Stable and Capacity Achieving XOR-based Policies for the Broadcast Erasure Channel with Feedback

Sophia Athanasiadou*, Marios Gatzianas** ††, Leonidas Georgiadis* † § and Leandros Tassiulas‡ † §
* Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, 54 124, Greece.
† Centre for Research and Technology Hellas, Thermi, Thessaloniki, 60 361, Greece.
‡ Computer Engineering and Telecommunications Dept., Volos 38 221, Greece.
** School of Computer and Communication Sciences, EPFL, Switzerland.
Emails: sathanasiadou@ee.auth.gr, marios.gkatzianas@epfl.ch, leonid@auth.gr, leandros@uth.gr

***Abstract*—In this paper we describe a network coding scheme for the Broadcast Erasure Channel with multiple unicast stochastic flows, for the case of a single source transmitting packets to N users, where per-slot feedback is fed back to the transmitter in the form of ACK/NACK messages. This scheme performs only binary (XOR) operations and includes special rules for coding packets that ensure instantaneous decodability. Drawing on the results of network stability under statistical overhearing, we provide a stabilizing policy using this coding scheme. Furthermore, we show that, for $N = 4$ and i.i.d. erasure events, the stability region of such a system effectively coincides with its information-theoretic capacity region, and provide a stabilizing policy that employs this XOR-based scheme.**

I. Introduction

The information-theoretic capacity region of the Broadcast Erasure Channel (BEC) in the case of one transmitter and N multiple unicast sessions has been recently determined in [1] and [2]. Both of these papers proposed capacity achieving algorithms based on transmission of linear combinations of packets. However, these schemes are characterized by high complexity (as operations take place in a sufficiently large sized finite field) and decoding delay, since a sufficient number of linear combinations has to be received until a packet is decoded. In [3], we proposed a network coding scheme that overcomes these obstacles by using only XOR operations, generalizing the 2 user network coding scheme in [4] to the case of 3 users. Thus, two low complexity algorithms were proposed, namely `XOR1` and `XOR2`, which additionally had the advantageous property of "instantaneous decodability", i.e. a receiver is able to decode packet p destined for it as soon as it receives an XOR combination of packets containing p. Algorithm `XOR2` was proved to achieve capacity for the case of i.i.d. channels as well as spatially independent channels with erasure probabilities that do not exceed 8/9.

The network coding scheme proposed in the present paper is based on this framework. All packets arriving at the transmitter are placed in one of N queues. To exploit the feedback information of the broadcast channel, we construct a set of virtual packet queues at the transmitter. Each virtual queue reflects the information the transmitter has obtained (through feedback) about the packets in that queue. After each transmission, and based on the received feedback, packets are moved among the queues until they are successfully received by their intended destination. Coding and packet movement rules are imposed in order to ensure instantaneous decodability of packets and better exploitation of coding opportunities.

The system considered in [3] is a saturated system, where a predefined number of packets needs to be transmitted to each user. This model is not frequently encountered in practice. Moreover, `XOR1` and `XOR2` cannot be easily generalized to more than 3 users. The reason for this is that according to these algorithms, at each time slot, coding choices have to be rigorously determined so that each transmission is optimally exploited, in terms of allowing multiple users to simultaneously decode their packets as well as create favorable future coding opportunities. However, for $N > 3$, the number of coding choices increases dramatically so that there is no clear intuition on the optimal choice.

The contribution of this paper is twofold: 1) we develop a systematic framework for constructing instantaneously decodable feedback-based XOR coding schemes for the BEC with multiple unicast sessions and arbitrary N. This is accomplished by using a backpressure-type online algorithm that makes each coding choice based on instantaneous quantities (i.e. queue sizes). While in [3] we examined a saturated system, in this paper we consider the more realistic model where packets may arrive randomly at the transmitter at any time slot. 2) For the specific case of 4 users and i.i.d. erasure events, we present a stabilizing policy on top of this network coding scheme and prove that proper XOR combining, ensuring instantaneous decoding, is sufficient to achieve the capacity of the 4-user BEC with i.i.d. erasures, even though the proposed policy uses only a subset of all possible coding choices.

Network stability has also been studied along with network coding schemes in [5] and [6], albeit in different contexts. In [5], the stability of single hop BEC with feedback is examined but the traffic considered is broadcast and linear network coding schemes are used, so there is no guarantee for instantaneous decodability. In [6], there are multiple unicast flows but the model examined is different, as there exists a relay node overhearing transmissions which then transmits XOR packets through an error-free channel.

The paper is structured as follows: in Sections II and III the system model and network coding scheme are described, while in Section IV we present the proposed policy. We also derive an outer bound on the stability region of the system and prove that for $N = 4$ and i.i.d. erasure events the stability region of such a system coincides with the capacity upper bound of the standard broadcast erasure channel with feedback. Section

[§] This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: Thales. Investing in knowledge society through the European Social Fund. [††] M. Gatzianas is supported by the ERC Starting Project Grant NOWIRE ERC-2009-StG-240317.

