Online Broadcasting with Network Coding

Lorenzo Keller, Eleni Drinea, Christina Fragouli
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL)
E-mail: {lorenzo.keller, eleni.drinea, christina.fragouli}@epfl.ch

***Abstract*—Consider a source broadcasting M packets to N receivers over independent erasure channels, where perfect feedback is available from the receivers to the source, and the source is allowed to use coding. We investigate offline and online algorithms that optimize delay, both through theoretical analysis as well as simulation results.**

I. Introduction

By broadcasting we refer to the case where a common source has a number of packets to distribute to a set of destinations. A packet transmitted by the source is observed by each destination through an independent erasure channel. This model can be used in a variety of applications: in cellular networks, packet losses due to fading are experienced as packet erasures at higher levels; in overlay networks, where a source multicasts to a number of receivers along a tree, packet losses due to congestion can also be modeled through erasure channels. Moreover, understanding this simple model offers a first step towards variations, where for example receivers demand subsets of the message set the source has.

Broadcasting has been studied in the literature, mainly along the following three directions:
1) Work in scheduling examines offline and online algorithms, where feedback is available from the destinations to the source, but the source does not employ coding. Performance criteria include achievable rate and delay, see for example [1].
2) For applications such as satellite communications, where use of feedback might be impractical, elegant forward error correction schemes have been proposed, such as Raptor codes [2] and LT codes [3]. Under the constraint of not using feedback, these codes achieve rate optimality, and minimize encoding and decoding complexity.
3) Theoretical work has examined use of feedback to achieve the optimal rate and zero probability of error [4] over broadcast erasure channels.

Our work focuses on using feedback and coding over a broadcast erasure channel, towards the goal of optimizing the delay the receivers experience. Use of coding across information packets at the transmitter may increase the delay, as a receiver may need to collect several coded packets before being able to decode. On the other hand, use of coding may allow to achieve higher information rates, thus reducing the delay. Thus rate and delay are clearly intertwined quantities.

We define a scheme to be rate-optimal if every successfully received packet brings new information to its receiver. We say that a scheme has zero delay, if it is rate optimal, and additionally, each receiver can instantaneously decode an information packet from each received transmission. We formally define our notion of delay in Section II.

In the network coding literature, work has looked at use of acknowledgments and coding to optimize the achievable rate, under the condition that each received packet is either useless or can be immediately decoded by the destination [5]. Such schemes, although simple and elegant to implement, do not offer rate or delay optimality guarantees. Another line of work has looked into use of coding and acknowledgments in order to minimize the queue size at a transmitter [6]. This performance metric is quite different from optimizing for delay.

In this paper, we are interested in minimizing the delay, under the constraint that the employed scheme is rate-optimal. In Section II, we formally introduce our model and notation. In Section III we present some theoretical results pertaining to offline algorithms. Offline algorithms assume perfect knowledge of the future erasure patterns, thus offering a performance benchmark. In Section IV we discuss a number of online algorithms, and compare their performance through simulation results. Finally, Section V concludes the paper and discusses future research directions.

II. Model, Definitions and Basic Properties

A. Model

Consider a source broadcasting M packets to N receivers. The source receives perfect feedback from each receiver and may transmit coded packets. We assume that time is slotted. At each time slot the source transmits one packet. Each receiver R_i successfully receives the packet with probability $1 - p_i$, independently from every other receiver.

We can think of the M information packets as defining an M-dimensional space U over a finite field $\mathbb{F}_q$, where each information packet corresponds to one of the orthonormal basis vectors $\{e_1, e_2, \ldots, e_M\}$. During the network operation each receiver R_i collects a subspace Π_i of the M-dimensional space. A received vector brings *innovative information* to a receiver R_i if it increases the dimension of Π_i by one.

Definition 1: A scheme is rate-optimal, if every successfully received packet brings innovative information to its receiver.
In a rate-optimal scheme, a receiver R_i that has received t packets, has collected a t-dimensional subspace Π_i of the M-dimensional space. For $t = M$, the receiver can successfully decode all information packets, and is no longer interested in the packet transmissions.

B. Some Useful Properties

Before proceeding with our definition of delay, we present without proof some basic lemmas that we use in our paper development. We will use the notation $\Pi_1 \cup \Pi_2$ and $\Pi_1 \cap \Pi_2$ to denote the common span and the intersection of two subspaces Π_1 and Π_2, respectively.

