The Two-User Broadcast Packet Erasure Channel with Feedback and Memory

Michael Heindlmaier, Clemens Blöchl
Institute for Communications Engineering, Technische Universität München, Munich, Germany
Email: michael.heindlmaier@tum.de, clemens.bloechl@web.de

***Abstract*—The two-user broadcast erasure channel with feedback and memory is analyzed. It is shown that memory increases the capacity region for this scenario. Several heuristic algorithms are proposed and analyzed. Although these schemes do not achieve capacity, significant gains can be observed compared to the memoryless case.**

I. Introduction

The memoryless broadcast erasure channel (BEC) with feedback (FB) is one of the examples in multiuser communication systems where the capacity region is known and feedback increases the capacity region. The result was found recently in [1] for the two-user case and was extended to the general K-user case in [2], [3]. For the two-user case with independent erasure probabilities ϵ_1 and ϵ_2, the capacity region consists of all rate pairs $(R_1, R_2) \in \mathcal{C}_f = \mathcal{C}_1 \cap \mathcal{C}_2$, where

$$\mathcal{C}_1 = \left\{(R_1, R_2) \geq 0 : \frac{R_1}{1-\epsilon_1} + \frac{R_2}{1-\epsilon_1\epsilon_2} \leq 1\right\},$$
$$\mathcal{C}_2 = \left\{(R_1, R_2) \geq 0 : \frac{R_1}{1-\epsilon_1\epsilon_2} + \frac{R_2}{1-\epsilon_2} \leq 1\right\},$$

and rates are measured in symbols per transmission. This region is strictly larger than the capacity region for the memoryless BEC without FB, denoted $\mathcal{C}_{\text{nofb}}$, consisting of all rate pairs

$$(R_1, R_2) \geq 0 : \frac{R_1}{1-\epsilon_1} + \frac{R_2}{1-\epsilon_2} \leq 1. \tag{1}$$

The latter region is one of the few examples for broadcast channels where time sharing between different users is optimal.

Motivated by possible applications, we try to extend this work for channels with memory. In many practical scenarios erasures occur in bursts. If no feedback is present, interleaving can decorrelate the erasures and erasure correcting codes designed for memoryless channels can be used. With feedback however, more sophisticated methods are possible or simpler coding schemes suffice. We analyze some of the capacity-achieving algorithms proposed in [1] when erasures do not occur independently in time and introduce two new schemes. While none of the proposed algorithm can be shown to be capacity-achieving for the case with memory, significant gains can be exploited if memory is present.

The work for the broadcast erasure channel with feedback is closely related to the index coding problem [4]. For index coding, one usually seeks to minimize the number of transmissions that are necessary to complete a multicast transmission [5], [6]. Without erasures, the index coding reduces to a hard combinatorial problem [4], [7], [8].

A related setup has been investigated in [9], where the authors study the multiple-input multiple-output broadcast packet erasure channel with feedback for two users, without memory however. In [10], the scheme in [1] was extended to Rayleigh fading channels.

For the general case of discrete channels, the two-user finite-state broadcast channel with feedback and unidirectional user cooperation has been investigated in [11]. To the best of our knowledge for the case without user cooperation, the general capacity region remains unknown.

II. System Model

We keep the notation similar to [1]. In particular, a sequence $X_1, X_2, \ldots, X_n$ is denoted by X^n. Two users U_1 and U_2 wish to receive two independent messages $W_1 \in \mathcal{W}_1$, $W_2 \in \mathcal{W}_2$ in n uses of the channel from one transmitter.

Let $X_t \in \mathcal{X}$ denote channel input in time slot t, $t = 1, \ldots, n$. Each channel input corresponds to a packet of L bits, so $\mathcal{X} = \mathbb{F}_q$ with $q = 2^L$. All rates, entropies and mutual informations are with respect to logarithms to the base q, so $\mathcal{W}_1 = \{1, 2, \ldots, q^{nR_1}\}$, $\mathcal{W}_2 = \{1, 2, \ldots, q^{nR_2}\}$.

Packet X_t is broadcast to both users, where it is either completely erased or perfectly received. Let E denote the erasure symbol. The channel output at user i in time slot t is denoted by $Y_{i,t} \in \mathcal{X} \cup \{E\}$. Define $Z_{i,t}$ as a binary random variable indicating if an erasure occurred at user i in slot t, i.e. $Z_{i,t} = \mathbb{1}\{Y_{i,t} = E\}$. Equivalently, one can write $Y_{i,t} = Z_{i,t}E + (1 - Z_{i,t})X_t$. $Y_{i,t}$ is thus a function of X_t and $Z_{i,t}$, but $Y_{i,t}$ also determines $Z_{i,t}$ without ambiguity.

