Broadcast Erasure Channel with Feedback - Capacity and Algorithms

Leonidas Georgiadis[1]
Dept. of Elect. & Comput. Eng.
Aristotle University of Thessaloniki
Thessaloniki, 54006, GREECE
Email: leonid@auth.gr

Leandros Tassiulas[1]
Dept. of Comput. Eng. & Telec.
University of Thessaly
Volos, GREECE
Email: leandros@uth.gr

Abstract — **We consider the two-user broadcast erasure channel where feedback in the form of ack messages is fed back to the transmitter. We provide an upper bound to the capacity region of this system. We then present two algorithms whose rate region (information bits per transmitted bit) becomes arbitrarily close to the upper bound for large packet sizes. The first algorithm relies on random coding techniques while the second relies only on XOR operations between pairs of packets. Complexity and feedback information tradeoffs for the two algorithms are discussed. For the case where, in addition to traffic destined exclusively to either one of the users there is additional multicast traffic, we present an algorithm that shows that the rate region of the system can be increased by allowing intersession coding. Finally, for the case where there are random arrivals to the system we present an algorithm, based on the previous algorithms, whose stability region gets close to the capacity region for reasonably large packet sizes. The latter algorithm operates without knowledge of arrival process and channel statistics.**

I Introduction

The capacity region of the 2-user "stochastically degraded" Broadcast channel without feedback is well known [1]. It is also known [2] that for the "physically degraded" broadcast channel, a notion stronger than "stochastically degraded", feedback does not increase capacity.

The Erasure Broadcast channel, where a message transmitted to a user is either received without error or erased, can be shown to be stochastically degraded and its capacity region without feedback in information bits per transmission is given by

$$\mathcal{R} = \left\{ (r_1, r_2) \geq 0 : \frac{r_1}{1-\varepsilon_1} + \frac{r_2}{1-\varepsilon_2} \leq L \right\},$$

where L is the packet length in bits. This result has been generalized to N users with multiple inputs [3]. The form of the region implies that this capacity can be achieved by separately encoding each of the user sessions and by timesharing the two encoding schemes.

The Broadcast Erasure channel is not physically degraded in general, and hence the capacity region with feedback may increase. In this paper we study the Broadcast Erasure channel with feedback. Due to the form of the received messages, feedback in this case consists of simply announcing (both users) to the transmitter whether a transmitted message has been received correctly (i.e., not erased) or not. We develop an upper bound for the capacity region of the system with feedback when there are two unicast sessions, one for each user. We then present an algorithm, based on random linear coding of a number of packets from both unicast sessions, whose rate region in information bits per transmitted bit becomes arbitrarily close to the developed bound as L increases. We also consider the case where there is in addition a multicast session containing information which both users must receive and show that inter-coding between the three sessions (user 1, user 2, multicast) increases the rate region beyond the one obtained by timesharing. Next, we present an algorithm that uses only XOR operations between packets whose rate region is also getting arbitrarily close to the capacity region of the system when there are only unicast sessions. Finally, we show that in a system with random arrivals the developed algorithms can be used to design algorithms that stabilize the system whenever the arrival rates are within the rate region of the system, without relying on knowledge of arrival process or channel parameter statistics.

The erasure channel, single and multihop, has received a lot of attention lately, mainly due to the applicability of this model to packet transmission and packet dropping mechanisms in the Internet and in wireless communication protocols. The capacity of wireless erasure networks for a single multicast session is presented in [4] and a capacity achieving algorithm for the same model, using random linear coding has been presented in [5]. The benefits of using feedback for network coding and related algorithms are presented in [6], [7], [8], [9].

The previous works concentrate mainly on single unicast or multicast transmission. In the current work we concentrate on the two-user broadcast erasure channel and consider multiple unicast as well as combination of unicast and multicast transmission. The study was motivated by the feedback schemes proposed in [10] and the

[1] This work is supported by the project N-CRAVE (FP7 - 215252) funded by the European Commission.

algorithm using XOR operations is in the spirit of the algorithms proposed in [11], [12].

II System Model

Consider a slotted system where messages (packets) of length L bits are transmitted within each slot. We assume that the unit of time is the time needed to transmit a bit. Time slot $[(l-1)L, lL)$, $l = 1, 2, ...$ is referred to as "slot l".

The system consists of a single transmitter and two users. A packet transmitted in slot l is broadcasted to both users. At the end of slot l, a user may either receive the packet correctly, or the packet may be lost or dropped by the user, in which case we say that an erasure occurred - we denote this event by the symbol E. Define

$$Z_{i,l} = \begin{cases} 1 & \text{if an erasure occurs for user } i \text{ in the } l\text{th slot} \\ 0 & \text{otherwise} \end{cases}$$

The random pair $(Z_{1,l}, Z_{2,l})$, $l = 1, 2....$ has an arbitrarily distribution, and the sequence $\{(Z_{1,l}, Z_{2,l})\}_{l=1}^{\infty}$ consists of independent identically distributed pairs. Denote by (Z_1, Z_2) a generic pair of $\{0, 1\}$ random variables having the distribution of $(Z_{1,l}, Z_{2,l})$ and define

$$\Pr(Z_1 = 1) = \varepsilon_1, \ \Pr(Z_2 = 1) = \varepsilon_2, \ \Pr(Z_1 Z_2 = 1) = \varepsilon_{12}.$$

In the sequel, to avoid trivial cases we assume that $\varepsilon_i < 1$, $i = 1, 2$.