Lemma 1: A node R_i can decode the information packet e_j if and only if $e_j \in \Pi_i$.

Lemma 2: Let Π_1 denote the subspace spanned by the vectors $< p_1 \dots p_{t_1} >$ receiver R_j has collected up to time t_1, and Π_2 the subspace spanned by the vectors $< p_{t_1+1} \dots p_{t_2} >$ receiver R_j collects between times $t_1 + 1$ and t_2. In a rate-optimal scheme, if $e_i \in \Pi_1$, then $e_i \notin \Pi_2$, for all i, $t_1 < M$ and $t_2 \leq M$.

Corollary 1: Consider the subspaces Π_1 and Π_2 spanned by the vectors $< e_1 \dots e_{n/2} >$ and $< e_{n/2+1} \dots e_n >$ respectively, and create the subspace Π_3 spanned by the vectors $< e_1 + e_{n/2+1} \dots e_{n/2} + e_n >$. Then $\Pi_1 \cap \Pi_3 = \emptyset$ and $\Pi_2 \cap \Pi_3 = \emptyset$.

C. Our Measures of Delay

Definition 2: The delay $\mathcal{D}_i$ that a receiver R_i experiences is defined as the number of packets that are successfully received but do not allow to immediately decode an information packet. Note that in a scheme that is not rate-optimal, a successfully received packet that does not bring innovative information to the receiver, will by definition incur a delay of one.

For our theoretical analysis, we will be interested in both the *total delay* defined as

$$\mathcal{D}_t \triangleq \sum_{i=1}^{N} \mathcal{D}_i,$$

and the *worst case delay* experienced across all receivers, given by

$$\mathcal{D}_w \triangleq \max_i \mathcal{D}_i.$$

We will also examine through simulation results the *median delay* defined as

$$\mathcal{D}_m \triangleq \text{median}_i \ \mathcal{D}_i.$$

Different schemes may result in different values for $\mathcal{D}_t$, ($\mathcal{D}_w$); we will say that a particular value of $\mathcal{D}_t$ ($\mathcal{D}_w$) is *optimal for a given erasure pattern*, if this is the minimum value achievable by all possible rate-optimal transmission schemes.

Note that our definition of delay does not take into account:

- The order of the decoded packets; re-ordering the packets does not incur a penalty.
- The time between an information packet is successfully received for the first time (even as part of a coded packet) and the time it is decoded.

III. Offline Algorithms

Offline algorithms assume that we have perfect knowledge of the future, and thus, the source can encode the packets taking into account the erasure patterns that will occur at each future time-slot. This is in contrast to online algorithms, were we assume perfect knowledge of the past erasure patterns, but we need to decide on the current transmission with no knowledge regarding the erasure pattern this transmission and future transmissions will be subject to. We first describe an optimal offline algorithm when $N = 3$ that results in $\mathcal{D}_w = 0$ (and hence $\mathcal{D}_t = 0$). We then show that when $N = 4$, there exist erasure patterns where $\mathcal{D}_w$, $\mathcal{D}_t$ are bounded away from zero, and in fact can be as large as $\mathcal{O}(M)$.

A. The case of $N = 2$ and $N = 3$ receivers

For the case of $N = 2$ receivers, we can find an optimal offline algorithm with $\mathcal{D}_w = 0$ [5], [7]. We will here prove that the same is true for $N = 3$ as well.

Theorem 1: There exists a polynomial time offline algorithm that can find an optimal schedule achieving $\mathcal{D}_w = 0$ for $N = 3$. This algorithm needs to know only one future erasure pattern, and uses binary operations.

