Feedback-based Coding Algorithms for Broadcast Erasure Channels with Degraded Message Sets

Marios Gatzianas, Shirin Saeedi Bidokhti and Christina Fragouli

School of Computer and Communication Sciences, EPFL, Switzerland.

***Abstract*—We consider single-hop broadcast packet erasure channels (BPEC) with degraded message sets and instantaneous feedback regularly available from all receivers, and demonstrate that the main principles of the virtual-queue-based algorithms in [1], which were proposed for multiple unicast sessions, can still be applied to this setting and lead to capacity-achieving algorithms. Specifically, we propose a generic class of algorithms and intuitively describe its rationale and properties that result in its efficiency. We then apply this class of algorithms to three examples of BPEC channels (with different numbers of users and 2 or 3 degraded message sets) and show that the achievable throughput region matches a known capacity outer bound, assuming feedback availability through a separate public channel. If the feedback channel is not public, all users can still decode their messages, albeit at some overhead which results in an achievable throughput that differs from the outer bound by $O(N/L)$, where L is the packet length. These algorithms do not require any prior knowledge of channel statistics for their operation.**

I. Introduction

The capacity region of an N-user BPEC channel has been determined for the cases of a single multicast session and, recently in [2], [1], for N unicast sessions. Both scenarios can be considered as extremes of a more general N-user setting, in which there exists, for each $\mathcal{S} \subseteq \{1, \dots, N\}$, a message $W_{\mathcal{S}}$ (equivalently, a set of packets $\mathcal{K}_{\mathcal{S}}$) that is intended for all users in $\mathcal{S}$.

Since the determination of the capacity region under the most general setting is still an open problem, this paper studies some special cases, in the hope that the results will provide further insight into the general case, as well as indicate the necessary properties of high-throughput algorithms. Specifically, motivated by the fact that feedback can strictly increase the capacity region of a BPEC channel (as has been convincingly demonstrated in [3] for 2 unicast sessions), this paper considers an N-user BPEC with feedback and a two-degraded message set, as well as the special case of a 3-user BPEC with 3 degraded messages, and modifies the algorithms in [1] to propose capacity-achieving algorithms in this setting.

The algorithms in [1] are recast into a systematic queue-based approach for performing inter-session network coding, which can also be tailored to problems other than the ones studied here (for example, a modification of the algorithm presented for the 3-user BPEC with a 3-degraded message set can be used for the 3-user BPEC with 2 messages, where message W_1 is intended for users 1,2 while W_2 is intended for user 3). To this end, and in order to illustrate all aspects of this class of algorithms, we present three concrete examples of application.

The problem of degraded multicasting (considered here for the case $N = 3$) has received a lot of attention, since this setting naturally models situations such as multiple-layer video transmission, where different users request different quality versions of essentially a single entity (e.g. a video stream). A recent work is [4], where it was shown that an extension of the result by Körner and Marton from two to three users (with a two-message degraded set) is not optimal. As a special case, [5] studied a 3-user system with 2 degraded messages over a combination network whose links are subject to iid erasures and derived its capacity region. However, in contrast to this paper, these works did not consider the use of feedback.

The paper is structured as follows. Section II contains the description of the system model and presents the three examples that will be used to illustrate the proposed class of algorithms, along with a summary of the results. Section III presents the class of algorithms and explains, in intuitive terms, the main ideas behind it, including the important feedback-based actions taken by the transmitter. After a brief summary, in Section IV, of the approach used to derive capacity outer bounds for the BPEC channels, Section V presents the exact algorithmic procedure for each of the three selected examples as well as the derivation of the achievable throughput region (inner bound). The latter is seen to match the outer bound in all 3 cases. Section VI concludes the paper.

II. Problem Formulation

A. System model

We consider a time-slotted communication system consisting of a single source/transmitter and N users/receivers with a degraded message set requirement. Denote the set of users with $\mathcal{N} = \{1, \dots, N\}$. The source has N messages $W_1, \dots, W_N$ of rates $R_1, \dots, R_N$, respectively, where user k wants to receive all messages up to and including W_k. In each slot, the input to the channel is a packet of length L bits (hereafter referred to as "input symbol") and the channel is modelled as memoryless broadcast erasure so that each broadcast packet/symbol is either received unaltered at a user or is "erased" and is not received by the user at all. At the end of each slot, all users

This work was supported by the ERC Starting Grant Project NOWIRE ERC-2009-StG-240317.

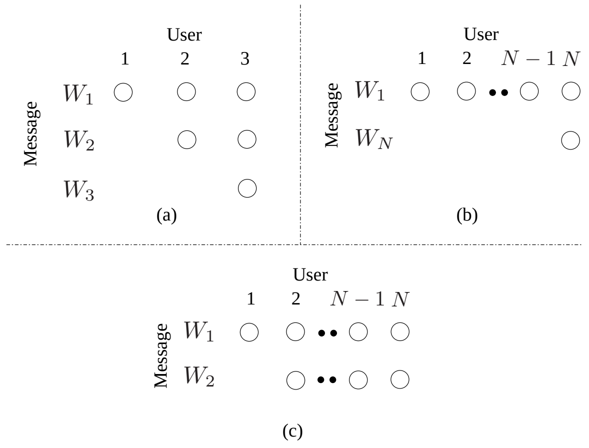

Fig. 1. The message sets of the three channels under investigation.

send a simple ACK/NACK reply (through a separate error-free and zero-delay channel) to inform the transmitter whether the packet was received or not.