According to the definitions above, if packet X_l is transmitted in time slot l then the output received by user i is,

$$Y_{i,l} = Z_{i,l} E + (1 - Z_{i,l}) X_l. \tag{2}$$

At the end of slot l user $i = 1, 2$ provides feedback to the transmitter as to whether the packet has been received or erased, which according to (2) is equivalent to having each user inform the transmitter about the value of $Y_{i,l}$.

In the next section we provide an upper bound to the information theoretic capacity of this channel

III Upper Bound to Channel Capacity

In this section we use the definitions and notation in [1] and [2] . A generic sequence $X_1, ..., X_n$ is denoted by X^n.

Definition: Let $\mathcal{W}_i$, $i = 1, 2$, be a message index set of size 2^{nR_i} and let $(W_1, W_2) \in \mathcal{W}_1 \times \mathcal{W}_2$. An $\left(2^{nR_1}, 2^{nR_2}, n\right)$ code for the broadcast channel with feedback consist of

- an encoder that at slot l transmits message X_l which is a function of $\{(W_1, W_2), X_m, Y_{1,m}, Y_{2,m}\}_{m=1}^{l-1}$- when $l = 1$, X_1 is a function of (W_1, W_2) only.
- two decoders
$$g_i : Y_i^n \rightarrow \mathcal{W}_i, i = 1, 2.$$

Denote by C the channel under consideration and define a modified channel $\widehat{C}$ with $\widehat{Z}_{1,l} = Z_{1,l} Z_{2,l}$ and $\widehat{Z}_{2,l} = Z_{2,l}$. For channel $\widehat{C}$ an erasure to output 1 occurs only when erasures to both outputs in channel C occur. The following hold.

Lemma 1 *Channel $\widehat{C}$ is physically degraded.*

Lemma 2 *If $\mathcal{C}_f$ and $\widehat{\mathcal{C}}_f$ are the feedback capacity regions of channels C and $\widehat{C}$ respectively, then*

$$\mathcal{C}_f \subseteq \widehat{\mathcal{C}}_f.$$

Theorem 3 *Let*

$$\begin{aligned} \mathcal{R}_1 &= \left\{ (r_1, r_2) \geq 0 : \frac{r_1}{1 - \varepsilon_1} + \frac{r_2}{1 - \varepsilon_{12}} \leq L \right\} \\ \mathcal{R}_2 &= \left\{ (r_1, r_2) \geq 0 : \frac{r_1}{1 - \varepsilon_{12}} + \frac{r_2}{1 - \varepsilon_2} \leq L \right\} \end{aligned} .$$

Then

$$\mathcal{C}_f \subseteq \mathcal{R}_1 \cap \mathcal{R}_2$$

Proof. According to Lemma 1 channel $\widehat{C}$ is physically degraded. Hence its feedback capacity is the same as the capacity without feedback, i.e., $\widehat{\mathcal{C}}_f = \mathcal{R}_2$ and according to Lemma 2, $\mathcal{C}_f \subseteq \widehat{\mathcal{C}}_f = \mathcal{R}_2$. Reversing the roles of Z_1 and Z_2,we also get $\mathcal{C}_f \subseteq \mathcal{R}_1$ and the result follows. ■

IV Algorithm Approximating Capacity for Large L

In this section we present a variable length coding algorithm whose rate region is very close to the region $\mathcal{R}_1 \cap \mathcal{R}_2$ for reasonably large packet sizes. The algorithm operates in phases. In Phases 1 and 2, the transmitter transmits packets destined to user 1 and 2 respectively, retransmitting a packet until *at least one* of the users receives the packet. Phase 3 consists of transmitting coded packets, where each packet is a random linear combination of packets destined to either one of the users. The latter packets are determined by the feedback received during Phases 1 and 2.

Algorithm I

Three bits, b_1, b_2, b_3 from each packet are reserved to convey control information to the users. These bits are set as follows

- $b_1 = 0, b_2 = 0$: packet destined to user 1 (Phase 1). Bit b_3 indicates control information to be specified below.
- $b_1 = 0$, $b_2 = 1$: packet destined to user 2 (Phase 2). Bit b_3 indicates control information to be specified below.

- $b_1 = 1$: phase number 3. Bits b_2, b_3 indicate control information to be specified below.

Two integers k_1 and k_2 are selected. These integers are known to both users and denote respectively the number of packets from each user that must be transmitted.

1. **Phase 1:**

   - **Transmitter:** Transmits k_1 destination 1 packets as follows

   (a) Set $b_1 = 0, b_2 = 0$, hence the packet is destined to user 1

   (b) A packet is retransmitted until it is received by *at least one* of the destinations.

