum of one packet from the first group $\{A_3, A_{4.1}, \cdots, A_{4.4}\}$ and one packet from the second group $\{B_3, B_{4.1}, B_{4.3}\}$, then such a transmission can serve both destinations simultaneously. However, as will be elaborated shortly after, *there are other combinations s_1 can send that also serve both destinations simultaneously.*

We first argue that any packet in $A_{7.2}$ or $A_{7.3}$ can also be viewed as a $\mathbf{W}_2$ packet that is heard by d_1 but not d_2. We first notice that all the $B_{4.2}$ and $B_{4.3}$ packets are mixed one-to-one with the $A_{7.2}$ and $A_{7.3}$ packets as they correspond to the time instants for which a Phase 2.s_2 packet is received by d_2, see (13) and (12). Hence, whenever d_2 receives an $A_{7.2}$ (resp. $A_{7.3}$) packet, d_2 can subtract such a packet from one of the linear sums it received during Phase 2.s_2 and decode one of $B_{4.2}$ and $B_{4.3}$ packets. As a result, $A_{7.2}$ and $A_{7.3}$ can be viewed as $\mathbf{W}_2$ packets that have not been heard by d_2. Since $A_{7.2}$ and $A_{7.3}$ are pure $\mathbf{W}_1$ packets, *they will not interfere with any $s_1 \to d_1$ communications,*[3] and can thus be viewed as "transparent" or equivalently as "*heard by d_1*".

By the above arguments, we can enlarge the second group from $\{B_3, B_{4.1}, B_{4.3}\}$ to $\{B_3, B_{4.1}, A_{7.2}, A_{7.3}\}$. Note that upon receiving all $A_{7.2}$ and $A_{7.3}$ packets d_2 can decode all $B_{4.2}$ and $B_{4.3}$ packets. We thus remove $B_{4.3}$ from the second group since $B_{4.3}$ is now superseded by $A_{7.2}$ and $A_{7.3}$.

Let $B_{7.2}^{\text{sum}}$ denote the linear sums in which the $B_{7.2}$ packets participated during Phase 2.s_1. The difference between $B_{7.2}$ and $B_{7.2}^{\text{sum}}$ is that packets in $B_{7.2}$ are pure uncoded $\mathbf{W}_2$ packets while each packet in $B_{7.2}^{\text{sum}}$ is a linear sum of one $B_{7.2}$ packet and one A_4 packets that was sent during Phase 2.s_1 and received by d_1 but not by d_2, see Line 3 of Phase 2.s_1 and (10). We note that upon receiving a linear sum in $B_{7.2}^{\text{sum}}$, d_2 can decode the corresponding $B_{7.2}$ packet by subtracting the A_4 packet it received during Phase 1.s_1. Also, each linear sum in $B_{7.2}^{\text{sum}}$ is heard by d_1 during Phase 2.s_1, see (10). As a result, *each linear sum in $B_{7.2}^{sum}$ can also be viewed as a $\mathbf{W}_2$ packet heard by d_1 but not by d_2*. The second group $\{B_3, B_{4.1}, A_{7.2}, A_{7.3}\}$ can thus be further enlarged to $\{B_3, B_{4.1}, A_{7.2}, A_{7.3}, B_{7.2}^{\text{sum}}\}$.

[2]The reason that we still count $A_{4.2}$ and $A_{4.3}$ even though their reception status are $\overline{s_2}d_1$ and s_2d_1, see (11), is as follows. Those A_4 packets were mixed with B_7 packets during Phase 2.s_1. Therefore even though d_1 has received a linear combination of $A_{4.2}$ (resp. $A_{4.3}$) and some B_7 packets, d_1 still cannot decode the original $A_{4.2}$ (resp. $A_{4.3}$) packets. Hence $A_{4.2}$ and $A_{4.3}$ still belong to the category: heard by d_2 *but not by* d_1. The reason why $A_{7.2}$ is not counted is similar due to the mixture with B_4.

[3]Detailed definition of the concept of *non-interfering* can be found in [9].

The first row of Table I summarizes the packets that are (i) available at s_1 and (ii) can potentially be used for NC. The second row of Table I summarizes the symmetric situation faced by s_2. One can also see that all A_1, A_2, A_5, and A_6 packets have arrived at d_1 during Phase 1.s_1, see (8); and all $A_{7.1}$ and $A_{7.3}$ packets can be *decoded by* d_1 during Phase 2.s_2, see (12). The first super-column of Table I thus contains all the $\mathbf{W}_1$ packets that still need to be sent to d_1. Symmetrically, the second super-column of Table I also contains all the $\mathbf{W}_2$ packets that still need to be sent to d_2.

