Delay with Network Coding and Feedback

Eleni Drinea, Christina Fragouli, Lorenzo Keller
School of Computer and Communication Sciences, EPFL
Email: {eleni.drinea, christina.fragouli, lorenzo.keller}@epfl.ch

***Abstract*—We consider the problem of minimizing delay when broadcasting over erasure channels with feedback. A sender wishes to communicate the same set of μ messages to several receivers over separate erasure channels. The sender can broadcast a single message or a combination (encoding) of messages at each timestep. Receivers provide feedback as to whether the transmission was received. If at some time step a receiver cannot identify a new message, delay is incurred. Our notion of delay is motivated by real-time applications that request progressively refined input, such as the successive refinement of an image encoded using multiple description coding.**

Our setup is novel because it combines coding techniques with feedback information to the end of minimizing delay. It allows $\Theta(\mu)$ benefits as compared to previous approaches for offline algorithms, while feedback allows online algorithms to achieve smaller delay than online algorithms without feedback. Our main complexity results are that the offline minimization problem is NP-hard when the sender only schedules single messages and that the general problem remains NP-hard even when coding is allowed. However we show that coding does offer delay and complexity gains over scheduling. We also discuss online heuristics and evaluate their performance through simulations.

I. Introduction

Current and emerging applications, such as satellite imaging, roadside to vehicle communication, internet tv, wireless downlink broadcasting, require content to be downloaded quickly and reliably from a host over possibly unknown channels. In practical networks, transmissions are subject to errors: packets get dropped due to congested links, wireless fading and interference, expired timestamps, etc. Such losses are perceived as packet erasures at higher layers, and are often modeled using independent erasure channels.

To cope with unknown channels, feedback information is often available at the broadcasting source. Thus the source, when deciding what to transmit next, knows which receivers successfully received each of its past transmissions. Feedback can be efficiently employed in a wireless environment: the source might acquire such information by taking advantage of the symmetry of wireless links, or by collecting acknowledgment packets explicitly using specifically designed control traffic [7], or implicitly, by overhearing transmissions from the receiver nodes [11].

In this paper, we consider the problem of combining coding techniques and feedback information over broadcasting channels to offer reliable content delivery under delay guarantees. Our notion of delay is motivated from real-time applications with progressively refined input. Such a paradigm is provided by multiple description coding that we adopt as our illustrating example in the following; however, our notion of delay is relevant to a much more general class of applications.

Multiple description is a well studied data compression technique which allows for robust and graceful recovery in the presence of unknown channel conditions. The theoretical problem was introduced in the 80's (e.g., [9]), but the research interest in the field was significantly invigorated during the last few years, due to the numerous identified network applications, such as image and video delivery (e.g., [3]). The main idea is to encode a file, for example an image, using a number μ of equally important descriptions. Each description is sent separately to the receiver, which, depending on the channel conditions, may receive only some of them. The descriptions are such that if a receiver receives *any* single one, it can reconstruct a coarse version of the image; more received descriptions allow for a more accurate reconstruction. Notice that only the *number* of different received descriptions matters for the reconstruction accuracy and not the order of reception.

Consider now an application that requires fast delivery of images over a wireless network, for example from a road-basestation of a transportation network to passing vehicles. Assume that the image is encoded using multiple description, and thus the basestation has μ blocks to deliver. When communicating towards a single receiver, simple sequential transmission of the blocks suffices. The problem becomes much more challenging when the image needs to be broadcasted to a number of receivers, each of which receives data over its own erasure channel. The base station may combine the feedback information with a *scheduling* algorithm to decide which image block to broadcast next. In this work, we propose instead to use a *coding* algorithm that transmits encodings of the image blocks. Our proposed coding is additional to the multiple description data compression: it decides which and how many image blocks it will combine together, and falls in the area of network coding, as its main purpose is to better share the network resources among the contending receivers. Also, our ideas apply to more general settings such as single-hop networks with many sources.

Every time receiver r_j receives successfully, it wants to learn some missing piece of information, namely *any* image block it does not know yet. This motivates us to increment the delay d_j of r_j by one every time r_j successfully receives either (i) an image block it already knows, or (ii) an encoding of image blocks which, when combined with r_j's successful receptions so far, does not allow r_j to immediately learn at least one unknown image block. This definition allows us to disengage delay from the erasure frequency as delay may only occur at successful receptions. It also allows us to capture two causes of delay: delay due to useless received packets, namely packets that bring duplicate information to their receiver, and delay due to packets that, although useful, do not allow their receiver to decode some unknown block at the time of their reception. Finally, our definition of delay is the simplest instantiation possible, as it does not take into account any ordering: we thus hope that a good understanding of this problem can serve as a first step towards more combinatorially demanding delay definitions.