   (c) If a packet is received by user 2 and erased at user 1, it is placed in a queue Q_1. Hence Q_1 contains all packets destined to 1 and seen only by 2.

   (d) Set $b_3 = 1$ if the *last correctly received* packet by user 2 has been erased at user 1**.** Else $b_3 = 0$. For the first transmitted packet $b_3 = 0$.

   - **User 1:** If a packet is received correctly, the user sets a counter $M_1 \leftarrow M_1 + 1$, where initially $M_1 = 0$.Hence M_1 indicates the number of correctly received packets by user 1.The appropriate feedback is sent to the transmitter at the end of every slot.

   - **User 2:** Upon correct reception, if $b_3 = 1$,the user stores the *last correctly received* packet in a local queue Q_1^2 (this packet has been erased at user 1). Else the user discards the last correctly received packet (since this packet has also be received correctly by user 1). Note that this way, at any time, Q_1^2, contains the packets of Q_1 with sole exception the last correctly received packet by user 2 during phase 1. In addition, the appropriate feedback is sent to the transmitter at the end of every slot.

2. **Phase 2**: Same rules as Phase 1, with the roles of user 1 and 2 reversed.

3. **Phase 3:** Let K_i^r be the number of packets in queue Q_i, $i = 1, 2$. Note that since Q_1 contains packets erased at user 1 and correctly received by user 2, and since every packet in phase 1 is received by at least one of the users, it holds, $K_1^r = k_1 - M_1$. Hence user 1 knows K_1^r. Similarly, user 2 knows K_2^r.

   - **Transmitter:** The following actions are taken

   (a) Set $b_1 = 1$, indicating phase 3. If the last correctly received packet by user 2 during phase 1 has been erased at user 1, then $b_2 = 1$. Else $b_2 = 0$. Bit b_3 is set in the same manner, by replacing user 2 with user 1 and phase 1 with phase 2.

   (b) The transmitter transmits random linear combinations of the $L - 3$ *information bits* of the $K_1^r + K_2^r$ packets in queues Q_i, $i = 1, 2$- bits b_1, b_2, b_3 are not encoded. The coding coefficients are generated by a process known apriori to both users and hence do not need to be transmitted.

   - **User 1:** Upon correct reception of the first packet, the user knows, by reading b_3, whether the last correctly received packet in phase 1 has been erased or not at user 2. If this packet is erased at user 2, node 1 adds the packet to local queue Q_2^1. Hence, at this time $Q_2^1 = Q_2$, and user 1 can infer the value of K_2^r. Since it also knows the value of K_1^r, the user knows the number of packets that compose the linear combination of each received packet. Since the user knows the content of the K_2^r packets in queue Q_2, only the K_1^r packets destined to user 1 need to be recovered from the observed linear combinations. The user observes the received packets until is has enough information (dimensions) to decode the K_1^r packets. Feedback is sent to the transmitter *only* when this decoding is successful.

   - **User 2:** The corresponding actions as user 1.

Note: The random linear coding in phase 3 can be replaced by other coding algorithms. We use the proposed method for definiteness.

A Analysis of the algorithm

We present here the main ideas of the analysis skipping the technical details. Assume k_1, k_2 large. The number, N_i, of transmissions needed to complete phase i of the algorithm, i.e., for a user i packet to be correctly received by at least one of the destinations, is about $N_i = \frac{k_i}{1-\varepsilon_{12}}, i = 1, 2$. The number, M_1, of user 1 packets transmitted during phase 1 that were delivered to user 1 is $M_1 = k_1 \frac{1-\varepsilon_1}{1-\varepsilon_{12}}$. Defining similarly $M_{2,}$ we have $M_2 = k_2 \frac{1-\varepsilon_2}{1-\varepsilon_{12}}$, and hence for $i = 1, 2$, $K_i^r = k_i - M_i = k_i \frac{\varepsilon_i - \varepsilon_{12}}{1-\varepsilon_{12}}$.

Consider now phase 3. Since user 1 knows K_2^r out of the $K_1^r + K_2^r$ packets involved in each of the random linear combinations received the user will be able to recover the K_2^r packets with high probability after receiving correctly about $G_1^r = \frac{2^{L-3}}{2^{L-3}-1} K_1^r$ such linear combinations - the coefficient $\frac{2^{L-3}}{2^{L-3}-1}$ accounts for the possibility of receiving "non-innovative" packets. The number of transmissions needed in order to correctly receive these G_1^r packets is $G_1^r / (1 - \varepsilon_1)$. A similar argument holds for user 2. Hence the number of transmissions needed so that both users recover their undelivered packets during phase 3 is about,

$$N_3 = \max\left\{\frac{G_1^r}{1-\varepsilon_1}, \frac{G_2^r}{1-\varepsilon_2}\right\} \tag{3}$$

$$= \frac{2^{L-3}}{2^{L-3}-1}\max\left\{\frac{k_1\left(\varepsilon_1-\varepsilon_{12}\right)}{\left(1-\varepsilon_1\right)\left(1-\varepsilon_{12}\right)}, \frac{k_2\left(\varepsilon_2-\varepsilon_{12}\right)}{\left(1-\varepsilon_2\right)\left(1-\varepsilon_{12}\right)}\right\}.$$