V concludes the paper. Due to space constraints, all proofs of the presented results are omitted and can be found in [7], which also contains some examples of the proposed scheme's application that illustrate its finer points.

II. System model and notation

Sets are denoted by calligraphic letters, e.g. $\mathcal{M}$, and the empty set by $\emptyset$. The cardinality of set $\mathcal{M}$ is denoted by $|\mathcal{M}|$ and we write $M = |\mathcal{M}|$. Random variables are denoted by capital letters and their values by small case letters. Vectors are denoted by bold letters, e.g. $\boldsymbol{A} = (A_1, ..., A_n)$.

We consider a time-slotted system where slot t corresponds to the time interval $[t,\ t+1)$. The system consists of a base station B and a set $\mathcal{N} = \{1, 2, ..., N\}$ of receivers (users). At the beginning of slot t, $A_i(t)$ data packets arrive at B with rate $\lambda_i = \mathbb{E}(A_i)$; these packets must be delivered to receiver i and are referred to as "flow i" packets, where $\boldsymbol{A}(t) = (A_1(t), ..., A_N(t))$. All packets consist of L bits with an individual transmission-time duration of 1 slot. A packet broadcasted by B may be either correctly received or completely erased by any receiver.

After each transmission, the receivers send feedback to B (through an error-free zero-delay channel) informing whether the transmitted packet has been correctly received or not (ACK/NACK feedback). We also assume that if no packet is transmitted in a slot (idle slot), then all receivers realize that the slot is idle. Packet arrivals and packet erasures are assumed to be independent and identically distributed across time, but arbitrarily correlated across users.

The proposed policy operates on a network of queues, constructed at the transmitter, and moves packets among the queues according to certain rules (namely, the Basic Coding Rule and Rules for Packet Movement, to be introduced shortly). Since exogenous packets arrive randomly, the important concept of stability is defined as follows: let $\hat{Q}^{\pi}(t)$ be the aggregate number of packets in all queues under the application of a policy $\pi \in \Pi$, where Π is the set of policies that satisfy the Basic Coding Rule and Rules for Packet Movement. An arrival vector $\boldsymbol{\lambda}$ is stable under $\pi \in \Pi$ iff it holds $\lim_{q\to\infty} \limsup_{t\to\infty} \Pr(\hat{Q}_{\pi}(t) > q) = 0$. The stability region $\mathcal{R}^{\pi}$ of π is the closure of the set of all arrival rates that are stable under π while the stability region $\mathcal{R}_{\Pi}$ under the class of policies Π is the closure of the set $\cup_{\pi\in\Pi}\mathcal{R}^{\pi}$. A policy $\pi^* \in \Pi$ is stabilizing within Π iff it holds $\mathcal{R}^{\pi^*} = \mathcal{R}_{\Pi}$.

III. Network coding scheme description

Exogenous packets arriving at B and being intended for user $i \in \mathcal{N}$ are called "native packets for i". A packet is simply termed "native" if it is a native packet for some user (due to the unicast traffic, a packet is native for exactly one user). According to the policies to be described below, all transmitted packets are either native, or XOR combinations of native packets. Hence, any transmitted packet p can be written as $p = \bigoplus_{l=1}^{n} q_l$, where q_l are native packets, and we say that "p contains q_l". It is possible, and actually beneficial, for p to contain native packets for more than one user. To shorten the subsequent description, we say that a packet p is an XOR combination of native packets even when p contains a single native packet. The following definitions will be crucial in the subsequent analysis.

Definition 1. User i is a *listener* of a packet p iff both of the following conditions are true: 1) p is an XOR combination of packets, not necessarily native, that i has correctly received, and 2) p contains no native packet for i that is unknown to i. Equivalently, if p contains a native packet q for user i, then the packet q is known to (i.e. has already been decoded by) i.

Definition 2. User i is a *destination* of a packet p iff either p is a native packet intended for user i that has not been received yet by (i.e. is unknown to) i, or p can be decomposed as an XOR combination of the form $p = q \oplus c$, where 1) q is a native packet intended for i and unknown to i, and 2) i is a *listener* of c.

To clarify the above definitions, we give two examples. Let us denote all native packets for users i, j with $\tilde{p}, \tilde{q}$, respectively. Suppose $c = \tilde{p} \oplus \tilde{q}$ is transmitted, where $\tilde{p}, \tilde{q}$ have been previously received by j, i, respectively. Then, according to Definition 2, both i and j are *destinations* for c. Suppose now that c is received by a third user k, so that k is a *listener* for c, and c is transmitted again in the future, either individually or as part of some XOR combination. If i receives c (or is able to deduce its value, in case c is part of an XOR combination) in the future, then i instantly decodes its native packet $\tilde{p}$, ceases to be a *destination* for c, and becomes a *listener* for c, as c no longer contains a native packet of i that is unknown to i.