Different to [1], the sequence $\{Z_{i,t}\}_{t=1}^{\infty}$ is correlated in time. For simplicity we assume statistical independence of the pair $(Z_{1,t}, Z_{2,t})$. Hence, erasures occur independently over users, but not over time, with average erasure probability at user i

$$\epsilon_i = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \Pr[Z_{i,t} = 1]. \tag{2}$$

The correlation of the erasures is according to a Gilbert-Elliot model [12], [13]. The random channel state at user i in slot t is denoted by $S_{i,t} \in \mathcal{S}_i$, $\mathcal{S}_i = \{\mathrm{G}, \mathrm{B}\}$, referring to a good and bad state. The erasure probability can differ in each of the two states. Although the results can be adapted, we focus on the

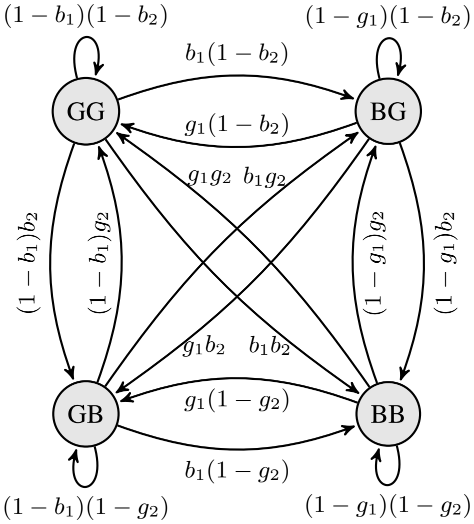

Fig. 1. Markov Chain of channel state space with transition probabilities.

case where $\Pr[Z_{i,t}=1|S_{i,t}=\mathrm{G}]=0$ and $\Pr[Z_{i,t}=1|S_{i,t}=\mathrm{B}]=1$.

The overall channel state in slot t is the concatenation of $S_{1,t}$ and $S_{2,t}$, denoted by $S_t \in \mathcal{S}$, $\mathcal{S}=\{\mathrm{GG},\mathrm{GB},\mathrm{BG},\mathrm{BB}\}$. Transitions between the channel states are parametrized by

$$\begin{aligned}\Pr[S_{i,t+1}=\mathrm{G}|S_{i,t}=\mathrm{B}]&=g_i\\ \Pr[S_{i,t+1}=\mathrm{B}|S_{i,t}=\mathrm{G}]&=b_i.\end{aligned} \tag{3}$$

The corresponding channel transitions for the four-state channel Markov chain is shown in Fig. 1. The parameters of the model are assumed to be known to the transmitter.

For the ergodic case the average erasure probability yields

$$\epsilon_i=\lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n}\Pr[S_{i,t}=\mathrm{B}]=\pi_{i,\mathrm{B}}=\frac{b_i}{g_i+b_i}, \tag{4}$$

where $\pi_{i,\mathrm{B}}$ is the steady-state probability of the bad state at user i. The initial channel state at user i, $S_{i,0}$ is distributed according to this steady-state distribution. This assumption represents the fact that the initial channel state is unknown. If $0<b_i<1$, $0<g_i<1$ for both $i=1,2$, the channel Markov chain is ergodic.

Define the channel memory μ_i for user i as in [14]:

$$\begin{aligned}\mu_i&=\Pr[S_{i,t+1}=s|S_{i,t}=s]-\Pr[S_{i,t+1}=s|S_{i,t}\neq s]\\ &=1-g_i-b_i \qquad \text{for } s\in\{\mathrm{G},\mathrm{B}\}.\end{aligned} \tag{5}$$

If $\mu_i>0$, the channel is called persistent, as the probability of remaining in a state is larger than the steady-state probability. If $\mu_i<0$, the channel is oscillatory, whereas for $\mu_i=0$ the channel is memoryless.

After slot t, each receiver provides feedback if an erasure occurred or not. This feedback is assumed without errors and delay. This enables the transmitter to choose its next channel input as a function of the previous channel state:

$$X_t=f_t(W_1,W_2,Z_1^{t-1},Z_2^{t-1}) \tag{6}$$

For simplicity we assume that this feedback is known to all nodes in the network. In most cases this assumption is not realistic, but it simplifies notation and algorithms. Usually, algorithms can be adapted to work also with feedback only present at the transmitter.

The decoder at user i maps its received sequence to an estimate $\hat{W}_i=h_i(Y_i^n)$. The capacity region $\mathcal{R}$ consists of the closure of all pairs (R_1,R_2) such that $\Pr[\hat{W}_i\neq W_i]\to 0$ as $n\to\infty$, $i=1,2$.