We denote with $\epsilon_{\{i\}}$ the probability that a packet is erased by user i and with $\epsilon_{\mathcal{S}}$ the probability that a packet is erased by *all* users $k \in \mathcal{S}$ (the packet may also be erased by some users in $\mathcal{N}-\mathcal{S}$; such an outcome also belongs to the event measured by $\epsilon_{\mathcal{S}}$). Hence, we explicitly allow for spatial correlation between erasures at different users, although the memoryless property implies that erasures are temporally iid. We use the standard information-theoretic definitions [6] of code with feedback, achievable rate and capacity region. Furthermore, instead of working with messages $W_1, \dots, W_N$, we assume that there exist sets $\mathcal{K}_1, \dots, \mathcal{K}_N$ of packets intended for the same users. We denote $K_j \triangleq |\mathcal{K}_j|$, for $j = 1, \dots, N$, and assume K_j to be sufficiently large to invoke the strong law of large numbers.

B. Main Results

We give a full capacity characterization for this problem when $N = 3$ and partial capacity characterization when there are N users and only two message sets present. More precisely, we consider the message sets shown in Fig. 1 and prove the following results:

Theorem 1: The capacity region of the 3-user 3-degraded message set (case (a) in Fig. 1) is given by

$$\mathcal{C}_{(a)} = \Bigg\{ \boldsymbol{R} \geq \boldsymbol{0} : \frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2+R_3}{1-\epsilon_{\{1,3\}}} \leq 1,\ \frac{R_1+R_2}{1-\epsilon_{\{2\}}} + \frac{R_3}{1-\epsilon_{\{2,3\}}} \leq 1,\ R_1+R_2+R_3 \leq 1-\epsilon_{\{3\}}, \quad (1)$$
$$\frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2}{1-\epsilon_{\{1,2\}}} + \frac{R_3}{1-\epsilon_{\{1,2,3\}}} \leq 1 \Bigg\},$$

for arbitrary erasure spatial dependence.

Theorem 2: The capacity regions for the BPEC channels with message sets shown in cases (b), (c) of Fig. 1 are given, respectively, by

$$\mathcal{C}_{(b)} = \Bigg\{ \boldsymbol{R} \geq \boldsymbol{0} : \max_{i=1,\dots,N-1} \left(\frac{R_N}{1-\epsilon_{\{i,N\}}} + \frac{R_1}{1-\epsilon_{\{i\}}} \right) \leq 1,\ R_1 + R_N \leq 1-\epsilon_{\{N\}} \Bigg\}, \quad (2)$$

and

$$\mathcal{C}_{(c)} = \Bigg\{ \boldsymbol{R} \geq \boldsymbol{0} : \max_{i=2,\dots,N} \left(\frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2}{1-\epsilon_{\{1,i\}}} \right) \leq 1,\ R_1 + R_2 \leq \min_{i=2,\dots,N} (1-\epsilon_{\{i\}}),\ R_1 \leq 1-\epsilon_{\{1\}} \Bigg\}, \quad (3)$$

for arbitrary spatial erasure dependence in both cases.

III. A CLASS OF ENCODING ALGORITHMS

A. Algorithmic description

For the reader's convenience, we first succinctly describe the proposed class of encoding schemes, followed by a discussion on the intuition behind it and the reason for its throughput efficiency (we hereafter use the term "efficiency" to exclusively refer to high throughput achievability). We adopt the random linear network coding approach, in which the transmitted packets are viewed as elements of a finite field $\mathbb{F}_q$ (of size $q = 2^L$) and the transmitter keeps sending suitable linear combinations of information packets until all users receive a sufficient number of linearly independent combinations with respect to their intended packets. This can be achieved with probability arbitrarily close to 1 for sufficiently large q.

We assume that the random coefficient generator used by the transmitter is also available to all receivers. For a public feedback channel, this allows the receivers to use their local generators to create the values of the coefficients without the need of having the transmitter append them as packet preambles. For non-public feedback (where each user only knows its own ACK/NACK), the issue of conveying ACK/NACK to all users (so that they can again create the coefficients by themselves) can be solved through an overhead scheme in the spirit of [1]; the induced overhead is $O(N/L)$.

The algorithm is next described in general terms so that it can also be applied to BPEC channels with more general degraded message sets than those described in Fig. 1. We analyze this algorithm and characterize the rates it achieves for the BPEC channels of Fig. 1 in Section V.

1) Virtual queue structure and indices: the source maintains a group of virtual queues $Q_{\mathcal{S}}$, indexed by all non-empty subsets $\mathcal{S} \subseteq \mathcal{N}$, as well as non-negative integers $T^i_{\mathcal{S}}$, for all $\mathcal{S} \subseteq \mathcal{N}$ and $i \in \mathcal{S}$. The above queues and indices are dynamically updated during the algorithm's execution, as will be subsequently described.

2) Initialization: for the case where user i requests the packets of all sets $\mathcal{K}_1, \ldots, \mathcal{K}_i$, the packets of set $\mathcal{K}_i$ are placed in queue $Q_{\mathcal{N}-\{1,\ldots,i-1\}}$, for $i = 1, \ldots, N$.[1] The T indices are initialized as $T^i_{\mathcal{S}} = \|Q_{\mathcal{S}}\|$, where $\|\cdot\|$ denotes the number of packets stored in the queue.