In the second example, suppose $c = \tilde{p}_l \oplus \tilde{q}_l$ is transmitted and received by i, where neither $\tilde{p}_l$ nor $\tilde{q}_l$ has been received by i. Then, according to Definition 1, i is not a *listener* of c, since it cannot decode $\tilde{p}_l$. Suppose now that $\tilde{c} = \tilde{p}_m \oplus c$ is transmitted next and received by i. Then, i is not a *destination* for $\tilde{c}$ (since Definition 2 would require i to be a *listener* of c) even though i is able to decode $\tilde{p}_m$. Since $\tilde{c}$ is an innovative packet for i, we conclude that the notion of "i is a *destination* for $\tilde{c}$" is stronger than "$\tilde{c}$ is innovative for i".

According to Definition 2, if user i is a *destination* for packet p, then p contains a unique unknown native packet for i, which we denote by $q = p(i)$ and call $p(i)$ the "unknown packet" of i in p. As will be seen, transmitted packets may have several users as *destinations* or *listeners*. The main features of the policies to be presented are as follows.

Algorithmic Features

1) Any transmitted packet is an XOR combination of native packets.
2) If a transmitted packet (equivalently, an XOR combination) p contains a native packet q intended for $i \in \mathcal{N}$, which is unknown to i, then i is a *destination* for p. Hence, since according to the definition of *destination* it holds either $p = q$ or $p = q \oplus c$, where c is known to i, user i can immediately decode q as $q = p \oplus c$.

Under the proposed policies, packets may be placed in various virtual queues, based on the received feedback. A general virtual queue $Q_{\mathcal{D}}^{\mathcal{L}}$ is characterized by two index sets $\mathcal{L}, \mathcal{D}$ satisfying the following criterion:

Compatibility Criterion (CC) for sets $\mathcal{L}, \mathcal{D}$

It must hold $\mathcal{L}, \mathcal{D} \subseteq \mathcal{N}$, $\mathcal{L} \cap \mathcal{D} = \emptyset$ and $\mathcal{D} \neq \emptyset$. Additionally, $\mathcal{L} = \emptyset$ only if $|\mathcal{D}| = 1$.

For simplicity, we will denote (virtual) queue $Q_{\{i,j\}}^{\{k\}}$ by Q_{ij}^{k}, and queue $Q_{\{i\}}^{\emptyset}$ by Q_i. We denote with $p_{\mathcal{D}}^{\mathcal{L}}$ a packet that is stored in virtual queue $Q_{\mathcal{D}}^{\mathcal{L}}$ and write $p_{\mathcal{D}}^{\mathcal{L}} \in Q_{\mathcal{D}}^{\mathcal{L}}$ (i.e. $\in$ denotes storage, when applied to packets). The proposed policies ensure that the following Basic Properties hold for any packet $p \in Q_{\mathcal{D}}^{\mathcal{L}}$:

Basic Properties (BP) of packets $p \in Q_{\mathcal{D}}^{\mathcal{L}}$:

1) Any receiver $i \in \mathcal{D}$ is a *destination* for p and any $i \in \mathcal{L}$ is a *listener* of p.
2) If p contains a native packet for user $i \notin \mathcal{D}$, then this native packet has already been decoded by (i.e. is known to) i. Equivalently, p contains no unknown native packet for any user in $\mathcal{D}^c$, where ^{c} denotes set complement.
3) For any native packet q for user i that has not yet been decoded by i, there exists exactly one packet $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ (for some $\mathcal{L}$, $\mathcal{D}$) such that $q = p(i)$.

Thus, each packet $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ can be written as $p = \oplus_{i\in\mathcal{D}} p(i) \oplus d$.

The Basic Properties justify, in retrospect, the conditions imposed on $\mathcal{D}, \mathcal{L}$ via CC. Specifically, the fact that $\mathcal{D}, \mathcal{L}$ contain *destinations* and *listeners*, respectively, for a packet p implies that $\mathcal{L} \cap \mathcal{D} = \emptyset$, since p cannot contain any packet that is unknown to a *listener* user (condition 2 of Definition 1). Hence, a *listener* can never be a *destination,* although a *destination* for a packet becomes a *listener* upon reception of the packet. The condition $\mathcal{D} \neq \emptyset$ captures the fact that a packet need only be stored in the queues for as long as it contains an unknown native packet for at least one user. Finally, before any transmissions occur, each native packet has a singleton *destination* set and an empty *listener* set (once a packet p is transmitted and received by at least one user, it acquires at least one *listener*).

We note that queues $Q_{\mathcal{D}}^{\mathcal{L}}$ are "virtual" in the sense that the packets need not actually be held in these queues. Instead, there are N "real" queues Q_i^R at B, where queue Q_i^R, $i \in \mathcal{N}$, contains native packets intended for user i that are unknown to this user. When a packet of flow $i \in \mathcal{N}$ arrives at B, it is placed in queue Q_i^R. The virtual queues $Q_{\mathcal{D}}^{\mathcal{L}}$ describe information known by the transmitter (through the received feedback) about the packets in the real queues.