III. Upper Bounds

The capacity region $\mathcal{R}$ is a subset of the set $\mathcal{R}_{\mathrm{UB}}$ consisting of all pairs $(R_1,R_2)\geq 0$ for which

$$\begin{aligned}R_1&\leq 1-\epsilon_1\\ R_2&\leq 1-\epsilon_2\\ R_1+R_2&\leq 1-\epsilon_1\epsilon_2.\end{aligned} \tag{7}$$

This upper bound is very intuitive for the erasure channel because it also holds for broadcast erasure channels without feedback and without memory. A formal proof that neither memory nor feedback loosens the bounds is given in Appendix A. Whereas this upper bound is rather trivial and loose for the memoryless cases with and without feedback, as shown in Fig. 2, boundary points of $\mathcal{R}_{\mathrm{UB}}$ can be achieved when memory is present:

Consider the limiting cases where both g_i and b_i are very close to 0 or 1 for both users. Feedback about erasures in the past state almost completely removes the uncertainty about the next transmission. The transmitter knows almost exactly when packets will be lost and can adapt its transmission strategy accordingly. Coding is unnecessary as there is no uncertainty about the channel. When n is large, there will be $n(1-\epsilon_1)$ erasure-free slots to U_1, $n(1-\epsilon_2)$ erasure-free slots to U_2. $n(1-\epsilon_1\epsilon_2)$ slots are erasure-free to both users, so every boundary point on $\mathcal{R}_{\mathrm{UB}}$ can be approached even without time sharing, and thus $\mathcal{R}=\mathcal{R}_{\mathrm{UB}}$. This illustrates that memory does increase the capacity region of the broadcast erasure channel with feedback. Note that this is not the case for the point-to-point erasure channel with feedback and memory.

IV. Analysis of Algorithms

This section first reviews two capacity-achieving algorithms for the memoryless case from [1], called Algorithm I and Algorithm III. We then describe two alternative heuristic strategies. All algorithms are possible implementations of the encoder $X_t=f_t(W_1,W_2,Z_1^{t-1},Z_2^{t-1})$. In our case this is equivalent to an encoder $X_t=f_t(W_1,W_2,S_1^{t-1},S_2^{t-1})$ as the channel state S_t is determined by $Z_{1,t},Z_{2,t}$ without ambiguity.

The feedback informs the sender if the previous packet has been erased or not. In the following, packets destined for U_2 (U_1) that are received only by U_1 (U_2) are put into buffer Q_2 (Q_1) at the transmitter. Hence buffer Q_1 contains packets destined for U_1 that have been received at U_2 but not at U_1, and vice versa for Q_2. Each user keeps every received packet in a buffer, even if it is not destined for it.

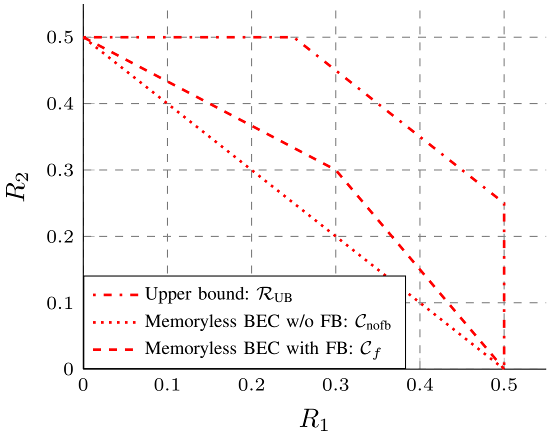

Fig. 2. Bounds for several rate regions and $\epsilon_1 = \epsilon_2 = 0.5$.

A. *Algorithm A*

Algorithm A corresponds to Algorithm I in [1] and consists of 3 phases for the sender:

1) Transmit each of the nR_1 packets destined for U_1 uncoded until it is received at either U_1 or U_2.
2) Same for the nR_2 packets destined for U_2.
3) Transmit random linear combinations of all packets in buffers Q_1 and Q_2 until both users can decode all its requested packets.

The corresponding rate region is denoted $\mathcal{R}_{\text{A}}$. Note that each of the phases of Algorithm A runs for long blocks and thus only long-term average erasure probabilities affect the performance, for sufficiently large values of n. As shown in [1], Algorithm A can approach every point in $\mathcal{C}_f$ for the memoryless case, so every rate pair $(R_1, R_2) \in \mathcal{C}_f$ can still be approached when memory is present for the corresponding average erasure probabilities ϵ_1, ϵ_2. Hence $\mathcal{R}_{\text{A}} = \mathcal{C}_f$, but the required block length n to approach points on the boundary will be larger in general for erasures with memory, as the underlying process has to reach steady state.