Proof: Let $p(t)$ denote the packet the source sends at time t. Let $B_S(t)$ be the set of information packets that the source has used to build the coded packets up to time $t-1$, and let $S_i(t) \subseteq B_S(t)$ be the subset of the $B_S(t)$ information packets that node R_i has *not* decoded, before the transmission of packet $p(t)$. Moreover, define $S_{ij}(t) \triangleq S_i(t) \cap S_j(t)$. For simplicity of notation, we may omit the time index t when there is no confusion. At $t = 1$, before transmissions begin, $B_S = S_1 = S_2 = S_3 = \emptyset$. We will create the packet $p(t)$ to be transmitted at time t as follows. We distinguish three cases:

1) Assume that only one node R_i receives $p(t)$. Then

   a) If $S_i(t) = \emptyset$ (is empty), then we select $p(t) = p_{new} \notin B_S(t)$. We update $B_S(t+1) = B_S(t) \cup \{p_{new}\}$, $S_j(t+1) = S_j(t) \cup \{p_{new}\}$ for $j \neq i$, and $S_{jk}(t+1) = S_{jk}(t) \cup \{p_{new}\}$ for $j, k \neq i$.

   b) Otherwise, transmit $p(t) = p \in S_i(t)$, and update $S_i(t+1) = S_i(t) \backslash \{p\}$. Also, if $p \in Sij(t)$, update $Sij(t+1) = S_{ij} \setminus \{p\}$.

2) Assume two nodes R_i and R_j receive $p(t)$. Then

   a) If $S_i(t) = \emptyset$ or $S_j(t) = \emptyset$, transmit $p(t) = p_{new} \notin B_S(t)$, and update $B_S(t+1) = B_S(t) \cup \{p_{new}\}$, $S_k(t+1) = S_k(t) \cup \{p_{new}\}$ for $k \neq i, j$.

   b) Otherwise, if $S_{ij}(t) \neq \emptyset$, transmit packet $p \in S_{ij}$, while if $S_{ij}(t) = \emptyset$ transmit $p(t) = p_i + p_j$ with $p_i \in S_i$ and $p_j \in S_j$.

3) Assume all three nodes successfully receive the transmitted packet. We will then send $p(t) = p_{new} \notin B_S(t)$ (in fact, sending a new packet is the only choice, because this algorithm belongs to the class of algorithms described by Proposition 1 at the end of the next subsection, and thus there exists at least one receiver R_i with $S_i = \emptyset$.)

Once one of the three receivers has received all M packets, we can use the optimal algorithm for $N = 2$ receivers. Note that the most computationally intensive task at each step of the described algorithm, is the search of whether $p \in S_{ij}(t)$. Since $S_{ij}(t)$ has size at most M, the search can be performed in $\mathcal{O}(M \log M)$ operations. ∎

B. The general case

We will now consider the general case with $N > 3$ receivers. Note that the worst case delay we may have is $\mathcal{O}(M)$, since, encoding for example the transmitted packets using an MDS code allows to achieve this delay. Our next results show that, even for $N = 4$, there exists an erasure pattern such that this worst case bound is achieved for any even $M \geq 4$.

We start by introducing some notation. Let $M = 2k$, for positive integer $k \geq 2$. For $1 \leq j \leq 4$, let π_j be the time period corresponding to transmissions $(j-1) \cdot M/2 + 1$ up to $j \cdot M/2$. Each receiver receives every packet sent during the following periods and receives nothing during the other periods: R_1 receives in π_1 and π_2, R_2 in π_1 and π_3, R_3 in π_2 and π_3, and R_4 in π_3 and π_4. The resulting erasure pattern is depicted in Table I for $M = 4$; we shall henceforth call this pattern E, where the dependence on M is implied. Since we only consider rate-optimal schemes, no receiver requires further transmissions after time $2M$ since each has already collected M packets by then. The following lemma shows

Time slot	R_1	R_2	R_3	R_4
1	–	–	x	x
2	–	–	x	x
3	–	x	–	x
4	–	x	–	x
5	x	–	–	–
6	x	–	–	–
7	x	x	x	–
8	x	x	x	–

TABLE I
ERASURE PATTERN FOR $N = 4$, AND $M = 4$, WHERE "X" DENOTES ERASURE, AND "–" SUCCESSFUL RECEPTION.

that the total delay $\mathcal{D}_t$ is $\mathcal{O}(M)$ for the erasure pattern E.

Lemma 3: Consider $N = 4$ receivers that want to receive the universe U of $M = 2k$ packets. For the erasure pattern E and any rate-optimal transmission scheme

$$\mathcal{D}_t \geq M/2. \tag{1}$$

A transmission scheme that achieves the equality above is depicted in Table II for $M = 4$ (generalizing for $M = 2k$ is straightforward).