Since any packet p is an XOR combination of native packets, the notation $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ (which we previously read as "p is stored in $Q_{\mathcal{D}}^{\mathcal{L}}$") should actually be interpreted as: $Q_{\mathcal{D}}^{\mathcal{L}}$ contains a data structure, corresponding to p, that only stores pointers to the real native packets that comprise p. Accessing this structure and XORing all native packets pointed to by its contents allows to recreate the "actual" packet p. Hence, at any instance in time, we can partition the "real" queue Q_i^R into disjoint "real" queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$, $i \in \mathcal{D}$, where packet $q \in Q_i^R$ is also in $Q_{\mathcal{D}}^{\mathcal{L}}(i)$ if for some $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ it holds $q = p(i)$. The existence of a packet p satisfying the last condition is guaranteed by the third BP.

The following example illustrates the relation between real and virtual queues: let $q_i^{\mathcal{L}}$ be a native packet intended for user i that has been received by all users in $\mathcal{L}$ (hence, all users in $\mathcal{L}$ are *listeners* for q_i) and consider the packets q_1^2, q_2^1. Since these packets have not been delivered to their destinations, they are stored in real queues Q_1^R, Q_2^R, respectively. Assume the transmitter next decides to send packet $p = q_1^2 \oplus q_2^1$ (it will be seen that this is a valid choice) and p is received only by user 3 so that we can write $p = (q_2^1 \oplus q_1^2)^3$ where the upper index always denotes *listeners*. It follows immediately that $p \in Q_{12}^3$ and $q_1^2 \in Q_{12}^3(1)$, $q_2^1 \in Q_{12}^3(2)$. Only packets q_1^2, q_2^1 are "actually" stored in the queues as individual L-bit entities; packet p, which is stored in the virtual queue Q_{12}^3 need only contain pointers (or links) to q_1^2, q_2^1.

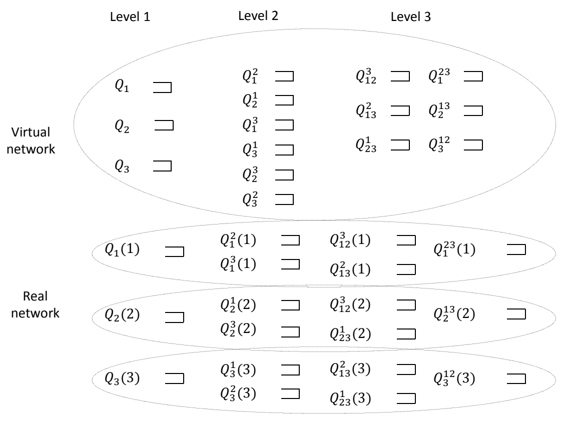

Figure 1. Virtual and real queues.

Therefore, viewing each queue as a set of packets, we have $Q_i^R = \bigcup_{\mathcal{L},\mathcal{D}} Q_{\mathcal{D}}^{\mathcal{L}}(i)$, where the union is over all sets $\mathcal{L}, \mathcal{D}$ satisfying CC. Notationally, the real and virtual queues are easily distinguished by the fact that the former always have an index in parentheses.

The virtual queues are helpful in the description of the policies to be presented. The real queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$ will be useful in the analysis performed in Section IV. We classify virtual queues into N levels, where level $w \in \{1, ..., N\}$ contains all queues $Q_{\mathcal{D}}^{\mathcal{L}}$ such that $|\mathcal{L} \cup \mathcal{D}| = w$. Moreover, we classify queues of level $w \geq 3$ into sublevels, where sublevel $w.u$ includes queues of level w with $|\mathcal{L}| = u$, $u \in \{1, ..., w-1\}$. In Fig. 1, we give an example of virtual and real queues when $N = 3$, where $Q_i(i)$ is an alias for Q_i^R.

In general, during the operation of the system, XOR combinations of packets are transmitted, at most one from each of the virtual queues $Q_{\mathcal{D}}^{\mathcal{L}}$. While the actual choice of packets depends on the received feedback and the specific algorithm that is employed, the following rule, which is dictated by the instantaneous decodability requirement, always holds.

Basic Coding Rule (BCR)

A set $\mathcal{P} = \left\{p_{\mathcal{D}_1}^{\mathcal{L}_1}, \ldots, p_{\mathcal{D}_m}^{\mathcal{L}_m}\right\}$ of $m = |\mathcal{P}|$ packets, *one* from each of the *different* virtual queues $\left\{Q_{\mathcal{D}_1}^{\mathcal{L}_1}, \ldots, Q_{\mathcal{D}_m}^{\mathcal{L}_m}\right\}$, can be combined (by XORing) into a single coded packet only if

$$\mathcal{D}_n \subseteq \mathcal{L}_r, \ \forall r \neq n, \ n, r \in \{1, ..., m\}. \tag{1}$$

In other words, the *destination* set of each packet is a subset of each of the others packets' *listener* sets.