B. *Algorithm B*

Algorithm B corresponds to Algorithm III in [1] and requires only single-packet buffers Q_1, Q_2 and XOR coding operations. With slight abuse of notation, we write the buffer state in slot t as $Q_t = (Q_{1,t}, Q_{2,t}) \in \mathcal{Q}$. Q_t can take one of the values in $\mathcal{Q} = \{00, 10, 01, 11\}$, referring to the number of packets in Q_1, Q_2, respectively. Algorithm B consists of 2 phases for the sender:

1) As long as both users miss some of the requested packets, do the following steps:
   a) If $Q = 00$, transmit a packet for U_1.
   b) If $Q = 01$, transmit a packet for U_1.
   c) If $Q = 10$, transmit a packet for U_2.
   d) If $Q = 11$, transmit the binary addition (XOR) of the two packets in Q_1, Q_2.
2) Transmit the remaining packets uncoded to the user that is still missing some, until all packets have been delivered.

Steps $1.b)$ and $1.c)$ try to create coding opportunities, i.e. to create a situation where both buffers are nonempty and a coded packet is sent. If such a transmission is received by both users, 2 packets are delivered and one previous erasure is compensated. The average achievable transmission rates of the algorithm can be analyzed with a Markov chain with states corresponding to the 4 different queue states. The corresponding rate region is denoted $\mathcal{R}_{\text{B}}$. In contrast to Algorithm A, the performance of Algorithm B for the memoryless case differs from the performance for erasures with memory. To illustrate this, consider the following situation: Before time slot t, both buffers are empty, so the sender transmits a packet for U_1 according to step $1.a)$. The state in slot t turns out to be $S_t = \text{BG}$, so the packet is correctly received at U_2 but erased at U_1. In slot $t+1$, the sender transmits a packet for U_2 according to $1.c)$. If the channel to U_2 is persistent, then it is likely that this transmission is successfully received by U_2 and transmitting a packet for U_2 is a reasonable choice. However if the channel to U_2 is oscillatory, then it is likely that this packet is erased at U_2 (it might still be received at U_1). Numerical results will show that the performance of this algorithm degrades compared to the memoryless case if both channels are oscillatory. Algorithm B does not adapt to the channel, so algorithms exploiting channel state information can potentially perform better.

C. *Algorithm C*

In order to adapt to the channel state, we formulate the problem as a Markov Decision Process: We restrict us to single-packet buffers Q_1, Q_2, a possibly suboptimal choice. The overall state space is given by $\mathcal{S}_Q = \mathcal{S} \times \mathcal{Q}$ with 16 states. We write $S_{Q,t} = \text{GG00}$ if the channel state is $S_t = \text{GG}$ and the buffer state is $Q_t = 00$, and equivalently for the other cases.

In each slot t the sender chooses an action A_t from the set of actions $\mathcal{A} = \{a_1, a_2, a_3\}$, where

a_1 corresponds to transmission of a packet for U_1,
a_2 corresponds to transmission of a packet for U_2,
a_3 corresponds to transmission of a coded packet.

Note that the action A_t is based on the state in the previous slot $S_{Q,t-1}$, which is determined through the feedback. The possibilities for the successive state $S_{Q,t}$ also depend on the choice of A_t.

With each action $A_t = a$, a reward $\rho_i^a(s_Q, s'_Q)$ for user i is associated when going from state $S_{Q,t-1} = s_Q$ to $S_{Q,t} = s'_Q$ after action $A_t = a$. We choose this reward as the number of requested packets that reach U_i in that state transition. For example,

$$\rho_1^{a_1}(s_Q = \text{GB01}, s'_Q = \text{GG01}) = 1,$$
$$\rho_2^{a_1}(s_Q = \text{GB01}, s'_Q = \text{GG01}) = 0,$$

because both users receive a packet that is requested only by

TABLE I
POLICY FOR THE FIRST PHASE OF ALGORITHM C

Q_{t-1}	S_{t-1}	$A_t = a(S_{Q,t-1})$
11	arbitrary	a_3
$\neq 11$	GG	a_1 if $\theta_1(1-b_1) > \theta_2(1-b_2)$, otherwise a_2
	GB	a_1 if $\theta_1(1-b_1) > \theta_2 g_2$, otherwise a_2
	BG	a_1 if $\theta_1 g_1 > \theta_2(1-b_2)$, otherwise a_2
	BB	a_1 if $\theta_1 g_1 > \theta_2 g_2$, otherwise a_2

U_1. As another example, we have

$$\rho_1^{a_3}(s_Q = \text{GB11}, s_Q' = \text{GG00}) = 1,$$
$$\rho_2^{a_3}(s_Q = \text{GB11}, s_Q' = \text{GG00}) = 1,$$

because both users can decode a packet. Note that this case leads to a change of the buffer state. To facilitate coding opportunities, action a_3 is only permitted when $Q_t = 11$.