Proof: Let $\Pi_1, \ldots, \Pi_4$ be the subspaces the source transmits during $\pi_1, \ldots, \pi_4$. Also, let $\alpha_1, \alpha_2 \geq 0$ be the delays R_1 experiences during π_1, π_2 respectively, $\beta_1, \beta_2 \geq 0$ the delays R_2 experiences during π_1, π_3 respectively, $\gamma_1, \gamma_2 \geq 0$ the delays R_3 experiences during π_2, π_3 respectively, and $\delta_1, \delta_2 \geq 0$ the delays R_4 experiences during π_3, π_4 respectively. Then $\alpha_1 = \beta_1$ since R_1 and R_2 both receive Π_1 during π_1 and this is the first period they receive any packets. However, it could be $\gamma_1 > 0$ while $\alpha_2 = 0$ since R_1 has already received some packets in π_1 hence might be able to decode packets from Π_2 that R_3 can not.

From rate-optimality it holds that:

(i) $\Pi_1 \cup \Pi_2 = U$, $\Pi_1 \cup \Pi_3 = U$, and $\Pi_2 \cup \Pi_3 = U$, and

(ii) $\Pi_i \cap \Pi_j = 0$.

Let $E_k \triangleq \{e_i\} \subseteq \Pi_k$ be the set of $\{e_i\}$ vectors contained in Π_k, $k = 1, \ldots, 3$. Then:

(iii) $E_i \cap E_j = 0$ from (ii),

(iv) $\alpha_1 = \beta_1 \geq \frac{M}{2} - |E_1|$, $\gamma_1 \geq \frac{M}{2} - |E_2|$, $\delta_1 \geq \frac{M}{2} - |E_3|$ (because R_1, R_2 observe only Π_1, R_3 only Π_2 and R_4 only Π_3), and

(v) $|E_1| + |E_2| + |E_3| \leq M$.

Since $\mathcal{D}_t = \alpha_1 + \beta_1 + \alpha_2 + \gamma_1 + \beta_2 + \gamma_2 + \delta_1 + \delta_2$, we have

$$\begin{aligned} \mathcal{D}_t &\geq \alpha_1 + \beta_1 + \gamma_1 + \delta_1 \\ &\geq 2(\frac{M}{2} - |E_1|) + (\frac{M}{2} - |E_2|) + (\frac{M}{2} - |E_3|) \\ &= 2M - |E_1| - (|E_1| + |E_2| + |E_3|) \\ &\geq \frac{M}{2}. \end{aligned} \tag{2}$$

where the first inequality follows from $\alpha_2, \beta_2, \gamma_2, \delta_2 \geq 0$, the second from (iv), and the last from $|E_1| \leq M/2$ and (v). ■

Equation (2) further yields the following lower bound for the total delay of R_2, R_3 and R_4.

Corollary 2: The total delay of R_2, R_3 and R_4 is at least $M/2$ under *any* rate-optimal transmission scheme for E.

We conclude that for the transmissions schemes that achieve equality for (1), it must hold that $\alpha_1 = 0$. This implies $\beta_1 = 0$ as well; since achieving the lower bound in (1) also requires $\alpha_2 = \beta_2 = \gamma_2 = \delta_2 = 0$, we obtain the following corollary.

Corollary 3: Any rate-optimal transmission scheme that optimizes $\mathcal{D}_t$ for the erasure pattern E satisfies

$$\gamma_1 + \delta_1 = M/2.$$

Intuitively this corollary states that delays introduced in π_1 are more costly because they delay two receivers (R_1 and R_2).

Corollary 2 refines the simple lower bound of $M/8$ for $\mathcal{D}_w$ provided by Lemma 3 to $\lceil M/6 \rceil$. It is now clear from Corollary 3 that for the transmission schemes that achieve the optimal $\mathcal{D}_t$, $\mathcal{D}_w$ exceeds the lower bound of $\lceil M/6 \rceil$. However, Table III depicts a transmission scheme that achieves $D_w = \lceil M/6 \rceil$ for $M = 8$ (generalizing for any $M = 2k$ is straightforward). Hence we obtain the following corollary.

Corollary 4: Consider $N = 4$ receivers that want to receive the universe U of $M = 2k$ packets. For the erasure pattern E and any rate-optimal transmission scheme

$$\mathcal{D}_w \geq \lceil M/6 \rceil. \tag{3}$$

By Corollary 2, the schemes that achieve the optimal $\mathcal{D}_w$ satisfy the following property.