Note that the BCR implies that $\mathcal{D}_n \cap \mathcal{D}_r = \emptyset$, for all $r \neq n$, $n, r \in \{1, ..., m\}$. Indeed, $i \in \mathcal{D}_n$ implies, through (1), that $i \in \mathcal{L}_r$ and, since according to CC it holds $\mathcal{D}_r \cap \mathcal{L}_r = \emptyset$, it follows that $i \notin \mathcal{D}_r$. In fact, BP and the BCR imply that any user $i \in \cup_{n=1}^m \mathcal{D}_n$ that receives the transmitted XOR combination will be able to instantaneously decode its native packet. The following Lemma ensures that Algorithmic Features are satisfied at all time slots.

Lemma 3. *Assume that Algorithmic Features 1, 2 are satisfied at the beginning of slot t. Then, if the packet to be transmitted at slot t is composed using the BCR, Algorithmic Features 1, 2 are also satisfied at slot $t+1$.*

Next, we describe how packets are moved between virtual queues $Q_{\mathcal{D}}^{\mathcal{L}}$ (and, consequently, between the corresponding real queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$), based on the received feedback. Generally, packets may be placed in various queues such that the Basic Properties are satisfied. For example, assume that packet $p = p_{12}^{3} \oplus p_{3}^{12}$ is transmitted (p satisfies the BCR), and that only user 2 receives the packet; hence packet $p_{12}^{3}(2)$ reaches its destination and is removed from the real queue $Q_{12}^{3}(2)$. For the other packet movements, two choices are available:

1) Packet p_{12}^{3} is moved to virtual queue Q_{1}^{23} and p_{3}^{12} is not moved: regarding the real queues, only packet $p_{12}^{3}(1)$ is moved to queue $Q_{1}^{23}(1)$. This is consistent with BP since, after receiving p, receiver 2 becomes a *listener* for $p_{12}^{3} = p \oplus p_{3}^{12}$, while receiver 3 is already a *listener* for that packet.

2) Packet p is moved to virtual queue Q_{13}^{2} and p_{12}^{3}, p_{3}^{12} are removed from virtual queues Q_{12}^{3}, Q_{3}^{12} respectively: hence, packet $p_{12}^{3}(1)$ is moved to queue $Q_{13}^{2}(1)$ and $p_{3}^{12}(3)$ is moved to queue $Q_{13}^{2}(3)$. Again, this is consistent with BP.

Intuitively, the higher the level (and within the same level the higher the sublevel) of a virtual queue in which a packet p is stored, the better are the opportunities of combining the packet with other packets in such a way that multiple successful receptions are effected with a single transmission. Hence, when multiple choices for packet movement arise, we select the one that ensures that all packets involved in a transmission are placed in a higher level (or within the same level but at higher sublevels), else they are not moved. Thus, in the example above, we choose the first option. This is the underlying rationale on which packet movement is based in the following description.

Rules for Packet Movement (RPM): Let packet p of the form $p = p_{\mathcal{D}_1}^{\mathcal{L}_1} \oplus \ldots \oplus p_{\mathcal{D}_m}^{\mathcal{L}_m}$ satisfying the BCR be chosen for transmission, and let $\mathcal{S}$ be the set of users that receive p (the packet is erased at the rest of the receivers). If $\mathcal{S} = \emptyset$, the packet is retransmitted. Otherwise, $\mathcal{S} \neq \emptyset$ and let $\tilde{\mathcal{S}}$ be the subset of receivers in $\mathcal{S} \cap (\cup_{k=1}^{m} \mathcal{L}_k)$ with the following property: $i \in \tilde{\mathcal{S}}$ iff it belongs to at least $m-1$ of the sets $\mathcal{L}_k$, $k \in \{1, ..., m\}$. Hence, before transmission of p, user $i \in \tilde{\mathcal{S}}$ is a *listener* for all but at most one of the packets $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ $k \in \{1, ..., m\}$.

Every receiver in $\mathcal{S}$ which is a destination of p can decode its corresponding packet i.e., for any $k \in \{1, \ldots, m\}$ and $i \in \mathcal{D}_k \cap \mathcal{S}$, packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ is delivered to destination i. This is performed as follows: the BCR implies that $i \in \mathcal{L}_r$ for all $r \neq k$. Hence, this $i \in \mathcal{D}_k \cap \mathcal{S}$ can compute $p_{\mathcal{D}_k}^{\mathcal{L}_k} = p \oplus \bigoplus_{r \neq k} p_{\mathcal{D}_r}^{\mathcal{L}_r}$, where all RHS terms are known to i after transmission of p (p is received correctly by i, since $i \in \mathcal{S}$). But $i \in \mathcal{D}_k \cap \mathcal{S}$ is also a *destination* for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$, so that it can decode the unique unknown native packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ contained in $p_{\mathcal{D}_k}^{\mathcal{L}_k}$.