3) Encoding scheme: the source sequentially processes each non-empty queue $Q_{\mathcal{S}}$, for all $\mathcal{S} \subseteq \mathcal{N}$, in order of increasing $|\mathcal{S}|$; queues with equal $|\mathcal{S}|$ are processed in order of increasing number of non-zero $T^i_{\mathcal{S}}$ indices (an arbitrary tie-breaker is used for queues with equal $|\mathcal{S}|$ and number of non-zero indices). The reason for selecting this order of processing will become apparent soon and further demonstrated in Section V. During the processing of $Q_{\mathcal{S}}$, the transmitter sends a linear combination of *all* packets stored in queues $Q_{\mathcal{S}}$ and $Q_{\mathcal{T}}$, for all $\mathcal{T} \supset \mathcal{S}$. We hereafter denote this joint group of queues as $Q_{\mathcal{T}\supseteq\mathcal{S}}$, so that the transmitted packet s has the form $s = \sum_{p \in Q_{\mathcal{T}\supseteq\mathcal{S}}} a_s(p)p$, with $a_s(p)$ the randomly chosen coefficients. Note that, although multiple queues are involved, we formally consider this phase as being applied to the "processing" of $Q_{\mathcal{S}}$.

4) Feedback-based queue management: while "processing" queue $Q_{\mathcal{S}}$, the source takes certain actions after each transmitted packet based on its received feedback. Denote the transmitted packet with s and let $\mathcal{G}$ be the set of users that successfully received s. The source now performs the following actions, collectively referred to as ACTFB:

Step 1: if $\mathcal{G} = \varnothing$ or it holds $T^i_{\mathcal{T}} = 0$, for all $\mathcal{T} \supseteq \mathcal{S}$ and $i \in \mathcal{G}$, then s is retransmitted (using the same coefficients). Otherwise,

Step 2: for each user $i \in \mathcal{S} \cap \mathcal{G}$, find the smallest cardinality set $\mathcal{S}^* \supseteq \mathcal{S}$ such that $T^i_{\mathcal{S}^*} > 0$ (with an arbitrary tie-breaker if more than one such sets exist) and decrease $T^i_{\mathcal{S}^*}$ by 1.

Step 3: if $\mathcal{G} \cap (\mathcal{N} - \mathcal{S}) \neq \varnothing$ and s was erased by at least one user in $\mathcal{S}$, then s is added to queue $Q_{\mathcal{S}\cup\mathcal{G}}$. Additionally, for each $i \in \mathcal{S} - \mathcal{G}$ with $T^i_{\mathcal{S}} > 0$, indices $T^i_{\mathcal{S}}$, $T^i_{\mathcal{S}\cup\mathcal{G}}$ are decreased/increased by 1, respectively.

The second and third items in the above list do not refer to mutually exclusive cases, and both of them may indeed be performed in a single slot.

5) Condition for stopping the processing of $Q_{\mathcal{S}}$: the source keeps processing each queue $Q_{\mathcal{S}}$ (i.e. it performs items 3,4 of this list) until it holds $T^i_{\mathcal{S}} = 0$ for all $i \in \mathcal{S}$. The algorithm terminates when all queues $Q_{\mathcal{S}}$, with $\mathcal{S} \subseteq \mathcal{N}$, have been processed.

B. The intuition behind the algorithm

The interpretation of the algorithm is greatly facilitated using the concept of token from [1].

Token: We denote with $\mathcal{D}_i$, for $i \in \mathcal{N}$, the set of packets intended for user i so that any transmitted packet s is, potentially, a linear combination of all packets in $\cup_{i\in\mathcal{N}}\mathcal{D}_i$.

Definition 1: Let $s = \sum_{p \in Q_{\mathcal{T}\supseteq\mathcal{S}}} a_s(p)p$ be a transmitted packet. Then, s is called a *token* for user i iff it can be written in the form

$$s = \sum_{u \in \mathcal{D}_i} b^{(i)}_s(u)u + c^{(i)}_s, \tag{4}$$

where $b^{(i)}_s(u), c^{(i)}_s$ are both known to user i.

The vector $\boldsymbol{b}^{(i)}_s$ can become known to user i due to the fact that the random coefficient generator of $a_s(p)$ is also available to i, along with public feedback, which allows user i to recreate the values of $a_s(p)$, and therefore $\boldsymbol{b}^{(i)}_s$. The value of $c^{(i)}_s$ can become known to user i due to packets received by i prior to packet s.

We now turn to the intuition behind the algorithm. A packet s that is transmitted from the source at time slot t can be "useful" for user i, if received by i, in two different ways:

(1) At the time of transmission, packet s is already a token for user i, so that, upon reception of s, user i creates an equation, which can be made linearly independent of previously created equations at user i (through a proper selection of $a_s(p)$). In this sense, packet s offers an immediate "benefit" to user i.

(2) At the time of transmission, packet s is not a token for user i, but since it is received at user i, it can offer side information to user i. In this sense, packet s can be combined in the future with a side information packet of another user and create a packet that is useful to both of them, thus offering a "delayed benefit". Note that from the arrival time of packet s at user i onwards, this packet becomes a token for user i as well.