Before proceeding, we define an extended group $\overline{A_3} \triangleq A_3 \cup A_{4.1} \cup B_{7.2} \cup B_{7.3} \cup A_{7.2}^{\text{sum}}$, which contains the packets that are available at s_2 and belong to the first super-column of Table I. Symmetrically, define $\overline{B_3} \triangleq B_3 \cup B_{4.1} \cup A_{7.2} \cup A_{7.3} \cup B_{7.2}^{\text{sum}}$. Recall that the main challenge of the problem is that s_1 can only acquire $\mathbf{W}_2$ packets through weak overhearing $q < \frac{p}{3-p}$. Therefore, s_1 generally does not have enough $\mathbf{W}_2$ packets that can be mixed with the big amount of A_3, $A_{4.1}$ to $A_{4.4}$ packets it has (since s_1 having all $\mathbf{W}_1$ packets to begin with). As a result, at least heuristically s_2 *should refrain from using any of the $\overline{B_3}$ packets and leave all of them to s_1 since the $\overline{B_3}$ packets are those precious few $\mathbf{W}_2$ packets that are actually overheard by s_1 and can be mixed with the $\mathbf{W}_1$ packets at* s_1. Symmetrically, s_1 should refrain from using any of the $\overline{A_3}$ packets and leave all of them to s_2. From the above reasoning, s_1 should mix only $A_{4.4}$ and $\overline{B_3}$ packets while s_2 should mix only $B_{4.4}$ and $\overline{A_3}$ packets.

Based on the above intuition, we devise the following subroutines that treat each packet group as a queue.

---

§ MIX $A_{4.4}$ AND $\overline{B_3}$

Source s_1 transmits a linear sum of two packets: the head-of-the-line (HOTL) packet of $A_{4.4}$ and the HOTL packet of $\overline{B_3}$, the latter can already be a linear sum of multiple packets if the HOTL packet happens to be a $B_{7.2}^{\text{sum}}$ packet.
if d_1 receives the packet **then**
    Let the HOTL packet of $A_{4.4}$ leave the system.
else if s_2 receives the packet **then**
    Let the HOTL packet of $A_{4.4}$ leave the system. Let the current packet, a linear sum of an $A_{4.4}$ and a $\overline{B_3}$ packets, enter the queue $\overline{A_3}$.
end if
if d_2 receives the packet **then**
    Let the HOTL packet of $\overline{B_3}$ leave the system.
end if

The subroutine "MIX $B_{4.4}$ AND $\overline{A_3}$" is symmetric to the above subroutine, i.e., s_2 would transmit a linear sum of a packet in $B_{4.4}$ and a packet in $\overline{A_3}$. We now describe the next two phases.

---

§ PHASE 3

Phase 3 runs for t_{P3} time slots, where $t_{\text{P3}} \triangleq \frac{|A_{4.4}|-|B_{4.4}|}{p+q(1-p)}$.
For each time slot, s_1 transmits the HOTL packet of $A_{4.4}$.
if d_1 receives the packet **then**
    Let the HOTL packet of $A_{4.4}$ leave the system.

	effectively a $\mathbf{W}_1$ packet heard by d_2 but not by d_1						effectively a $\mathbf{W}_2$ packet heard by d_1 but not by d_2					
packets available at s_1	A_3	$A_{4.1}$	$A_{4.2}$	$A_{4.3}$	$A_{4.4}$		B_3	$B_{4.1}$	$A_{7.2}$ and $A_{7.3}$			$B_{7.2}^{\text{sum}}$
packets available at s_2	A_3	$A_{4.1}$	$B_{7.2}$ and $B_{7.3}$			$A_{7.2}^{\text{sum}}$	B_3	$B_{4.1}$	$B_{4.2}$	$B_{4.3}$	$B_{4.4}$	

TABLE I
THE PACKETS AT s_1 AND s_2 THAT CAN POTENTIALLY BE USED FOR NC.

5: **else if** s_2 receives the packet **then**
6: Let the HOTL packet of $A_{4.4}$ leave the queue $A_{4.4}$ and enter another queue $\overline{A_3}$.
7: **end if**

§ PHASE 4

1: **while** $\min(|A_{4.4}|, |B_{4.4}|) > 0$ **do**
2: **while** $|\overline{A_3}| > 0$ **do**
3: Source s_2 runs the subroutine "MIX $B_{4.4}$ AND $\overline{A_3}$".
4: **end while**
5: **while** $|\overline{B_3}| > 0$ **do**
6: Source s_1 runs the subroutine "MIX $A_{4.4}$ AND $\overline{B_3}$".
7: **end while**
8: **end while**

The intuition of Phases 3 and 4 is as follows. We first take a look at the subroutine "MIX $A_{4.4}$ AND $\overline{B_3}$". Suppose we send a linear sum of an $A_{4.4}$ and a $\overline{B_3}$ packet. When d_2 receives it, d_2 can decode the $\overline{B_3}$ packet by subtracting the $A_{4.4}$ packet it received in Phase 1.s_1. Therefore, the $\overline{B_3}$ packet can leave the system as specified in Line 8 of MIX $A_{4.4}$ AND $\overline{B_3}$. Similarly, when d_1 receives it, d_1 can decode the $A_{4.4}$ packet since d_1 has, effectively, heard the $\overline{B_3}$ packet, see our previous discussion. Therefore, the $A_{4.4}$ packet can leave the system as specified in Line 3 of MIX $A_{4.4}$ AND $\overline{B_3}$. When the packet is heard by s_2 but not by d_1 (Line 4), s_2 now has the knowledge about the linear sum. Therefore, the linear sum can be viewed as in the category "available at s_2 and being a $\mathbf{W}_1$ packet that is effectively heard by d_2 but not by d_1."[4] Hence, we add the linear sum to the queue $\overline{A_3}$ as in Line 5 of MIX $A_{4.4}$ AND $\overline{B_3}$. The above intuition can be made rigorous through the *packet evolution* arguments in [9].