In each state S_Q we choose the action maximizing the expected weighted sum reward, i.e.

$$a(s_Q) = \arg\max_{a\in\mathcal{A}} \mathbb{E}[\theta_1\rho_1^a(s_Q, S_Q') + \theta_2\rho_2^a(s_Q, S_Q')], \quad (8)$$

where θ_1, θ_2 are two positive constants weighting the two rewards. The corresponding policy is summarized in Table I and determines the first of the two phases of Algorithm C:

1) As long as both users miss some of the requested packets, choose the action according to Table I.
2) Transmit the remaining packets uncoded to the user that is still missing some, until all packets have been delivered.

The long-term average rates of phase 1 can be analyzed with the corresponding Markov chain. Let π_{s_Q}, $s_Q \in \mathcal{S}_Q$ be the steady-state probability of the Markov chain. The long-term average rates for user i in phase 1 is given by

$$\begin{aligned}\bar{\rho}_i &= \sum_{s_Q\in\mathcal{S}_Q} \pi_{s_Q}\mathbb{E}[\rho_i^{a(s_Q)}(s_Q, S_Q')] \\ &= \sum_{s_Q\in\mathcal{S}_Q}\sum_{s_Q'\in\mathcal{S}_Q} \pi_{s_Q}\Pr[s_Q'|s_Q, a(s_Q)]\rho_i^{a(s_Q)}(s_Q, s_Q'). \quad (9)\end{aligned}$$

Although straightforward in principle, computer algebra systems are needed to solve π_{s_Q} analytically due to the cardinality of the state space. Unfortunately the resulting expressions for $\bar{\rho}_i$ become too complicated to provide further insight to the problem. Similar problems arise for the analysis of Algorithm B. For performance comparisons, we have to resort to numerical evaluations. Based on $\bar{\rho}_i$, the rate region $\mathcal{R}_\text{C}(\theta_1, \theta_2)$ of Algorithm C can be computed, for the choice of θ_1 and θ_2. See Appendix B for details. We define $\mathcal{R}_\text{C}$ as the convex hull of the union of all regions $\mathcal{R}_\text{C}(\theta_1, \theta_2)$ for all possible choices of θ_1, θ_2.

D. Algorithm D

The motivating example in Section III showed that coding might not be necessary if the uncertainty about future transmissions is rather small. For Algorithm D the action space is reduced to $\mathcal{A} = \{a_1, a_2\}$, so coding operations are not permitted. Hence buffer contents cannot be exploited and the relevant state space reduces to the channel state space $\mathcal{S}$. Similar to before, a reward $\rho_i^a(s, s')$ is associated for user i when going from state $S_{t-1} = s \in \mathcal{S}$ to $S_t = s' \in \mathcal{S}$ after action $A_t = a$.

TABLE II
POLICY FOR THE FIRST PHASE OF ALGORITHM D

S_{t-1}	$A_t = a(S_{t-1})$
GG	a_1 if $\theta_1(1-b_1) > \theta_2(1-b_2)$, otherwise a_2
GB	a_1 if $\theta_1(1-b_1) > \theta_2 g_2$, otherwise a_2
BG	a_1 if $\theta_1 g_1 > \theta_2(1-b_2)$, otherwise a_2
BB	a_1 if $\theta_1 g_1 > \theta_2 g_2$, otherwise a_2

The policy for Algorithm D is given by

$$a(s) = \arg\max_{a\in\{a_1,a_2\}} \mathbb{E}[\theta_1\rho_1^a(s, S') + \theta_2\rho_2^a(s, S')]. \quad (10)$$

The resulting policy is summarized in Table II and determines the first of the two phases of Algorithm D:

1) As long as both users miss some of the requested packets, choose the action according to Table II.
2) Transmit the remaining packets uncoded to the user that is still missing some, until all packets have been delivered.

The long-term average rates of phase 1 can be computed similarly as before. The rate region for Algorithm D is denoted by $\mathcal{R}_\text{D}(\theta_1, \theta_2)$. Again, let $\mathcal{R}_\text{D}$ be the convex hull of the union of all regions $\mathcal{R}_\text{D}(\theta_1, \theta_2)$ for all possible choices of θ_1, θ_2.