We can now compute the rates for the two destinations. The rate, in packets received correctly per packet transmission, for destination 1 is,

$$\begin{aligned} r_1 &= \frac{k_1}{N_1+N_2+N_3} \\ &= \frac{\phi\left(1-\varepsilon_{12}\right)}{1+\frac{2^{L-3}}{2^{L-3}-1}\max\left\{\phi\frac{\varepsilon_1-\varepsilon_{12}}{(1-\varepsilon_1)}, (1-\phi)\frac{\varepsilon_2-\varepsilon_{12}}{(1-\varepsilon_2)}\right\}} \\ &\geq \frac{2^{L-3}-1}{2^{L-3}}\frac{\phi\left(1-\varepsilon_{12}\right)}{1+\max\left\{\phi\frac{\varepsilon_1-\varepsilon_{12}}{(1-\varepsilon_1)}, (1-\phi)\frac{\varepsilon_2-\varepsilon_{12}}{(1-\varepsilon_2)}\right\}} \end{aligned} \tag{4}$$

where $\phi = \frac{k_1}{k_1+k_2}$. Similarly, for destination 2,

$$r_2 \geq \frac{2^{L-3}-1}{2^{L-3}}\frac{(1-\phi)\left(1-\varepsilon_{12}\right)}{1+\max\left\{\phi\frac{\varepsilon_1-\varepsilon_{12}}{(1-\varepsilon_1)}, (1-\phi)\frac{\varepsilon_2-\varepsilon_{12}}{(1-\varepsilon_2)}\right\}} \tag{5}$$

Since 3 of the bits of each packet are used for control information, it follows from (4) and (5), after eliminating ϕ, that the region of achievable rates $\mathcal{R}^a$ in bits per packet transmission is at least,

$$\mathcal{R}^a = \left\{\left(r_1, r_2\right) \geq 0 : M^a\left(r_1,r_2\right) \leq \frac{2^{L-3}-1}{2^{L-3}}\left(L-3\right)\right\},$$

where

$$M^a\left(r_1,r_2\right) = \max\left\{\frac{r_1}{1-\varepsilon_1}+\frac{r_2}{1-\varepsilon_{12}}, \frac{r_1}{1-\varepsilon_{12}}+\frac{r_2}{1-\varepsilon_2}\right\}.$$

For large L, region $\mathcal{R}^a$ is a close approximation of the region defined by Theorem 3. In fact, in units of information bits per transmitted bit (i.e., scaling by L) the region $\mathcal{R}^a$ becomes arbitrarily close to the bound given by Theorem 3 and the rate of approximation is $O\left(1/L\right)$.

V Rate Region Including Multicast Information

In this section we assume that in addition to information destined to each of the nodes, there is multicast information that needs to be transmitted to both destinations. Let r_{12} be the rate of multicast information. For N users, when there is no feedback and there is a single multicast session, it is known that the channel capacity region is $\left\{r : 0 \leq r \leq \frac{1}{1-\max_{n=1,\ldots,N}\{\varepsilon_i\}}\right\}$. This result has been generalized to the network case [4]. For two users, by timesharing separate encodings for the three sessions (user 1, user 2, multicast), rate triplets $\boldsymbol{r} = (r_1, r_2, r_{12})$ belonging to the region

$$\widetilde{\mathcal{R}} = \left\{\boldsymbol{r} \geq \mathbf{0} : \frac{r_1}{1-\varepsilon_1}+\frac{r_2}{1-\varepsilon_2}+\frac{r_{12}}{1-\max\left\{\varepsilon_1,\varepsilon_2\right\}} \leq L\right\},$$

are achievable. We show in this section that when feedback is allowed, intersession coding involving all three sessions can increase the rate region.

If we employ Algorithm I to transmit user 1 and user 2 information, employ a separate code for the multicast information and timeshare between these codings the resulting rate region $\widetilde{\mathcal{R}}^a$ is

$$\widetilde{\mathcal{R}}^a = \left\{\boldsymbol{r} \geq \mathbf{0} : \widetilde{M}^a\left(r_1, r_2, r_{12}\right) \leq \frac{2^{L-3}-1}{2^{L-3}}\left(L-3\right)\right\}, \tag{6}$$

where

$$\begin{gathered} \widetilde{M}^a\left(r_1, r_2, r_{12}\right) = \\ \max\left\{\frac{r_1}{1-\varepsilon_1}+\frac{r_2}{1-\varepsilon_{12}}+\frac{r_{12}}{1-\max\left\{\varepsilon_1,\varepsilon_2\right\}}\right., \\ \left.\frac{r_1}{1-\varepsilon_{12}}+\frac{r_2}{1-\varepsilon_2}+\frac{r_{12}}{1-\max\left\{\varepsilon_1,\varepsilon_2\right\}}\right\}. \end{gathered}$$

Consider now Algorithm II:

Algorithm II:

Let k_1, k_2, k_{12} be the number of packets to be transmitted from each of the three sessions (user 1, user 2 and multicast). These numbers are known to the users.