We give an example for these two types of useful information. A packet $p_3 \in \mathcal{K}_3$ offers immediate benefit only to user 3, if received successfully at user i. Nonetheless, p_3 can also offer delayed benefit as follows: if p_3 is only received at user 2 (so that we denote p with p^2_3, where the upper index indicates the user who received the packet), it can be potentially combined with a packet of the form $p_2 \in \mathcal{K}_2$ later on, so that the resulting linear combination $\langle p_2, p^2_3\rangle$ allows both users 2 and 3 to create an equation upon successful reception. Clearly, this efficient mixing of side information requires the transmitter to know exactly which user knows which packets; this knowledge is acquired through feedback.

The following statement indicates that the property of "tokeness" is maintained through linear combinations [1].

Proposition 1: Consider a set of packets $\mathcal{P}$ such that each $p \in \mathcal{P}$ is a token for all users in a set $\mathcal{U}$. Then, any packet of the form $s = \sum_{p\in\mathcal{P}} a_s(p)p$ is also a token for all users $i \in \mathcal{U}$.

Based on the above, the most "useful" type of transmitted packet is a linear combination of packets each of which is a token for *all* N users. Before any transmissions occur and when no side information has been received yet, each packet is a token only for its intended destinations, so that the most efficient packets, at the beginning of the algorithm, belong to $\mathcal{K}_1$. However, as transmissions are scheduled and side information is created, additional packets may become tokens for all N users. For example, for the case $N = 3$, a packet $p_2 \in \mathcal{K}_2$ that is received by 1 becomes, after its reception, a token for all 3 users. Hence, tokeness is a time-dependent

[1]equivalently, the packets in set $\mathcal{K}_i$ are intended for users $i, \ldots, N$. For the most general case where a set of packets $\mathcal{K}_{\mathcal{S}}$ is intended for all users in set $\mathcal{S}$, the packets of $\mathcal{K}_{\mathcal{S}}$ are placed in queue $Q_{\mathcal{S}}$.

notion with an absorbing property, i.e. once a packet becomes a token for a user, it remains a token for this user until the end of the algorithm.

The indices $T_{\mathcal{S}}^{i}$ are interpreted as the number of linearly independent combinations that user i still needs to receive during the processing of $Q_{\mathcal{S}}$ (item 3 in the list of Section III-A). When it holds $T_{\mathcal{S}}^{i}=0$, then user i has gathered all available information from $Q_{\mathcal{S}}$ and the queue is no longer useful for i, though it is still useful for other users $j \in \mathcal{S}$ with $T_{\mathcal{S}}^{j}>0$. Processing of $Q_{\mathcal{S}}$ therefore stops, as explained, only when *all* $T_{\mathcal{S}}^{i}$, $i \in \mathcal{S}$, become 0.

The efficiency of the algorithm lies in the fact that it tries to "exploit" each slot as much it can, in the sense that the transmitted packet potentially allows each user, upon successful reception of the packet, to either create an equation or gain side information. While processing $Q_{\mathcal{S}}$, packets are being transmitted that are tokens for all users in $\mathcal{S}$. When such a packet is received by a user $i \in \mathcal{S}$, the user forms an equation. When such a packet is, however, received at a user $i \notin \mathcal{S}$ (for which it is not a token at the time of transmission), the packet can still be useful, if properly handled. ACTFB tries to optimally handle this based on the feedback received and according to the following principle:

Maximal tokeness (MT): a transmitted packet s should be placed in queue $Q_{\mathcal{S}'}$ iff $\mathcal{S}'$ is the maximal set such that s is a token for *all* users in $\mathcal{S}'$ (e.g. if packet s is a token for users $1,2$ as well as $1,2,4$, and is not a token for any user $j \notin \{1,2,4\}$, then s is placed in queue $Q_{\{1,2,4\}}$).

Notice that the queue initialization also conforms to the MT rule. Since we have qualitatively defined the "efficiency" of a packet as the number of users for which it is a token, we conclude that enforcing the MT principle results in a packet becoming more efficient as it is stored in $Q_{\mathcal{S}}$ of increasing $|\mathcal{S}|$.

Interpretation of ACTFB: The three steps of ACTFB follow naturally from the MT principle and the interpretation of indices $T_{\mathcal{S}}^{i}$. Specifically, step 1 captures the fact that it is possible for a packet s to be received by no user, in which case s is retransmitted, since it still contains useful information. The second condition in step 1 of ACTFB is more intriguing but can be simply restated as follows: if a packet s, which is a linear combination of all packets in $Q_{\mathcal{T}\supseteq S}$ is received *only* by users which have already recovered all available equations in *all* queues $Q_{\mathcal{T}\supseteq S}$ (i.e. $T_{\mathcal{T}}^{i}=0$ for all $\mathcal{T}\supseteq\mathcal{S}$), then the packet is retransmitted, since it cannot offer any new information to the users that received it. Although this may lead to inefficiency, the analysis performed in Section V indicates that this does *not* happen for the message sets in Fig. 1 (i.e. the algorithm achieves a capacity outer bound). The question of whether this property holds for general N is an open problem.