V. COMPARISON AND DISCUSSION

We first note that Algorithms B, C, and D react to different state spaces. Whereas Algorithm B only depends on queue states and Algorithm D only on channel states, Algorithm C bases its decisions on the combination of both. We also note that the performance of Algorithm C is at least as good as the performance of Algorithm D. We pick some typical examples showing a general trend of the simulations. In the following figures, we plot rate regions for all algorithms as well as lower and upper bounds. Additionally, we add the rate regions for Algorithm C and D for the particular choice $\theta_1 = \theta_2 = 1$. Consider Fig. 3: We observe that Algorithm B improves quite a bit compared to the memoryless capacity region. We also note that Algorithm B can perform better or worse than Algorithms C and D for $\theta_1 = \theta_2 = 1$, depending on the rates. The rate regions for all choices of θ_1, θ_2 ($\mathcal{R}_\text{C}$ and $\mathcal{R}_\text{D}$) however provide higher rates than Algorithm B. We further see that the difference between Algorithms C and D is not large.

Fig. 4 however shows that also Algorithm B can provide rate points that are outside of $\mathcal{R}_\text{C}$ and $\mathcal{R}_\text{D}$. In general we observe that Algorithm B works quite well for persistent memory. For oscillatory memory however, like in Fig. 5, we see that Algorithm B performs even worse than Algorithm A, whereas Algorithms C and D can adapt to the channel state and increase the rate region quite significantly. The difference between both regions is small again, raising the conjecture that the coding gain is actually decreased when memory and feedback is present.

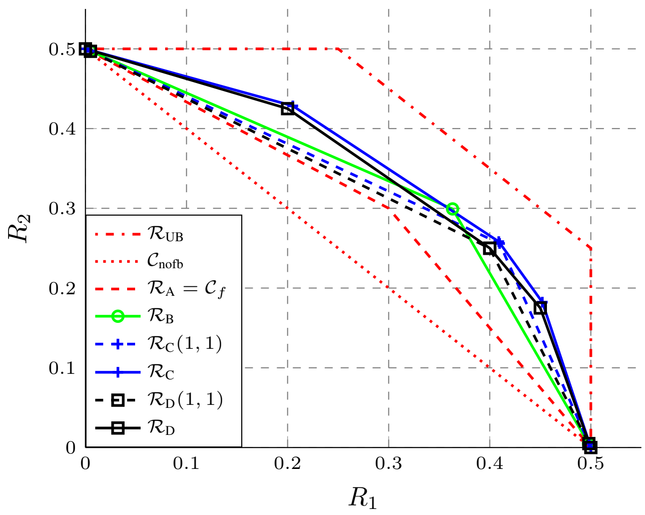

Fig. 3. Rate regions for $\epsilon_1 = \epsilon_2 = 0.5$, $g_1 = 0.2$, $g_2 = 0.3$.

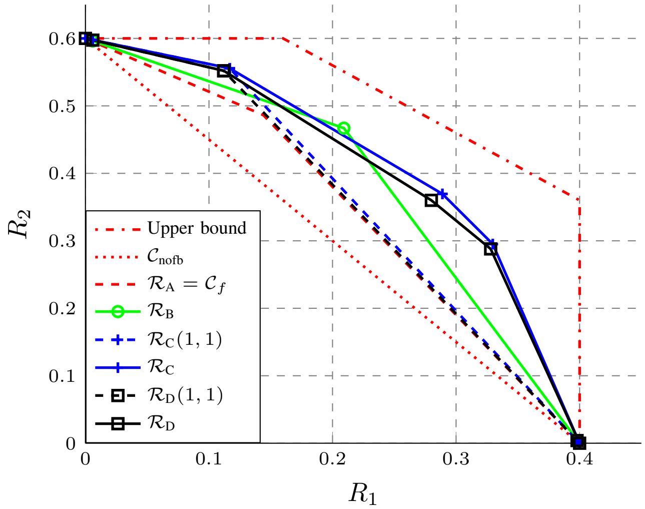

Fig. 4. Rate regions for $\epsilon_1 = 0.6$, $\epsilon_2 = 0.4$, $g_1 = 0.2$, $g_2 = 0.3$.

VI. Conclusion

The two-user broadcast erasure channel with feedback and memory was investigated. Starting from extreme cases, where upper bounds can be achieved we analyze different heuristic strategies. Significant gains can be observed compared to the case without memory and motivate the development of further strategies. The capacity region for this setup was found recently and a publication is in preparation.

Acknowledgments

The authors are supported by the German Ministry of Education and Research in the framework of the Alexander von Humboldt-Professorship and by the grant DLR@Uni of the Helmholtz Allianz. The authors thank Gianluigi Liva and Andrea Munari for discussions and the motivation of the problem, Navid Reyhanian for pointing out an error in an earlier version of the paper and Gerhard Kramer for his helpful comments.