Corollary 5: Any rate-optimal transmission scheme that optimizes $\mathcal{D}_w$ for the erasure pattern E satisfies

$$\lfloor M/6 \rfloor + \lceil M/6 \rceil \leq \beta_1 + \gamma_1 \leq 2 \cdot \lceil M/6 \rceil.$$

Observe that the total delay of any of the above schemes exceeds the total delay of any of the schemes described by Corollary 3 by at least $\lfloor M/6 \rfloor$. This implies that depending on the measure of delay we wish to minimize, different strategies should be considered.

Besides its interest for the study of offline algorithms, the erasure pattern E is also worth studying because it can model a dynamic scenario where new receivers join the system at different times and we do not want to penalize them with very long delays.

Time slot	R_1	R_2	R_3	R_4
1	e_1	e_1	x	x
2	e_2	e_2	x	x
3	e_3	x	e_3	x
4	e_4	x	e_4	x
5	x	$e_1 \oplus e_3$	$e_1 \oplus e_3$	$e_1 \oplus e_3$
6	x	$e_2 \oplus e_4$	$e_2 \oplus e_4$	$e_2 \oplus e_4$
7	x	x	x	e_1
8	x	x	x	e_2

TABLE II
OPTIMAL SCHEDULE FOR D_t FOR ERASURE PATTERN IN TABLE I.

Time slot	R_1	R_2	R_3	R_4
1	$e_1 \oplus e_2$	$e_1 \oplus e_2$	x	x
2	e_3	e_3	x	x
3	e_4	e_4	x	x
4	e_5	e_5	x	x
5	$e_3 \oplus e_6$	x	$e_3 \oplus e_6$	x
6	$e_4 \oplus e_7$	x	$e_4 \oplus e_7$	x
7	e_1	x	e_1	x
8	e_8	x	e_8	x
9	x	e_2	e_2	e_2
10	x	e_6	e_6	e_6
11	x	e_7	e_7	e_7
12	x	$e_5 \oplus e_8$	$e_5 \oplus e_8$	$e_5 \oplus e_8$
13	x	x	x	e_5
14	x	x	x	e_4
15	x	x	x	e_3
16	x	x	x	e_1

TABLE III
OPTIMAL SCHEDULE FOR D_w FOR ERASURE PATTERN E AND $M = 8$.

It is worth mentioning here that the algorithm described in the proof of Theorem 1 does not readily generalize to the optimal offline solution for $N > 3$ (e.g., see Tables IV and V). It is interesting to understand why this algorithm succeeds for $N = 3$ but fails for $N = 4$. The "problem" occurs when three receivers successfully receive the transmitted packet. In this case, we can always find a packet such that at least two receivers can decode, but we cannot always find a packet so that all three receivers decode (e.g., see Table I).

Time slot	Packet sent	R_1	R_2	R_3	R_4
1	e_1	e_1	e_1	x	e_1
2	e_2	e_1,e_2	x	x	e_1,e_2
3	e_3	x	e_1,e_3	x	e_1,e_2,e_3
4	e_1	x	x	e_1	x
5	$e_2 \oplus e_3$	e_1,e_2,e_3	e_1,e_2,e_3	e_1,$e_2 \oplus e_3$	x
6	e_2	x	x	e_1,e_2, e_3	x

TABLE IV
SCHEDULE SELECTED BY GREEDY (DELAY 1).

We conclude this section with a proposition that shows that for any N, we can have a receiver that experiences zero delay.

Time slot	Packet sent	R_1	R_2	R_3	R_4
1	e_1	e_1	e_1	x	e_1
2	e_2	e_1,e_2	x	x	e_1,e_2
3	e_3	x	e_1,e_3	x	e_1,e_2,e_3
4	e_2	x	x	e_2	x
5	$e_2 \oplus e_3$	e_1,e_2,e_3	e_1,e_2,e_3	e_2, e_3	x
6	e_1	x	x	e_1,e_2, e_3	x

TABLE V
OPTIMAL SCHEDULE (DELAY 0).

Proposition 1: There exists a rate-optimal offline algorithm where at least one receiver has received all the information the source has transmitted.