- Employ Phases 1,2 of Algorithm I
- In Phase 3 employ linear random coding of packets K_1^r, K_2^r, k_{12}, i.e., include the multicast session packets in the process.

An analysis similar to the analysis of Algorithm I shows that the rate region of Algorithm II is

$$\widetilde{\mathcal{R}}^f = \left\{\left(r_1, r_2, r_{12}\right) \geq 0 : \widetilde{M}^f\left(r_1, r_2, r_{12}\right) \leq \frac{2^{L-3}-1}{2^{L-3}}\left(L-3\right)\right\}, \tag{7}$$

where

$$\begin{aligned} \widetilde{M}^f\left(r_1, r_2, r_{12}\right) \quad = \quad & \max\left\{\frac{r_1}{1-\varepsilon_1}+\frac{r_2}{1-\varepsilon_{12}}+\frac{r_{12}}{1-\varepsilon_1}\right., \\ & \left.\frac{r_1}{1-\varepsilon_{12}}+\frac{r_2}{1-\varepsilon_2}+\frac{r_{12}}{1-\varepsilon_2}\right\}. \end{aligned}$$

By comparing (6) and (7) we see that in general intersession coding of all three sessions increases the rate region when feedback is used.

Figure 1 shows the rate regions with and without feedback and with and without intersession coding. The region $OACE$ is the rate region without feedback. The region $OABCE$ is the rate region when feedback is used,

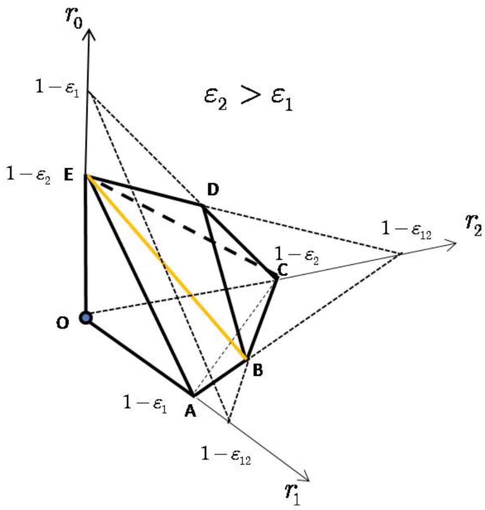

Figure 1: Rate regions for Algorithms I and II

intersession coding between user 1 and user 2 session is allowed, but separate coding is employed for the multicast session. The region $OABCDE$ is the rate region when intersession coding is employed. Notice that the points A, B, D, E lie on the same plane.

VI Algorithm Using Only XOR Operations

In this section we provide an algorithm that codes using only XOR operations between packets and achieves the performance of Algorithm I.

Algorithm III

Two integers k_1 and k_2 are selected. These integers are known to both users and denote respectively the number of packets from each user that must be transmitted. Three bits b_1, b_2, b_3 are reserved for control purposes. These bits are set as follows.

- b_1 indicates the phase of the algorithm (see below)
- b_2 indicates whether the transmitted packet is the result of an XOR operation, i.e., $b_2 = 0$: non-XORed packet, $b_2 = 1$: XORed packet
- b_3 indicates the user to which the packet is destined (if it is not the result of and XOR operation), i.e. $b_3 = 0$: user 1 packet, $b_3 = 1$: user 2 packet

The following buffers are maintained

- **Transmitter:** Two buffers Q_1, Q_2. Initially these buffers are empty. At the beginning of slot l, buffer Q_1 is either empty or contains a user 1 packet that has been transmitted at some prior time, erased at user 1 and received correctly by user 2. Buffer Q_2 is similarly defined with the roles of 1 and 2 reversed. As will be seen from the description of the algorithm, these buffers contain at most one packet. We denote by $Q_{i,l}$ the contents of buffer Q_i at time l.
- **User** 1: A buffer Q_2^1 containing at most one packet. Initially Q_2^1 is empty. We denote by $Q_{2,l}^1$ the contents of buffer Q_2^1 at time l.
- **User** 2: A buffer Q_1^2 containing at most one packet. Initially Q_1^2 is empty. We denote by $Q_{1,l}^2$ the contents of buffer Q_1^2 at time l.

The algorithm consists of two phases as follows. Let $R_i\left(l\right)$ be the number of user i packet left to be transmitted at the beginning of slot l.

- **Phase 1:** This phase lasts as long as $R_i\left(l\right) > 0$ for both $i = 1, 2$.

  Transmitter: For all packets in this phase, $b_1 = 0$.