Step 2 says that a user $i \in \mathcal{S}$ that receives s can construct an equation for its unknown packets (due to Proposition 1 and the fact that all packets in $Q_{\mathcal{S}}$ are, by construction, tokens for all users in $\mathcal{S}$). Hence, the corresponding counter $T_{\mathcal{S}}^{i}$ should be decreased by 1 to capture this fact.

Finally, step 3 corresponds to the case where packet s is not received by all receivers for which it is a token (i.e., all users in $\mathcal{S}$) but is received by some users that do not belong to $\mathcal{S}$ (i.e., packet s is not a token for them in the time of transmission). After reception of this packet, it also becomes a token for all such users (in set $\mathcal{G}-\mathcal{S}$) so that, before the next transmission occurs, packet s has become a token for all users in $\mathcal{S}\cup\mathcal{G}$, the latter set being maximal. Hence, according to the MT rule, s should be placed in $Q_{\mathcal{S}\cup\mathcal{G}}$ for the next transmission time slot.

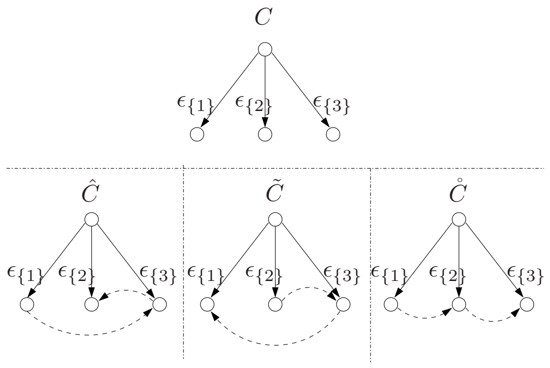

Fig. 2. Channel C shows a BPEC with $N=3$ users. Channels $\hat{C}$, $\tilde{C}$, $\mathring{C}$ are the auxiliary channels used to derive capacity outer bounds for the three-message set problem.

Furthermore, although there might be some user $j \in \mathcal{S}$ that did not receive packet s (i.e. $j \in \mathcal{S}-\mathcal{G}$), this information can be sent in the future through linear combinations transmitted from queue $Q_{\mathcal{S}\cup\mathcal{G}}$, into which packet s has been moved. In fact, since s is now a token for all users in $\mathcal{S}\cup\mathcal{G}$, it is more efficient for user $j \in \mathcal{S}-\mathcal{G}$ to receive a linear combination (containing s) from $Q_{\mathcal{S}\cup\mathcal{G}}$ rather than s itself from $Q_{\mathcal{S}}$, since the former combination can provide information to more users. This is accounted for by decreasing/increasing $T_{\mathcal{S}}^{i}$, $T_{\mathcal{S}\cup\mathcal{G}}^{i}$ by one.

Remark 1: Since the processing of $Q_{\mathcal{S}}$ may place some packets in $Q_{\mathcal{S}'}$, with $\mathcal{S}'\supset\mathcal{S}$ (as well as increase some $T_{\mathcal{S}'}^{i}$) and since all queues with non-zero T indices have to be processed eventually, selecting the order of processing according to increasing $|\mathcal{S}|$ means that each queue $Q_{\mathcal{S}}$ will be processed, according to item 3 of Section III-A, in only one stage. Selecting a different order such that, for two sets $\mathcal{S}_1,\mathcal{S}_2$ with $\mathcal{S}_1\supset\mathcal{S}_2$, queue $Q_{\mathcal{S}_1}$ is processed before $Q_{\mathcal{S}_2}$ allows for the possibility that $Q_{\mathcal{S}_1}$ may have to be processed twice, the second time being due to packets newly moved from $Q_{\mathcal{S}_2}$. To avoid this issue, which would make the analysis of the algorithm more difficult, we stick to processing the queues in order of increasing $|\mathcal{S}|$.

Before we apply this class of algorithms to the message sets of Fig. 1 and determine the achievable throughout regions, we briefly summarize, in the next Section, the procedure for deriving a capacity outer bound.

IV. Capacity outer bound

In this Section, we prove an outer bound only for the 3-user 3-degraded message set (case (a) in Fig. 1). The other cases are similarly handled and we refer the reader to [7] for the detailed derivation.

Consider any encoding scheme at the source that achieves rates R_1, R_2 and rate R_3 over the original erasure channel C. The cut-set bound yields a trivial outer capacity region of the form $\mathcal{C}_{cs} = \left\{\boldsymbol{R} \geq \boldsymbol{0} : \sum_{k=1}^{i} R_k \leq 1 - \epsilon_{\{i\}},\ i = 1, \ldots, 3\right\}$. The derivation of a tighter capacity outer bound than $\mathcal{C}_{cs}$ follows the often-used procedure [8], [3] of introducing additional auxiliary links of infinite capacity among the users. More precisely, we consider the auxiliary channels shown in Fig. 2, where dashed lines indicate infinite capacity links.