Proof: Let $p(t)$ denote the packet the source sends at time t. Let $\Pi_S(t)$ be the subspace spanned by the packets that the source S has transmitted, and $\Pi_i(t) \subseteq \Pi_S(t)$ the subspace received by destination R_i, before the transmission at time t. We will prove that, provided $\dim(\Pi_S(t)) < M$, there exists at least one receiver such that $\Pi_S(t) = \Pi_i(t)$. In other words, the set $R^*(t) = \{R_i \in \text{receivers} \,|\Pi_i(t) = \Pi_S(t)\}$ is not empty. Our proof uses induction. For $t = 1$, $\Pi_S(t) = \Pi_i(t) = \{0\}$, and the condition is satisfied. Assume the condition holds at time t. Then we distinguish two cases:

- There exists a receiver $R_i \in R^*(t)$ that receives $p(t)$. Clearly a rate-optimal code will then need to send a packet $p(t) \notin \Pi_S(t)$. We will then have that

$$\Pi_S(t+1) = \Pi_S(t) \cup p(t) = \Pi_i(t) \cup p(t) = \Pi_i(t+1)$$

and R_i will still belong in $R^*(t+1)$.

- There does not exist a receiver $R_i \in R^*(t)$ that receives $p(t)$. Then, we can select a packet $p(t) \in \Pi_S(t)$ that brings innovative information to all receivers. Note that $\Pi_S(t) = \Pi_S(t+1)$, and as a result, $R^*(t) \subseteq R^*(t+1)$. ■

IV. ONLINE ALGORITHMS

A. Greedy Algorithm

For $N = 2$ receivers, the greedy online algorithm achieves zero delay [5], [7]. This is not the case for $N = 3$.

Theorem 2: The greedy online algorithm with $N = 3$ has a worst case performance of at least $\frac{M}{2}$.

Proof: Consider an erasure pattern in which the first $\frac{M}{2}$ packets are received only by R_1, the next $\frac{M}{2}$ packets are received only by R_2 and the following $\frac{M}{2}$ packets are received by all three nodes. After M packets have been sent, R_1 and R_2 have decoded exactly two disjoint sets of $\frac{M}{2}$ blocks and R_3 has still not received any packet. According to Lemma 2 no blocks already decoded by some receiver can be decoded by R_3. Therefore he will not decode until $\frac{M}{2}$ packets have been sent to R_1 and R_2. The delay for R_3 is therefore $\frac{M}{2}$. ■

B. Proposed Algorithms

In this section we discuss some simple online algorithms that we implemented and tested their performance via simulations. More sophisticated algorithms which could be closer to the optimal are proving computationally challenging but this is a line of research we are currently pursuing.

The first three algorithms are used as benchmarks. The fourth algorithm optimizes the decoding delay, while the last one optimizes the delay due to reception of useless packets while also attempting to keep the decoding delay low when possible. Since we define the delay at a receiver to be the sum of these two kinds of delays, our findings (which we discuss in detail in the next section) are perhaps not surprising: none of these two algorithms performs consistently better for all ranges of parameters of our setup, though the last one is better for a wider range of parameters.

⋆ *Random network coding algorithm:* This algorithm uses the standard network coding approach: at each time step it sends a random linear combination of all packets from U. When the underlying field is large enough, any M random vectors are linearly independent w.h.p., hence this transmission scheme is rate-optimal w.h.p. Feedback information can be used to ensure rate optimality with smaller fields. The expected delay of each receiver is approximately $M-1$: w.h.p. no packets can be decoded until the last packet is received.

⋆ *Systematic random network coding algorithm:* This variant of the random network coding algorithm first sends all M packets from U unencoded. Thus each receiver receives an average of $(1-p)M$ unencoded packets, hence only needs to delay (i.e., is not able to decode) during approximately $pM-1$ time slots.

⋆ *Simple repetition algorithm:* This algorithm uses an approach similar to ARQ (Automatic Repeat reQuest) and operates in rounds. At the beginning of each round, the algorithm uses the feedback information to compute a set of packets Q from U that have not yet been received by at least one node; these packets are ordered according to some criterion (in our simulations, we just send a random permutation of the packets) and are then sent uncoded during the round. The algorithm ends when the computed Q is empty at the beginning of some round. This algorithm is not rate optimal for $N > 1$.