If $\mathcal{D}_k - \mathcal{S} = \emptyset$, i.e. all users in $\mathcal{D}_k$ receive p, then packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ is removed from virtual queue $Q_{\mathcal{D}_k}^{\mathcal{L}_k}$. Hence, if $\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S} = \emptyset$, then all packets are delivered to their destination and are removed from the corresponding queues. Otherwise, $\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S} \neq \emptyset$ and we distinguish two cases:

1) it holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m} (\mathcal{L}_k \cup \mathcal{D}_k) = \emptyset$ (equivalently, $\mathcal{S} \subseteq \cup_{k=1}^{m} (\mathcal{L}_k \cup \mathcal{D}_k)$): in this case, $p_{\mathcal{D}_k}^{\mathcal{L}_k}$, where $\mathcal{D}_k - \mathcal{S} \neq \emptyset$, is moved to $Q_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$ and each packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$, $i \in \mathcal{D}_k - \mathcal{S}$, to $Q_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}(i)$. The consistency of these packet movements with BP is subsequently proved in Lemma 4. Note also that, according to this rule, a packet is either not moved (if $\mathcal{D}_k \cap \mathcal{S} = \emptyset$), or is moved to a higher level (or within the same level but higher sublevel) queue.

2) it holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m} (\mathcal{L}_k \cup \mathcal{D}_k) \neq \emptyset$: We further distinguish two subcases.

- If $\left|\left(\cap_{k \in \{1,...,m\}} \mathcal{L}_k \cup \mathcal{S}\right) \cup \left(\cup_{k \in \{1,...,m\}} \mathcal{D}_k - \mathcal{S}\right)\right| > \max_{k \in \{1,...,m\}} |\mathcal{L}_k \cup \mathcal{D}_k|$, then packet p is moved to $Q_{\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^{m} \mathcal{L}_k \cup \mathcal{S}}$, packets $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ are removed from $Q_{\mathcal{D}_k}^{\mathcal{L}_k}$ and for each i in $\mathcal{D}_k - \mathcal{S}$ packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ is moved from $Q_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ to $Q_{\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^{m} \mathcal{L}_k \cup \mathcal{S}}(i)$.
- If $\left|\left(\cap_{k \in \{1,...,m\}} \mathcal{L}_k \cup \mathcal{S}\right) \cup \left(\cup_{k \in \{1,...,m\}} \mathcal{D}_k - \mathcal{S}\right)\right| \leq \max_{k \in \{1,...,m\}} |\mathcal{L}_k \cup \mathcal{D}_k|$, then
  1) if $\mathcal{S} \cap \cup_{k=1}^{m} \mathcal{L}_k = \emptyset$, no further action is taken.
  2) else, set $\mathcal{S} \leftarrow \mathcal{S} \cap \cup_{k=1}^{m} \mathcal{L}_k$ and follow the rules of case 1.

The validity of the above actions is proved in the following result [7], which also justifies in retrospect the selection of the above relations between set cardinalities.

Lemma 4. *Assuming that Basic Properties are satisfied at the beginning of slot t, then the application of the Basic Coding Rule and Rules for Packet Movement to the packet transmitted at slot t satisfies Basic Properties at the beginning of slot $t+1$.*

IV. Stabilizing policy, stability outer bound and capacity achievement for $N = 4$.

Let Π be the class of policies where the transmitted packets are selected according to BCR and the feedback-based packet movements are performed according to RPM. We will design a stabilizing policy π^* within Π, find an outer bound on the stability region $\mathcal{R}_{\Pi}$ and prove that this bound is tight, for i.i.d. erasure events and $N = 4$, and is achieved by π^*.

At the beginning of each slot, a decision must be made at the base station concerning which packets from the virtual queues must be XORed to form the packet $p = p_{\mathcal{D}_1}^{\mathcal{L}_1} \oplus \ldots \oplus p_{\mathcal{D}_m}^{\mathcal{L}_m}$ to be transmitted. Such a decision is called a "control" $I_{\mathcal{D}_1,...,\mathcal{D}_m}^{\mathcal{L}_1,...,\mathcal{L}_m}$ and we denote the set of such controls by $\mathcal{I}$. As mentione, the BCR is applied for the formation of packet p. For this system, an admissible policy consists of selecting, at the beginning of each time slot, one of the available controls $I_{\mathcal{D}_1,...,\mathcal{D}_m}^{\mathcal{L}_1,...,\mathcal{L}_m}$ to form a packet p for transmission. After p is transmitted, packets are moved among the real queues $Q_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ according to the RPM.