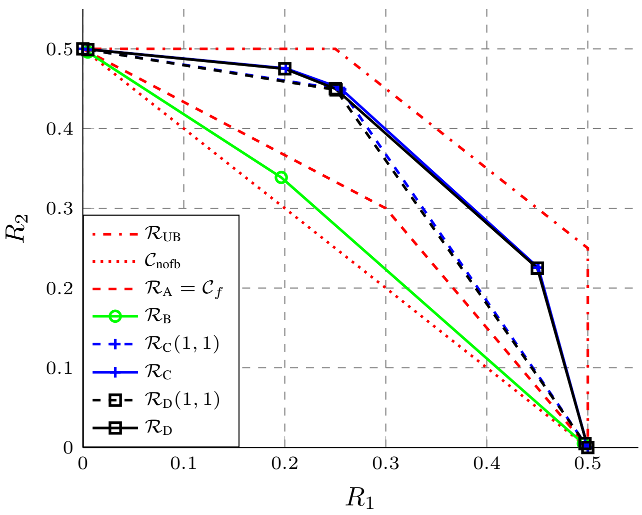

Fig. 5. Rate regions for $\epsilon_1 = \epsilon_2 = 0.5$, $g_1 = 0.8$, $g_2 = 0.9$.

Appendix A
Proof of Upper Bound

Consider a point-to-point erasure channel with memory and feedback with message $W \in \{1, 2, \ldots, q^{nR}\}$, channel input $X_t = f_t(W, Z^{t-1})$, $t = 1, \ldots, n$, channel state sequence S^n, channel output Y^n and erasure indicator Z^n. The joint probability mass function factorizes as

$$\begin{aligned}
&p(w, x^n, y^n, z^n, s^n)\\
&= p(w)p(s_0)\prod_{t=1}^{n} p(s_t|s_{t-1})p(z_t|s_t)p(x_t|w, z^{t-1})p(y_t|x_t, z_t).
\end{aligned}$$

We have

$$\begin{aligned}
nR &\le I(X^n \to Y^n) = \sum_{t=1}^{n} I(X^t; Y_t|Y^{t-1})\\
&\overset{(a)}{=} \sum_{t=1}^{n} \left[I(X_t; Y_t|Y^{t-1}) + I(X^{t-1}; Y_t|Y^{t-1}, X_t)\right]\\
&\overset{(b)}{=} \sum_{t=1}^{n} \left[I(X_t; Y_t|Y^{t-1}) + I(X^{t-1}; Y_t|Y^{t-1}, Z^{t-1}, X_t)\right]\\
&\overset{(c)}{=} \sum_{t=1}^{n} \left[H(X_t|Y^{t-1}) - H(X_t|Y^t)\right]\\
&= \sum_{t=1}^{n}\sum_{y^{t-1}} p(y^{t-1}) \left[H(X_t|y^{t-1}) - H(X_t|Y_t, y^{t-1})\right]\\
&= \sum_{t=1}^{n}\sum_{y^{t-1}} p(y^{t-1}) \left[H(X_t|y^{t-1}) -\right.\\
&\qquad \left.-p(Y_t = E|y^{t-1})H(X_t|Y_t = E, y^{t-1})\right]\\
&\overset{(d)}{=} \sum_{t=1}^{n}\sum_{y^{t-1}} p(y^{t-1})H(X_t|y^{t-1}) \left[1 - p(Y_t = E|y^{t-1})\right]\\
&\overset{(e)}{\le} n - \sum_{t=1}^{n} p(Y_t = E) \qquad (11)
\end{aligned}$$

where $I(X^n \to Y^n)$ denotes the directed information from a sequence X^n to Y^n as in [15], [16],

(a) is due to the chain rule of mutual information,

(b) is due to the fact that Z^{t-1} is a function of Y^{t-1},

(c) is due to the fact that X^{t-1} is conditionally independent of Y_t given (Y^{t-1}, Z^{t-1}, X_t),

(d) is due to the fact that $H(X_t|Y_t = E, y^{t-1}) = H(X_t|y^{t-1})$ and

(e) is due to the fact that $H(X_t|y^{t-1}) \leq 1$.

The first two inequalities in (7) follow immediately, the third one follows from letting both users cooperate.

Appendix B
Rate Region Computation

To compute the rate regions for Algorithm B, C and D in Sect. V, the following method is applied (similar to [1]): Recall that the average number of packets per slot in phase 1 for U_i is given by $\bar{\rho}_i$.

U_i requests $k_i = nR_i$ packets. The first phase of Algorithms B, C and D last as long as either U_1 or U_2 has received all of its requestes packets. The duration of phase 1 is denoted by n_1, the duration of phase 2 by n_2 and $n = n_1 + n_2$.