⋆ *Opportunistic algorithm:* This algorithm uses an opportunistic approach to improve the performance of the previous algorithm. Like COPE [5] it sends packets that can be immediately decoded by all receivers. Thus it never incurs decoding delays. However, it is not necessarily rate-optimal since it might not always be possible to select a packet that is both innovative for all receivers and instantly decodable. Hence it might incur delays due to reception of useless packets.

The algorithm works in rounds. Like the simple repetition algorithm, at the beginning of each round it builds the set Q of packets from U that have not yet been received by at least one node. The algorithm then chooses each packet c to be sent as follows. First, it removes an element e_q from Q at random and sets $c = e_q$. Then it goes (in some order) over every packet e_r that is still needed by some node. In particular, e_r might be a packet that was already transmitted and removed from Q during the round but one of the nodes that needed it experienced an erasure during the transmission. The algorithm sets $c = c \oplus e_r$ if $c \oplus e_r$ is decodable by all nodes that can decode c. Each time a packet is added to c the algorithm removes it from Q if it is present.

⋆ *Rate-optimal algorithm:* This algorithm mimicks the opportunistic algorithm but makes sure that each packet it sends is innovative for all receivers: after selecting c as above, it adds more packets from U to c, multiplied by coefficients from an appropriately large field, so that the final transmitted packet is innovative for every receiver.

We should mention here that the above algorithms impose very different decoding complexities on the receivers. In particular, the opportunistic algorithm imposes low complexity as the packets can be immediately decoded at the receivers, hence it does not involve solving systems of equations. On the other hand, the rate-optimal algorithm could be more computationally challenging but no worse than the systematic random network coding algorithm.

Finally, it should be clear that using different heuristics to select the packets sent at each round for the last three algorithms, might improve their performance. For instance instead of sending packets at random from the set of those that are required, we could favor packets that have high demand or receivers that experienced long delays. We are currently experimenting with such variations; preliminary results show that they do yield improvement, especially for $N/M < 1$.

C. Simulation Results

For our simulations, we assumed that the erasure channels between the source and each receiver are independend but all present the same erasure probability p. We discuss the performance of the algorithms of the previous section as a function of M, N and p. Our graphs show the median delay across all receivers. More specifically, to generate each graph, we run our algorithms many times. The outcome of each run is the median of the delays of the N receivers. Our plots show the median of these medians.

Since the delay is a function of three parameters, in our plots we vary one parameter and keep the other two constant. We then discuss what happens when we vary the parameters that we kept constant. Also, we do not simulate the two random network coding algorithms; instead we use the theoretical approximations for their performance.

a) Relationship between delay and M: Tests over a wide range of N and p show that the performance of all proposed algorithms for fixed N and p is linear in the number of packets M (see [8] for these results). This indicates that the delay introduced by a block is constant given N and p.

b) Relationship between delay and N: Figure 1 shows the performance of the proposed algorithms as a function of N when $p = 0.1$. As discussed, the delay of the two variants of random network coding is constant. The simple algorithm is optimal for one receiver but performs almost as bad as the systematic random network coding for two receivers. Its performance soon degrades below that of the random network coding algorithm and reaches a limit as N becomes large compared to M. We suspect that this happens because it is very likely that each packet from U will be required by at least one receiver at each round hence all blocks must be retransmitted at every round.

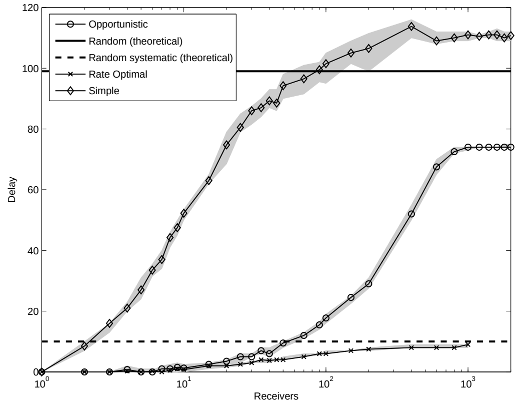

Fig. 1. Median delay as a function of N (in log scale) at $p = 0.1$ and $M = 100$. Ten experiments per point for the opportunistic and the rate optimal algorithms, 22 experiments per point for the simple algorithm. The gray area is the 95% confidence interval of the median.