1. If $Q_{1,l} = \varnothing, Q_{2,l} = \varnothing$ then transmit a user 1 packet in slot l, setting $b_2 = 0, b_3 = 0$; At the end of slot l,
   (a) If $Z_{1,l} = 1, Z_{2,l} = 0$ then place the transmitted packet in buffer Q_1
2. If $Q_{1,l} \neq \varnothing, Q_{2,l} = 0$ then transmit a user 2 packet in slot l, setting $b_2 = 0, b_3 = 1$; At the end of slot l,
   (a) If $Z_{1,l} = 0, Z_{2,l} = 1$ then place the transmitted packet in buffer Q_2
3. If $Q_{1,l} = 0, Q_{2,l} \neq \varnothing$ then transmit a user 1 packet in slot l,setting $b_2 = 0, b_3 = 0$; At the end of slot l,
   (a) If $Z_{1,l} = 1, Z_{2,l} = 0$ then place the transmitted packet in a buffer Q_1
4. If $Q_{1,l} \neq \varnothing$, $Q_{2,l} \neq \varnothing$ then XOR the *information part* of the two user packets (i.e., $L - 3$ bits) in buffers Q_1, Q_2 and transmit the result in slot l, setting $b_2 = 1$; At then end of slot l,
   (a) If $Z_{1,l} = 1, Z_{2,l} = 0$ then remove the packet in buffer Q_1. Else
   (b) If $Z_{1,l} = 0, Z_{2,l} = 1$ then remove the packet in buffer Q_2. Else
   (c) If $Z_{1,l} = 0, Z_{2,l} = 0$ then remove the packet from both buffers Q_1, Q_2.

- **User 1**: At the end of each slot the user sends the appropriate (ack, nack) feedback to the transmitter.

  1. If at the end of slot l user 1 receives a packet destined to user 2 ($b_2 = 0, b_3 = 1$), it places this packet in Q_2^1, replacing any other packet that may exist.

2. If at the end of slot l user 1 receives an XORed packet ($b_2 = 1$) then it XORs this packet with the packet stored in buffer Q_2^1 - the result is a user 1 packet.
3. If at the end of slot l user 1 receives a packet destined to itself ($b_2 = 0, b_3 = 0$) it accepts the packet.

- **User 2**: Similar actions, with the roles of 1 and 2 reversed.
- **Phase 2:** At this phase, packets destined to only one of the users, say user 1, are left to be transmitted. The transmitter sets $b_1 = 1$, $b_3 = 0$ and retransmits the packets until it is ensured, through the received feedback that all packets have been received correctly by user 1.

In the next section we discuss the correctness and performance analysis of this algorithm.

A Analysis of Algorithm III

We must show first that the operation of the algorithm is correct. From the description of the algorithm it follows that at any time during phase 1, Q_1 and Q_2 contain at most one packet. However, the following needs attention. Note that according to the algorithm, an XORed packet q sent at time slot l is of the form $q = Q_{1,l} \oplus Q_{2,l}$. When a user, say user 1, receives q in slot l, it performs the operation $\widetilde{q}_1 = q \oplus Q_{2,l}^1$. In order for user 1 to recover $Q_{1,l}$,it must hold that $\widetilde{q}_1 = Q_{1,l}$, i.e., it must be ensured that whenever an XORed packet is received by user 1, $Q_{2,l}^1 = Q_{2,l}$. This is not true always since it can be seen that if the transmitted packet is not an XORed packet then it may happen that $Q_{2,l}^1 \neq Q_{2,l}$. However, from the description of the algorithm the following lemma can be seen to hold.

Lemma 4 *During Phase 1, if at the end of time slot l user i, say user 1, receives the XORed packet $q = Q_{1,l} \oplus Q_{2,l}$ then $Q_{2,l} = Q_{2,l}^1$.*

We now proceed with an outline of the analysis of the algorithm. During Phase 1 the operation of the algorithm is described as a reward Markov Chain with 4 states. The states describe the content of buffers Q_1, Q_2, as follows.

$$\begin{aligned} A &= (Q_1, Q_2) = (\varnothing, \varnothing)\,,\ B = (1, \varnothing)\,, \\ C &= (\varnothing, 1)\,,\ D = (1, 1)\,. \end{aligned}$$

A state transition occurs each time a transmission takes place. The rewards consist of pairs $(\rho_{ss'}(1), \rho_{ss'}(2))$ where for a transition from state s to state s', $\rho_{ss'}(i)$ corresponds to the number of user i packets received *correctly by user* i during the corresponding transmission. This number may be random and we denote with $\overline{\rho}_{ss'}(i)$ its average value. The complete state diagram is described in Figure 2. A number i in a square indicates that a user i packet has been received correctly by user i . The formulas next to each arrow indicate transition probabilities.