It is not difficult to see that any encoding scheme at the source that achieves rates R_1, R_2 and rate R_3 over the original erasure channel C, could be used over channel $\hat{C}$ to reliably communicate messages W_2 and W_3 to user 3 and message W_1 to user 1. Similarly, any such code could be used over channel $\tilde{C}$ to reliably communicate messages W_1 and W_2 to user 2 and message W_3 to user 3. Finally, the same code could be used over channel $\mathring{C}$ to communicate message W_i to user i for $i = 1, 2, 3$. Therefore, these 3 unicast problems give outer bounds on the capacity region of our interest; i.e., denoting with $\hat{\mathcal{C}}, \tilde{\mathcal{C}}, \mathring{\mathcal{C}}$ the capacity regions of the aforementioned unicast problems over $\hat{C}$, $\tilde{C}$, $\mathring{C}$ we get

$$\mathcal{C}_{(a)} \subseteq \mathcal{C}_{cs} \cap \hat{\mathcal{C}} \cap \tilde{\mathcal{C}} \cap \mathring{\mathcal{C}}. \tag{5}$$

Over channel $\hat{C}$, a symbol is actually erased at user 3 in channel $\hat{C}$ iff both users 1, 3 erase the symbol in channel C, while user 2 erases the symbol in channel $\hat{C}$ iff all three users erase it in channel C. This implies that $\hat{C}$ is also a BPEC with parameters $\hat{\epsilon}_{\{1\}} = \epsilon_{\{1\}}$, $\hat{\epsilon}_{\{2\}} = \epsilon_{\{1,2,3\}}$ and $\hat{\epsilon}_{\{3\}} = \epsilon_{\{1,3\}}$. Analogous conclusions can be drawn for channel $\tilde{C}$, $\mathring{C}$.

Furthermore, since channels $\hat{C}$, $\tilde{C}$, $\mathring{C}$ are physically degraded [9], we have the following well-known results:

- feedback does not increase the capacity of a physically degraded channel [10].
- the capacity region (without feedback) of the N-user physically degraded BPEC is achieved through timesharing among the users [11].

Combining the above to evaluate $\hat{\mathcal{C}}$, $\tilde{\mathcal{C}}$ and $\mathring{\mathcal{C}}$ in (5), the RHS of (5) matches (1) of Theorem 1 and provides an outer bound to the capacity region of the 3-message set broadcast problem. We next show that the region in (1) is also achievable, which completes the proof of Theorem 1.

V. Examples of application

A. The 3-message degraded set

The algorithm uses all queues $Q_{\mathcal{S}}$, with $\mathcal{S} \subseteq \{1, 2, 3\}$, except for $Q_{\{1,2\}}$, which remains empty for the entire duration of the algorithm. After placing the packets of sets $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$ into queues $Q_{\{1,2,3\}}$, $Q_{\{2,3\}}$, $Q_{\{3\}}$, respectively, and initializing the T indices according to item 2 of Section III-A, the source processes the queues in the following order: $Q_{\{3\}}$, $Q_{\{1,3\}}$, $Q_{\{2,3\}}$, $Q_{\{1,2,3\}}$. This order is dictated by the rule (see item 3 of Section III-A) of processing queues $Q_{\mathcal{S}}$ in order of increasing $|\mathcal{S}|$ and the fact that it holds $T^1_{\{1,3\}} = 0$ for the entire algorithm's execution. Hence, $Q_{\{1,3\}}$ has fewer non-zero T indices than $Q_{\{2,3\}}$ and is processed first. Recall that the processing of $Q_{\mathcal{S}}$ entails transmitting linear combinations from all queues $Q_{\mathcal{T} \supseteq \mathcal{S}}$ until all $T^i_{\mathcal{S}}$ become 0.

We denote with $T^*_{\mathcal{S}}$ the (average) number of slots required for processing queue $Q_{\mathcal{S}}$, and with $T^i_{\mathcal{S}}(t)$ the value of $T^i_{\mathcal{S}}$ at slot t. Hence, it follows that

$$T^*_3 = \frac{T^3_{\{3\}}(0)}{1 - \epsilon_{\{1,2,3\}}} = \frac{K_3}{1 - \epsilon_{\{1,2,3\}}}, \tag{6}$$

since any packet that is received by at least one user leads, due to step 3 of ACTFB, to a reduction of $T^3_{\{3\}}$ by 1.

The values of the indices at the end of processing $Q_{\{3\}}$ (denote this epoch as t_3) are

$$\begin{aligned} T^3_{\{2,3\}}(t_3) &= K_2 + T^*_{\{3\}}(\epsilon_{\{1,3\}} - \epsilon_{\{1,2,3\}}), \\ T^3_{\{1,3\}}(t_3) &= T^*_{\{3\}}(\epsilon_{\{2,3\}} - \epsilon_{\{1,2,3\}}), \quad T^1_{\{1,3\}}(t_3) = 0, \\ T^3_{\{1,2,3\}}(t_3) &= K_1 + T^*_{\{3\}}(\epsilon_{\{3\}} - \epsilon_{\{1,3\}} - \epsilon_{\{2,3\}} + \epsilon_{\{1,2,3\}}), \\ T^2_{\{2,3\}}(t_3) &= K_2, \quad T^1_{\{1,2,3\}}(t_3) = T^2_{\{1,2,3\}}(t_3) = K_1, \end{aligned} \tag{7}$$

and follow again from the logic of step 3 of ACTFB (the terms inside parentheses express the probability that a packet is only seen by a specific subset of the 3 users).

To make the equations more compact, we denote with $\Delta^+_{\mathcal{S}} T^i_{\mathcal{T}}$, $\Delta^-_{\mathcal{S}} T^i_{\mathcal{T}}$ the total increase/decrease, respectively, of index $T^i_{\mathcal{T}}$ when the algorithm processes $Q_{\mathcal{S}}$, with $\mathcal{S} \subset \mathcal{T}$. Using this notation, we can write, for example, $\Delta^+_{\{3\}} T^3_{\{2,3\}} = T^*_{\{3\}}(\epsilon_{\{1,3\}} - \epsilon_{\{1,2,3\}})$.