The first phase will last for $n_1 = \min\left\{\frac{k_1}{\bar{\rho}_1}, \frac{k_2}{\bar{\rho}_2}\right\}$ slots on average, for large k_1, k_2. Let α denote the ratio between R_2 and R_1:

$$\alpha = \frac{R_2}{R_1} = \frac{k_2}{k_1}. \tag{12}$$

$$\text{It follows that } n_1 = k_1 \min\left\{\frac{1}{\bar{\rho}_1}, \frac{\alpha}{\bar{\rho}_2}\right\}. \tag{13}$$

Two cases must be distinguished:

- If $\bar{\rho}_2 < \alpha\bar{\rho}_1$, then U_1 has finished after phase 1 and $n_1 = \frac{k_1}{\bar{\rho}_1}$. The remaining $k_2 - n_1\bar{\rho}_2 = k_1\left(\alpha - \frac{\bar{\rho}_2}{\bar{\rho}_1}\right)$ packets must be transmitted to U_2. This takes
$$n_2 = \frac{k_1\left(\alpha - \frac{\bar{\rho}_2}{\bar{\rho}_1}\right)}{1 - \epsilon_2}$$
slots and yields the rates
$$R_1 = \frac{k_1}{n_1 + n_2} = \frac{1}{\frac{1}{\bar{\rho}_1} + \frac{\alpha - \bar{\rho}_2/\bar{\rho}_1}{1-\epsilon_2}} = \frac{\bar{\rho}_1}{1 + \frac{\alpha\bar{\rho}_1 - \bar{\rho}_2}{1-\epsilon_2}}, \tag{14}$$
$$R_2 = \alpha R_1 = \frac{\alpha\bar{\rho}_1}{1 + \frac{\alpha\bar{\rho}_1 - \bar{\rho}_2}{1-\epsilon_2}}. \tag{15}$$
- If $\bar{\rho}_2 > \alpha\bar{\rho}_1$, then U_2 has finished after phase 1 and $n_1 = \frac{\alpha k_1}{\bar{\rho}_2}$. The remaining $k_1 - n_1\bar{\rho}_1 = k_1\left(1 - \alpha\frac{\bar{\rho}_1}{\bar{\rho}_2}\right)$ packets must be transmitted to U_1. This takes
$$n_2 = \frac{k_1\left(1 - \alpha\frac{\bar{\rho}_1}{\bar{\rho}_2}\right)}{1 - \epsilon_1}$$
slots and yields the rates
$$R_1 = \frac{k_1}{n_1 + n_2} = \frac{1}{\frac{\alpha}{\bar{\rho}_2} + \frac{1 - \alpha\bar{\rho}_1/\bar{\rho}_2}{1-\epsilon_1}} = \frac{\bar{\rho}_2}{\alpha + \frac{\bar{\rho}_2 - \alpha\bar{\rho}_1}{1-\epsilon_1}}, \tag{16}$$
$$R_2 = \alpha R_1 = \frac{\alpha\bar{\rho}_2}{\alpha + \frac{\bar{\rho}_2 - \alpha\bar{\rho}_1}{1-\epsilon_1}} \tag{17}$$

A plot of the boundary of the rate region can be obtained by choosing different values for α.

References

[1] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback-capacity and algorithms," in *NetCod'09. Workshop on Network Coding, 2009.* IEEE, 2009, pp. 54–61.

[2] C.-C. Wang, "On the capacity of 1-to-k broadcast packet erasure channels with channel output feedback," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 931–956, 2012.

[3] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Multiuser broadcast erasure channel with feedback – capacity and algorithms," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5779–5804, Sept 2013.

[4] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[5] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *NetCod'08. Workshop on Network Coding, 2008.* Citeseer, 2008, pp. 1–6.

[6] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *NetCod'09. Workshop on Network Coding, 2009.*, 2009, pp. 1–6.

[7] M. A. R. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," in *IEEE INFOCOM Workshops 2008*. IEEE, 2008, pp. 1–4.

[8] Y. Birk and T. Kol, "Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2825–2830, 2006.

[9] C.-C. Wang and D. J. Love, "Linear network coding capacity region of 2-receiver mimo broadcast packet erasure channels with feedback," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2012, pp. 2062–2066.

[10] X. Song and O. Iscan, "Network coding for the broadcast rayleigh fading channel with feedback," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2012, pp. 2057–2061.

[11] R. Dabora and A. J. Goldsmith, "Capacity theorems for discrete, finite-state broadcast channels with feedback and unidirectional receiver cooperation," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 5958–5983, 2010.

[12] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[13] E. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963.

[14] M. Mushkin and I. Bar-David, "Capacity and coding for the gilbert-elliott channels," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277–1290, 1989.

[15] H. Marko, "The bidirectional communication theory–a generalization of information theory," *IEEE Transactions on Communications*, vol. 21, no. 12, pp. 1345–1351, 1973.

[16] J. Massey, "Causality, feedback and directed information," in *IEEE International Symposium on Information Theory and Applications (ISITA)*, 1990, pp. 303–305.