It turns out that this system falls in the class of systems whose stability has been studied in [8]. We next summarize the main results in [8] that are necessary in the subsequent development. Consider a network with a node set $\mathcal{M} \cup \{d\}$ and edge (i.e. link) set $\mathcal{E}$, where the special node d represents the destination of traffic originated at the nodes in $\mathcal{M}$. The node set $\mathcal{M}$ is identified with the set of all real queues $Q_{\mathcal{D}}^{\mathcal{L}}(k)$, with $k \in \mathcal{D}$, i.e. each node $m \in \mathcal{M}$ corresponds to exactly one real queue. Let $\mathcal{E}_{out}^{m}$, $\mathcal{E}_{in}^{m}$ denote, respectively, the set of outgoing and incoming links to node m. We allow self-loops in the network, i.e. for node $m \in \mathcal{M}$, there may be a link (m, m).

A finite set of controls $\mathcal{I}$ is available. For each control $I \in \mathcal{I}$, "transmission" takes place over the sets of outgoing links $\mathcal{E}_{out}^{m}$ of node $m \in \mathcal{N}$ as follows. If, at a given slot, control $I \in \mathcal{I}$ is applied, then for any node $m \in \mathcal{M}$ at most $\hat{\mu}_m(I) \geq 0$ packets may be transmitted over the set $\mathcal{E}_{out}^{m}$ in the following random manner: for each $m \in \mathcal{M}$ and $I \in \mathcal{I}$, there is a random sequence $\{R_n^m(I)\}_{n=1}^{\infty}$, where each $R_n^m(I)$ takes values in the set $\mathcal{E}_{out}^{m}$, with the following interpretation. The n-th packet transmitted over the set $\mathcal{E}_{out}^{m}$, when control I is applied, is received *only* by the recipient of the link $R_n^m(I)$. Of course, if $R_n^m(I) = (m, m)$, then the packet is not received by any node in $\mathcal{E}_{out}^{m}$-$\{m\}$, hence it remains at node m.

For a given n and I, the random variables $\{R_n^m(I)\}_{m \in \mathcal{N}}$ may be arbitrarily correlated. Moreover, the random sequences $\left\{\{R_n^m(I)\}_{m \in \mathcal{M}}\right\}_{n=1}^{\infty}$ are assumed to be i.i.d and we denote $p_e^m(I) \triangleq \Pr(R_n^m(I) = e)$, $e \in \mathcal{E}_{out}^m(I)$.

For control $I \in \mathcal{I}$, we define the set $\Gamma(I)$ as $\Gamma(I) = \left\{\boldsymbol{f} = \{f_e\}_{e \in \mathcal{E}} : \; f_e = p_e^m(I)\mu_m\right\}$, where $0 \le \mu_m \le \hat{\mu}_m(I)$, $i \in \mathcal{N}$, $e \in \mathcal{E}_{out}^m$, and the convex hull $\mathcal{H}$ of the sets $\Gamma(I)$ as $\mathcal{H} = \operatorname{conv}(\Gamma(I), I \in \mathcal{I})$. The stability region of the network is described by the following Theorem [8].

Theorem 5. *The stability region $\mathcal{R}_\Pi$ of the system is the set of arrival rates $\boldsymbol{\lambda} = \{\lambda_m\}_{m \in \mathcal{M}}$, $\lambda_m \ge 0$, for which there exists a vector $\mathbf{f} \in \mathcal{H}$ such that for any node $m \in \mathcal{M}$ it holds, $\sum_{e \in \mathcal{E}_{in}^m} f_e + \lambda_m \le \sum_{e \in \mathcal{E}_{out}^m} f_e$.*

Directly applying the results in [8], we obtain the following stabilizing policy: at the beginning of each time slot, the policy chooses, according to BCR, a control of the form $I = I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m} \in \mathcal{I}$, such that all real queues in the set $\tilde{\mathcal{M}}(I) = \bigcup_{l=1}^{m} \bigcup_{k \in \mathcal{D}_l} \left\{Q_{\mathcal{D}_l}^{\mathcal{L}_l}(k)\right\}$ are nonempty, and forms the appropriate packet to be transmitted on that slot, $p = \oplus_{l=1}^{m} p_{\mathcal{D}_l}^{\mathcal{L}_i}$. If control I is chosen, one packet from each of the real queues in the set $\tilde{\mathcal{M}}(I)$ may be moved to another queue inside the network, or may reach the destination (and thus exit the network), according to RPM. No packets from any of the other queues are moved. The algorithm for choosing the appropriate control is summarized below.

Algorithm for stabilizing policy π^*: At each decision slot:

1) For each control $I = I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m} \in \mathcal{I}$ that satisfies BCR and for each $i \in \tilde{\mathcal{M}}(I)$, compute $c_i(I) = \max\left(K_i - \sum_{e=(i,j):e \in \mathcal{E}_{out}^i} p_e^i(I) K_j, 0\right)$, where K_i is the length of the queue corresponding to node i, and define $C(I) = \sum_{i \in \tilde{\mathcal{M}}(I)} c_i(I)$.
2) Select the control that maximizes the reward, i.e. $I^* = \arg\max_{I \in \mathcal{I}} C(I)$ and apply RPM for I^*.