The source next processes $Q_{\{1,3\}}$ for a total of

$$T^*_{\{1,3\}} = \frac{T^3_{\{1,3\}}(t_3)}{1 - \epsilon_{\{2,3\}}}, \tag{8}$$

time slots (since $T^3_{\{1,3\}}$ is reduced if the transmitted packet is received by either 2 or 3), while the indices $T^3_{\{1,2,3\}}$, $T^1_{\{1,2,3\}}$ are modified, due to step 3 of ACTFB as follows

$$\begin{aligned} \Delta^+_{\{1,3\}} T^3_{\{1,2,3\}} &= T^*_{\{1,3\}}(\epsilon_{\{3\}} - \epsilon_{\{2,3\}}), \\ \Delta^-_{\{1,3\}} T^1_{\{1,2,3\}} &= T^*_{\{1,3\}}(1 - \epsilon_{\{1\}}). \end{aligned} \tag{9}$$

Notice that if $T^1_{\{1,2,3\}}$ becomes zero before the processing of $Q_{\{1,3\}}$ is complete, then user 1 has received enough linear combinations to decode its packets.

The source next processes $Q_{\{2,3\}}$ for a total of

$$T^*_{\{2,3\}} = \max\left(\frac{T^2_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,2\}}}, \frac{T^3_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,3\}}}\right), \tag{10}$$

time slots. The modification of the indices in $Q_{\{1,2,3\}}$ during this stage is

$$\begin{aligned} \Delta^+_{\{2,3\}} T^3_{\{1,2,3\}} &= \frac{T^3_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,3\}}}(\epsilon_{\{3\}} - \epsilon_{\{1,3\}}), \\ \Delta^+_{\{2,3\}} T^2_{\{1,2,3\}} &= \frac{T^2_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,2\}}}(\epsilon_{\{2\}} - \epsilon_{\{1,2\}}), \end{aligned} \tag{11}$$

$$\begin{aligned}\Delta^-_{\{2,3\}}T^3_{\{1,2,3\}} &= \left(T^*_{\{2,3\}} - \frac{T^3_{\{2,3\}}(t_3)}{1-\epsilon_{\{1,3\}}}\right)(1-\epsilon_{\{3\}}),\\ \Delta^-_{\{2,3\}}T^2_{\{1,2,3\}} &= \left(T^*_{\{2,3\}} - \frac{T^2_{\{2,3\}}(t_3)}{1-\epsilon_{\{1,2\}}}\right)(1-\epsilon_{\{2\}}).\end{aligned} \tag{12}$$

Hence, at the end of processing $Q_{\{2,3\}}$ (denote this epoch with t_{23}), the indices in $Q_{\{1,2,3\}}$ have the values $T^i_{\{1,2,3\}}(t_{23}) = [T^i_{\{1,2,3\}}(t_3) + \sum_{\mathcal{S}} \Delta^+_{\mathcal{S}} T^i_{\{1,2,3\}} - \sum_{\mathcal{S}} \Delta^-_{\mathcal{S}} T^i_{\{1,2,3\}}]^+$, where the summation is performed over $\mathcal{S} \in \{\{1,3\},\{2,3\}\}$ and $[x]^+ = \max(x,0)$. Therefore, the processing of $Q_{\{1,2,3\}}$ by itself requires $T^*_{\{1,2,3\}} = \max_{i=1,\dots,3} T^i_{\{1,2,3\}}(t_{23})/(1-\epsilon_{\{i\}})$ slots. Denoting the sum of the slots for all phases as T^*, the algorithm achieves a rate of $R_j = K_j/T^*$ for $j = 1,\dots,3$. Simple algebra reveals that the throughput region exactly matches (1). Notice that no assumption on spatially independent erasures was made, so that the result holds for arbitrary erasures.

B. 2-message degraded set: case (b) in Fig. 1

The algorithm only operates on queues $Q_{\mathcal{S}}$ such that $N \in \mathcal{S}$ and initially places the packets of sets $\mathcal{K}_1$, $\mathcal{K}_N$ into $Q_{\mathcal{N}}$, $Q_{\{N\}}$, respectively. Additionally, since, for $i \neq N$ and $\mathcal{S} \subset \mathcal{N}$, all indices $T^i_{\mathcal{S}}$ are always 0 (users $1,\dots,N-1$ only require packets from $Q_{\mathcal{N}}$), there is no point in combining each $Q_{\mathcal{S}}$ with all queues $Q_{\mathcal{T}}$, with $\mathcal{T} \supset \mathcal{S}$. Hence, it suffices to combine $Q_{\mathcal{S}}$ directly with $Q_{\mathcal{N}}$.