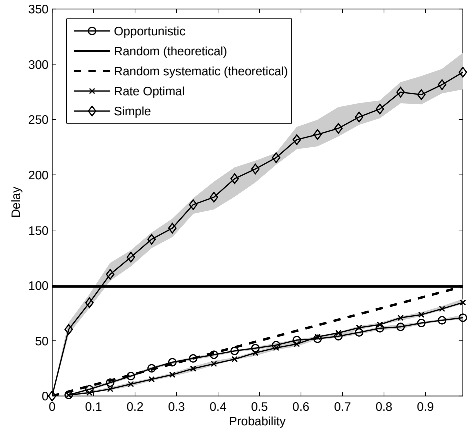

Fig. 2. Median delay as a function of p ($N = 40$, $M = 100$), 30 experiments per point, the gray area is the 95% confidence interval of the median.

The opportunistic algorithm achieves a significant improvement over the simple repetition algorithm and exhibits very small delay when N is very small. The delay again reaches a limit as N becomes large compared to M. For $p = 0.1$ as shown in Figure 1, the limiting delay is lower than that of the random network coding algorithm; however it increases with p and at $p = 0.5$, it already exceeds $M - 1$.

The rate-optimal algorithm behaves similarly to the opportunistic when N is small. In this case it outperforms the systematic network coding algorithm because it uses innovative yet easily decodable packets (linear combinations of few packets from U). However, for N large relative to M the algorithm converges to the behavior of the systematic random network coding, since it is then very likely that all packets from U have to be included to create an innovative packet.

Varying p does not change the shape of the delay function for any algorithm though it does change their relative performances. In particular, the delay of the rate-optimal algorithm always converges to $pM - 1$. We found that for $p = 0.5$ and N taking values around 12 the performance of this algorithm is worse than that of the opportunistic. Hence it is not the case that one of these two algorithms always outperforms the other.

Our simulations show that as N/M grows, the proposed feedback algorithms are not the best choice: although the rate-optimal performs the best, the complexity of keeping track of the state of the receivers is too high. Systematic network coding should be preferred since it performs comparably and does not use feedback. On the contrary, for $N/M < 1$, our proposed feedback algorithms constitute a viable solution.

c) Relationship between delay and p: The performance of the algorithms as a function of p when $N = 40$ is shown in Figure 2. For smaller N the opportunistic and the rate optimal algorithms tend to exhibit the same delay. For larger N the systematic and the rate-optimal approach have similar performances. For every N, all algorithms except random network coding have a delay that goes to zero when p goes to zero. The simple algorithm has a very bad performance even with small p. The rate-optimal algorithm is not the best choice for every p though it is better than the systematic network coding approach, for all p and N.

V. Conclusions and Discussion

We studied the problem of broadcasting M packets to N receivers over independent erasure channels with perfect feedback and source coding using rate-optimal transmission schemes. We presented the optimal offline algorithm in the case of $N = 3$ receivers that achieves zero delay. In contrast, we showed that for $N = 4$, there exists an erasure pattern that imposes a total delay of at least $M/2$ to the receivers under any rate-optimal transmission scheme. We also investigated simple online algorithms, and compared their performance to standard algorithms via simulations.

There are many interesting directions that we intend to extend this work. Open theoretical questions are finding the optimal offline algorithm for the general case of N receivers (or proving that this problem is NP-hard) and computing the competitive ratio. Other directions include design and analysis of more sophisticated online algorithms, and further comparison of rate-optimal with non-rate optimal transmission schemes in order to understand the interplay between delay due to non-innovative packets and decoding delay.

References

[1] W. Mao, "Competitive analysis of on-line algorithms for on-demand data broadcast scheduling", ISPAN' 00.

[2] A. Shokrollahi, "Raptor Codes", IEEE Trans. Inf. Theory, vol. 52, 2006.

[3] M. Luby, "LT codes", IEEE Symposium on the Foundations of Computer Science (STOC), pp. 271-280, 2002.

[4] J. Massey and P. Massey, "Zero error with feedback", online tutorial.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding", Sigcomm 2006.

[6] J. Sundararajan, D. Shah and M. Médard, "On queueing in coded networks –queue size follows degrees of freedom", ITW 2007.

[7] M. Durvy, C. Fragouli and P. Thiran, "On feedback for network coding", ISIT 2007.

[8] L. Keller, MS thesis, available online at *arni.epfl.ch* starting Oct. 1 2007.