We now take a look at Phase 4. In Line 3 of Phase 4, s_2 combines the $B_{4.4}$ and the $\overline{A_3}$ packets until s_2 uses up all $\overline{A_3}$ packets (see Line 2). During this process of executing MIX $B_{4.4}$ AND $\overline{A_3}$, we have also created new $\overline{B_3}$ packets that can be used in Line 6 of Phase 4. After s_1 uses up all $\overline{B_3}$ packets (see Line 5), new $\overline{A_3}$ packets have been created during the process. We are now ready to let s_2 execute MIX $B_{4.4}$ AND $\overline{A_3}$ again based on the newly generated $\overline{A_3}$. The above iteration between s_2 executing MIX $B_{4.4}$ AND $\overline{A_3}$ and s_1 executing MIX $A_{4.4}$ AND $\overline{B_3}$ continues until one of $A_{4.4}$ and $B_{4.4}$ becomes empty, see Line 1 of Phase 4.

We are now ready to discuss the intuition behind Phase 3. One key factor that guarantees the success of Phase 4 is to ensure that the numbers of the $A_{4.4}$, $B_{4.4}$, $\overline{A_3}$, and $\overline{B_3}$ packets are *well-balanced* so that when the iteration of Phase 4 finally stops ($\min(|A_{4.4}|, |B_{4.4}|) = 0$) *all four queues* $A_{4.4}$, $B_{4.4}$, $\overline{A_3}$, *and* $\overline{B_3}$ *are nearly empty*. Note that after finishing Phase 2.s_2, we usually have too many $A_{4.4}$ and too few $\overline{B_3}$ packets at s_1 since $R_1 = p - q \geq R_2 = q(2-p)$. We thus use Phase 3 to send $A_{4.4}$ packets alone, which reduces the amount of $A_{4.4}$ packets without using any $\overline{B_3}$ packets. One can prove that after Phase 3, the numbers of the $A_{4.4}$, $B_{4.4}$, $\overline{A_3}$, and $\overline{B_3}$ packets are *well-balanced* and we are ready for Phase 4.

What remains to be proven is that for any $\epsilon > 0$ there exists a sufficiently large n such that with probability no less than $(1-\epsilon)$, the above 4-phased scheme can finish transmission in $n(1+\epsilon)$ time slots without any errors[5] and in the end each d_i can decode at least $n(R_i - \epsilon)$ of the desired $\mathbf{W}_i$ packets. The detailed analysis is omitted due to the lack of space.

IV. CONCLUSION

This work has characterized the full capacity region of two symmetric nearby erasure channels with network-wide channel state feedback. This work was supported in parts by NSF grants CCF-0845968 and CNS-0905331.

REFERENCES

[1] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 789–804, March 2006.

[2] M. Gatzianas, S. Bidokhti, and C. Fragouli, "Feedback-based coding algorithms for broadcast erasure channels with degraded message sets," in *Proc. 8th Workshop on Network Coding, Theory, & Applications (NetCod)*. Cambridge, USA, June 2012.

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

[4] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback — capacity and algorithms," in *Proc. 5th Workshop on Network Coding, Theory, & Applications (NetCod)*. Lausanne, Switzerland, June 2009, pp. 54–61.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*, 2006.

[6] W. Kuo and C.-C. Wang, "On the capacity of 2-user 1-hop relay erasure networks — the union of feedback, scheduling, opportunistic routing, and network coding," in *Proc. IEEE Int'l Symp. Inform. Theory*. Saint Petersburg, Russia, August 2011, journal version is currently under preparation.

[7] X. Lin and N. Shroff, "Utility maximization for communication networks with multi-path routing," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 766–781, May 2006.

[8] V. Prabhakaran and P. Viswanath, "Interference channels with source cooperation," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 156–186, January 2011.

[9] C.-C. Wang, "Capacity of 1-to-K broadcast packet erasure channels with channel output feedback," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 957–988, February 2012.

[10] C.-C. Wang and D. Love, "Linear network coding capacity region of 2-receiver MIMO broadcast packet erasure channels with feedback," in *Proc. IEEE Int'l Symp. Inform. Theory*. Boston, USA, July 2012.

[11] R. Wu, V. Prabhakaran, and P. Viswanath, "Interference channels with half duplex source cooperation," in *Proc. IEEE Int'l Symp. Inform. Theory*. Austin, Texas, USA, June 2010, pp. 375–379.

[4] An alternative explanation is as follows. The event s_2 overhearing the sum of an $A_{4.4}$ and a $\overline{B_3}$ packet effectively moves one entry of $A_{4.4}$ from the first row (available only at s_1) to the second row (available also at s_2).

[5] For example, we need to prove that when Phase 2.s_1 stops, $l_4 \leq |A_4|$ with close-to-one probability.