We now derive an outer bound on $\mathcal{R}_\Pi$ by deparameterizing (i.e. eliminating the flow variables $\boldsymbol{f}$ in), Theorem 5. This bound is identical to the bound on the capacity region of the BEC with feedback presented in [1], [2].

Theorem 6. *For arbitrary N, the stability region $\mathcal{R}_\Pi$ (in bits per slot) satisfies the relation: $\mathcal{R}_\Pi \subseteq \left\{\boldsymbol{\lambda} : \; \max_{\sigma \in \mathcal{P}} \sum_{i \in \mathcal{N}} \frac{\lambda_{\sigma(i)}}{1-\epsilon_{\tilde{\sigma}(i)}} \le L\right\} \triangleq \mathcal{C}_u$, where $\mathcal{P}$ is the set of permutations σ on $\mathcal{N}$ and $\tilde{\sigma}(i) = \{\sigma(1), ..., \sigma(i)\}$.*

For the case of i.i.d erasure events and $N = 4$, we derive the following stronger and unexpected result [7].

Theorem 7. *For the case of 4 users and i.i.d erasure events, the upper bound of Theorem 6 is tight (i.e. $\mathcal{R}_\Pi = \mathcal{C}_u$) and the previously described policy $\pi^* \in \Pi$ is stabilizing.*

A final note is in place: although the recent work of [9] has shown that the information capacity region and system stability region for the erasure broadcast system with feedback are identical, we cannot prove Theorem 6 via [9] by claiming that $\mathcal{C}_u$ is the capacity region of the system under consideration. The reason is that in [1] and [2] it was shown that $\mathcal{C}_u$ is a capacity outer bound for a system where idle slots are not allowed, while both in the current system and in the system in [9] we explicitly allow the transmitter to transmit no packet during a slot ("idle slot"). In information-theoretic terms, the transmitter has an additional input symbol, corresponding to an idle slot, so that this "extended" BEC can have a capacity region that is a strict superset of $\mathcal{C}_u$.

However, it is proved in [7] that a capacity outer bound $\mathcal{C}_{ext}^{out}$ for the "extended" BEC differs from $\mathcal{C}_u$ by less than 1 bit and in fact the difference decreases to zero as $L \to \infty$, i.e. the outer bound to $\mathcal{R}_\Pi$ and the true capacity region of the system are almost identical. Hence, combining Theorem 7 and [9], we conclude that the stabilizing policy π^* is asymptotically optimal as $L \to \infty$ for the case $N = 4$ and i.i.d erasures.

V. Conclusions

We presented a network coding scheme for the broadcast erasure channel with N unicast sessions, which employs only XOR operations and guarantees instant decodability. Using this scheme, we proposed a provably stabilizing policy within a specific class of policies for arbitrary N. For the special case of 4 users and i.i.d. erasure events, we proved that any rate within the BEC capacity outer bound is stable under the proposed policy, therefore the stability region of the system is identical to the capacity outer bound of the BEC channel with feedback and differs from the extended (i.e. considering idle slots as information symbols) BEC capacity outer bound by less than 1 bit per slot. Future work could be aimed towards the generalization of the above to more than 4 users or the development of suboptimal variations of the proposed policy requiring less packet overhead. In the current work we did not consider overhead issues regarding packet addresses. Also, while the available controls are manageable for small N, they rapidly increase as N increases. The design of efficient suboptimal policies that take into account overhead and restrict the number of available controls is another direction of research.

References

[1] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Multiuser broadcast erasure channel with feedback — capacity and algorithms," in *Proc. 4th Workshop on Network Control and Optimization (NetCoop)*, December 2010, pp. 74–81. [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, published in IEEE Trans. Inform. Theory, vol. 58, no. 2, Feb. 2012.

[3] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "XOR-based coding for the 3-user broadcast erasure channel with feedback," in *Proc. WiOpt, 8th International RAWNET Workshop*, May 2012.

[4] 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.

[5] Y. Sagduyu and A. Ephremides, "On broadcast stability of queue-based dynamic network coding over erasure channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5463–5478, December 2009.

[6] D. Traskov, M. Médard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. IEEE GLOBECOM*, December 2009.

[7] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Stable XOR-based policies for the broadcast erasure channel with feedback." [Online]. Available: http://arxiv.org/abs/1211.5358

[8] G. Paschos, L. Georgiadis, and L. Tassiulas, "Scheduling with pairwise XORing of packets under statistical overhearing information and feedback," in *Proc. WiOpt, 7th International Workshop on Resource Allocation and Cooperation in Wireless Networks (RAWNET)*, May 2011.

[9] L. Georgiadis, G. Paschos, L. Tassiulas, and L. Libman, "Stability and capacity through evacuation times," in *Proc. Information Theory Workshop (ITW)*, September 2012.