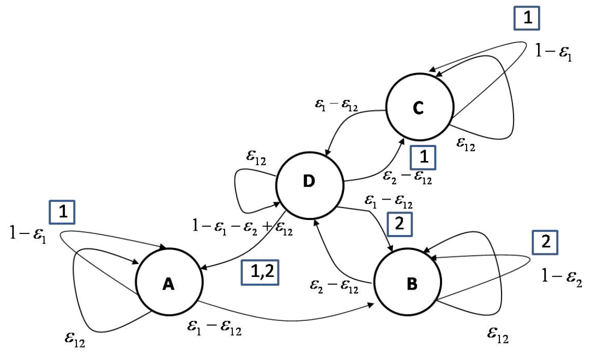
1
$1-\varepsilon_1$
C
$\varepsilon_1 - \varepsilon_{12}$
ε_{12}
ε_{12}
D
1
$\varepsilon_2 - \varepsilon_{12}$
$\varepsilon_1 - \varepsilon_{12}$
1
$1-\varepsilon_1$
$1-\varepsilon_1-\varepsilon_2+\varepsilon_{12}$
2
2
A
1,2
$\varepsilon_2 - \varepsilon_{12}$
B
$1-\varepsilon_2$
ε_{12}
$\varepsilon_1 - \varepsilon_{12}$
ε_{12}

Figure 2: The Markov Chain describing Algorithm III

We explain next how the transitions and rewards are defined when in state A. In this state, according to the algorithm user 1 packets are transmitted. From this state, a transition to the same state occurs if either the transmitted packet is erased at both users, an event of probability ε_{12}, or it is correctly received by user 1, an event of probability $1 - \varepsilon_1$. Hence this transition probability is, $1 - \varepsilon_1 + \varepsilon_{12}$. At this transition a reward of one user 1 packet is assigned if the packet is correctly received by user 1, hence the average rewards for this transition is $\overline{\rho}_{AA}(1) = \frac{(1-\varepsilon_1)}{1-\varepsilon_1+\varepsilon_{12}}$, $\overline{\rho}_{AA}(2) = 0$. A transition $A \to B$ occurs when a transmitted user 1 packet is erased at user 1 and received correctly at user 2, an event of probability $\varepsilon_1 - \varepsilon_{12}$. In this case, $\overline{\rho}_{AB}(1) = \overline{\rho}_{AB}(2) = 0$ since the packet has not been received correctly by the intended destination. The rest of the transitions and rewards are similarly obtained.

Let π_s, $s \in \mathcal{S} = \{A, B, C, D\}$ be the steady state probability of the Markov Chain described above. The long term average number of successfully transmitted user i packets is then,

$$\overline{\rho}_i = \sum_{s \in S} \sum_{s' \in \mathcal{S}} \overline{\rho}_{ss'(i)} p_{ss'} \pi_s. \tag{8}$$

Hence, for large k_i the number of slots needed to transmit the k_i packets is about $k_i/\overline{\rho}_i$ and the number of time slots needed for Phase 1 to complete is $N_1 = \min\left\{\frac{k_1}{\overline{\rho}_1}, \frac{k_2}{\overline{\rho}_2}\right\}$.

Let now $k_1/\overline{\rho}_1 \leq k_2/\overline{\rho}_2$. Then, Phase 1 lasts $N_1 = k_1/\overline{\rho}_1$ slots. During this phase, the k_1 user 1 packets complete transmission and the number of user 2 packets transmitted is about $N_1\overline{\rho}_2$. Hence there are about

$$\widehat{K}_2 = k_2 - N_1\overline{\rho}_2 = k_2 - k_1\frac{\overline{\rho}_2}{\overline{\rho}_1}$$

user 2 packets left to be transmitted during phase 2. The number of slots needed for the latter packets to be transmitted is about

$$N_2 = \frac{\widehat{K}_2}{1-\varepsilon_2} = \frac{k_2}{1-\varepsilon_2} - \frac{k_1 \overline{\rho}_2}{(1-\varepsilon_2)\,\overline{\rho}_1}.$$

Hence the achievable rates by Algorithm III are

$$r_1 = \frac{k_1}{N_1+N_2} = \frac{\phi}{\frac{1}{1-\varepsilon_2} + \phi\left(\frac{1}{\overline{\rho}_1} - \frac{\overline{\rho}_2}{(1-\varepsilon_2)\overline{\rho}_1} - \frac{1}{1-\varepsilon_2}\right)}, \quad (9)$$

$\phi = \frac{k_1}{k_1+k_2}$, and similarly,

$$r_2 = \frac{1-\phi}{\frac{1}{1-\varepsilon_{21}} + \phi\left(\frac{1}{\overline{\rho}_1} - \frac{\overline{\rho}_2}{(1-\varepsilon_2)\overline{\rho}_1} - \frac{1}{1-\varepsilon_2}\right)}. \quad (10)$$

The probabilities π_s and then $\overline{\rho}_1,\ \overline{\rho}_2$, can be computed easily based on the transition diagram in Figure 2. It can then be seen that

$$\frac{\overline{\rho}_1}{\overline{\rho}_2} = \frac{(\varepsilon_1-1)(\varepsilon_2-\varepsilon_{12})}{(\varepsilon_2-1)(\varepsilon_1-\varepsilon_{12})},\ \frac{1}{\overline{\rho}_1} - \frac{\overline{\rho}_2}{(1-\varepsilon_2)\,\overline{\rho}_1} = \frac{1}{1-\varepsilon_{12}}. \quad (11)$$

It can be seen from (9), (10), (11) and a similar analysis for the case $k_1/\overline{\rho}_1 > k_2/\overline{\rho}_2$, that the rate region of Algorithm III is

$$\mathcal{R}^a_{III} = \{\boldsymbol{r} \geq \boldsymbol{0} : M^a(r_1, r_2) \leq L-3\},$$

which is essentially the same region as that of Algorithm I, with the factor $\frac{2^L-1}{2^L}$ replaced by 1.