For $\mathcal{S} \subset \mathcal{N}$, we denote with $T^*_{\mathcal{S}}$, $T^N_{\mathcal{S}}(t_s)$, respectively, the number of slots required for processing $Q_{\mathcal{S}}$ and the value of index $T^N_{\mathcal{S}}$ at the beginning of the processing, so that it holds $T^*_{\mathcal{S}} = T^N_{\mathcal{S}}(t_s)/(1-\epsilon_{\mathcal{N}-(\mathcal{S}-\{N\})})$. A recursive formula can be written [7] for computing $T^N_{\mathcal{S}}(t_s)$ for all $\mathcal{S} \subseteq \mathcal{N}$, from which the total number of slots required by the algorithm can be computed (and hence, the achieved rate). Again, the proposed algorithm achieves capacity for arbitrary erasure spatial dependence.

C. 2-message degraded set: case (c) in Fig. 1

This is the simplest of the 3 cases to be examined, in the sense that only two queues are required, namely $Q_{\mathcal{N}}$ and $Q_{\mathcal{N}-\{1\}}$. The packets of sets $\mathcal{K}_1$, $\mathcal{K}_2$ are placed into $Q_{\mathcal{N}}$, $Q_{\mathcal{N}-\{1\}}$, respectively, along with the suitable initialization of $T^i_{\mathcal{N}}$ and $T^i_{\mathcal{N}-\{1\}}$. The source first combines the two queues until all $T^i_{\mathcal{N}-\{1\}}$ become 0 and then processes $Q_{\mathcal{N}}$ by itself as usual.

Thinking in similar lines as for the analysis of the 3-message degraded set, the number of slots $T^*_{\mathcal{N}-\{1\}}$ required to process $Q_{\mathcal{N}-\{1\}}$ is $T^*_{\mathcal{N}-\{1\}} = \max_{i\in\mathcal{N}-\{1\}} \frac{K_2}{1-\epsilon_{\{1,i\}}}$. The cumulative increase and decrease of $T^i_{\mathcal{N}}$ can be computed similar to (11), (12) as

$$\begin{aligned}\Delta^+T^i_{\mathcal{N}} &= \frac{K_2}{1-\epsilon_{\{1,i\}}}(\epsilon_{\{i\}} - \epsilon_{\{1,i\}}),\\ \Delta^-T^i_{\mathcal{N}} &= \left(T^*_{\mathcal{N}-\{1\}} - \frac{K_2}{1-\epsilon_{\{1,i\}}}\right)(1-\epsilon_{\{i\}}).\end{aligned} \tag{13}$$

The rest of the analysis follows the lines of Section V-A and reveals, after some simple algebra, that the proposed algorithm achieves capacity. Notice that, again, this result holds for arbitrary erasure spatial dependence.

VI. Conclusions

This paper revisited the virtual-queue based coding algorithm proposed in [1] for BPEC channels with multiple unicast sessions and demonstrated that its main concepts of token handling and keeping track of the number of linearly independent equations required by each user are still applicable for the setting of degraded message sets, essentially creating a whole “class” of algorithms for BPEC channels. Three simple examples were chosen to illustrate the main ideas of this class, and it became apparent that the complexity of this algorithmic class (in terms of the number of virtual queues needed) is mainly determined by the number of sessions and the relation between the message sets rather than the number of users.

References

[1] M. Gatzianas, L. Georgiadis, and L. Tassiulas, “Muliple user broadcast erasure channel with feedback — capacity and algorithms,” submitted to IEEE Trans. Inform. Theory. [Online]. Available: http://arxiv.org/abs/1009.1254

[2] C.-C. Wang, “Capacity of 1–to-K broadcast packet erasure channels with channel output feedback,” in *Proc. 48th Annual Allerton Conference*, October 2010, also appeared in *IEEE Trans. Inform. Theory*, vol. 58, no. 2, Feb. 2012.

[3] L. Georgiadis and L. Tassiulas, “Broadcast erasure channel with feedback — capacity and algorithms,” in *Proc. 5th Workshop on Network Coding Theory and Applications*, June 2009, pp. 54–61.

[4] C. Nair and A. E. Gamal, “The capacity region of a class of 3-receiver broadcast channels with degraded message sets,” in *Proc. ISIT*, June 2008.

[5] S. Gheorghiu, S. Saeedi, C. Fragouli, and A. Lopez, “Degraded multicasting with network coding over the combination network,” in *Proc. NetCod 2011*, July 2011.

[6] T. Cover and J. Thomas, *Elements of information theory*, 2nd ed. John Wiley, 2006.

[7] S. Saeedi, M. Gatzianas, and C. Fragouli, “A class of feedback-based coding algorithms for broadcast erasure channels with degraded message sets,” EPFL, Tech. Rep., 2012. [Online]. Available: http://infoscience.epfl.ch/record/175823/files/techreport.pdf

[8] L. Ozarow and S. Leung-Yan-Cheong, “An achievable region and outer bound for the Gaussian broadcast channel with feedback,” *IEEE Trans. Inform. Theory*, vol. 30, no. 4, pp. 667–671, July 1984.

[9] P. Bergmans, “Random coding theorem for broadcast channels with degraded components,” *IEEE Trans. Inform. Theory*, vol. 19, no. 2, pp. 197–207, March 1973.

[10] A. E. Gamal, “The feedback capacity of degraded broadcast channels,” *IEEE Trans. Inform. Theory*, vol. 24, no. 3, pp. 379–381, May 1978.

[11] A. Dana and B. Hassibi, “The capacity region of multiple input erasure broadcast channels,” in *Proc. International Symposium on Information Theory (ISIT)*, September 2005, pp. 2315–2319.