Comparison between Algorithms I and III. Compared to Algorithm I, Algorithm III has the following advantages:

- Only XOR operations are performed, hence the computational complexity is reduced
- The order of packets correctly received by a user is the same as the order by which these packets were transmitted, hence eliminating the need for packet reordering at the receiver.
- Since only XOR operations are involved in the decoding, in general decoding of packets is performed earlier than in Algorithm I.

On the other hand advantages of Algorithm I are:

- During Phase 3 of Algorithm I only a single feedback is needed by each user, while Algorithm III needs the complete feedback. Even though Phase 2 of Algorithm III can be replaced by a phase where one a single feedback is needed, in general the feedback requirements of Algorithm I are smaller than those of Algorithm III
- Algorithm I can be extended in a simple manner to include multicast traffic (Algorithm II). While timesharing between separate multicast and user 1 - user 2 sessions can also implemented with Algorithm III, it is not clear whether intersession coding can be used with the latter algorithm in order to obtain the rate region $\widetilde{\mathcal{R}}^f$.
- Algorithm II seems more promising for generalization to more than two users.

VII Finite Arrivals

In this section we assume that there are random arrivals to the system. For the sake of definiteness we concentrate on Algorithm III, however, the same approach can be used for the other two algorithms.

Let $A_i(T)$, $i = 1, 2$, be the number of packet arrivals during slot T, with destination user i. We assume that $\{A_1(T), A_2(T)\}_{T=1}^{\infty}$ are i.i.d. with arrival rates $\lambda_i = \mathbb{E}[A_i(T)]$, $i = 1, 2$.

The transmitter has two buffers of infinite size, where the exogenously generated packets for user 1 and 2 are queued. The following algorithm is a natural application to the system with arrivals, of the algorithm developed in Section VI.

Algorithm IV

Let $K_i(T)$, $i = 1, 2$ be the number of queued packets for the two sessions at time T. The algorithm operates in epochs. Epoch 1 starts at time $\widehat{T}_1 = 0$. If $K_1(0) = K_2(0) = 0$, the epoch ends at time $\widehat{T}_2 = L$ and epoch 2 starts at the same time. Else, the following subepochs are employed.

- **Sub-epoch 1.** The numbers $k_i = K_i\left(\widehat{T}_1\right)$, $i = 1, 2$ are (re)transmitted to the users until they are correctly received by both.
- **Sub-epoch 2.** Algorithm III with packets k_i, $i = 1, 2$, is used for the transmission of these packets.

In general, after the end of epoch j at time $\widehat{T}_{j+1}$, epoch $j+1$ starts at the same time, employing the same procedure as in epoch 1, with packets $k_i = K_i\left(\widehat{T}_{j+1}\right)$, $i = 1, 2$.

A Lyapunov Function Drift analysis can be used to show that Algorithm IV stabilizes the system whenever the arrival rates (λ_1, λ_2) translated in bits per slot are in the interior of region $\mathcal{R}^a_{III}$.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2006.

[2] A. E. Gamal, "The feedback capacity of degraded broadcast channels," *IEEE Transactions on Information Theory*, vol. IT-25, no. 2, pp. 379–381, March 1978.

[3] A. F. Dana and B. Hassibi, "The capacity region of multiple input erasure broadcast channels," in *Proceedings of ISIT*, 4-9 September 2005, pp. 2315–2319.

[4] A. F. Dana, R. Gowaika, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 789–804, March 2006.

[5] D. S. Lun, M. Medard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, March 2008.

[6] C. Fragouli, D. S. Lun, M. Medard, and P. Pakzad, "On feedback for network coding," in *Proceedings of 2007 Conference on Information Sciences and Systems (CISS)*, March 2007.

[7] M. Durvy, C. Fragouli, and P. Thiran, "Towards reliable broadcasting using ACKs," in *Proceedings of 2007 IEEE Internationl Symposium on Information Theory (ISIT)*, 2007.

[8] J. K. Sundarajan, D. Shah, and M. Medard, "ARQ for network coding," in *Proceedings of 2008 IEEE International Symposium on Information Theory (ISIT)*, 2008.

[9] ——, "Online network coding for optimal throughput and delay - the three receiver case," in *Proceedings of International Symposium on Information Theory and Its Applications (ISITA 2008)*, December 2008.

[10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 243–254, October 2006.

[11] E. Rozner, A. Padmanabha, Y. Mehta, L. Qiu, and M. Jafry, "ER: Efficinet retransmission scheme for wireless LANs," in *Proceedings of the 2007 ACM CoNEXT*, New York, 2007.

[12] Y. Wu, P. A. Chou, and S. Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proceedings of CISS 2005.*