The main questions we consider in this paper are (i) whether coding offers delay gains, and (ii) how to design coding schemes that minimize average and maximum delay, and what is the complexity of this task. We focus on the case where all receivers request the same content because understanding this simple model offers a first step towards variations, where receivers may demand different subsets of messages. It is worth noting that the popular solution of employing rate-less erasure correcting codes such as LT [13] or Raptor codes [16] for reliable broadcasting over erasure channels, yields very large delays (see I-A).

Our contributions include the following. We first show that minimizing average and maximum offline delay when the source uses scheduling is NP-hard. We then examine the complexity of the problem when coding is allowed and show that, although specific classes of erasure instances become trivial, the general problem remains NP-hard. However, we exhibit classes of erasure instances where coding offers significant gains in delay. We then discuss heuristic online algorithms for the case of i.i.d. erasures. We evaluate their performance through simulations and show that use of feedback and coding outperforms in terms of delay both scheduling, and Forward Error Correction (FEC) schemes that do not use feedback.

The remaining of the paper is organized as follows. Section II introduces our model. Sections III and IV study the complexity of offline scheduling and coding, and benefits from coding. Section V discusses online algorithms. Due to space limitations, detailed proofs of all theorems and propositions in these sections appear in [19]. Section VI concludes.

A. Related Work

A significant body of work has investigated the problem of scheduling user requests over a broadcast medium to maximize the per-user received rate and minimize the response time. Users typically arrive at different times, and ask for different content. No coding is employed and no errors occur. The difficulty of the problem, which was recently shown to be NP-hard [6], arises from having to share the common medium over the contending requests. Without erasures, our setting is an easy instance of this problem: when all users request the *same* data items in *any* order, even if they arrive at different times, a periodic (circular) transmission of the data items suffices.

In the presence of erasures, uncoded transmissions lead to repetitive reception and cannot achieve rates and thus delay close to optimal. With coding, delay and rate may become conflicting requirements. Rate-less codes for example need to encode at the source across μ blocks to operate close to capacity. A receiver must collect $\Theta(\mu)$ coded packets before it can decode, which incurs delay $\Theta(\mu)$. Indeed, when erasures occur, satisfying requests even for the same content becomes challenging [8]. In [17] use of MDS codes is proposed, but their performance is inferior to Raptor codes both in terms of complexity and adaptability to unknown channel conditions.

Our work can also be viewed as an instantiation of network coding with feedback. In [11], the goal is to optimize the achievable rate, when each received packet is either useless or can be immediately decoded by the destination. Such schemes, although simple to implement, do not offer rate or delay guarantees. Another line of work seeks to minimize the queue size at the sender (e.g., [18]). Also, [14] examines schemes that minimize mean completion time for broadcasting over a generalized variant of half duplex erasure channels. The last two performance metrics are quite different from delay.

A related broadcasting scenario, called Index Coding, was introduced in [4]. In the more general setting of [2], it is assumed that by time t each receiver knows some subset of the μ blocks (its *side information*), no erasures occur after time t, and each receiver wants exactly one of the blocks it is still missing. The goal is to find the minimum length of the codeword whose transmission will allow all receivers to simultaneously recover their missing blocks. In our setting, the assumptions above do not hold. Further, optimizing for the objective in [2] does not necessarily minimize ours.

In [10], solving Index Coding is viewed as solving a sequence of specific instances of the problem of network coding where the receivers place demands for specific sets of messages. In the full version of our paper, we show how minimizing delay is equivalent to solving a sequence of *specific* instances of a different problem, namely network coding with non-uniform demands (NUD). We exhibit a specific NUD instance that is hard (the one corresponding to answering if the delay has its minimum possible value), hence our results provide an alternative proof for the complexity results in [5].

Finally, our paper builds on a preliminary work [12], where we introduced the problem.

II. The Model

Consider a source that wants to convey μ messages to ρ receivers using broadcast transmissions. Time is slotted. At the beginning of each time slot $t \geq 1$, the source transmits packet $p(t)$. The source uses a *scheduling* scheme if $p(t)$ consists of one uncoded message for all t, while it uses a *coding* scheme if $p(t)$ may be an encoding (combination) of the messages. Receiver r_j receives the source transmissions over its erasure channel. We denote by $K_j^t \in \{0, 1\}$ the realization of r_j's channel at time t with $K_j^t = 1$ **iff** r_j receives $p(t)$. In the worst case (*offline)* model these realizations have given values, while in a probabilistic *(online)* setting they are random variables.

For all $t \geq 0$, receiver r_j informs the source of K_j^t. We assume perfect feedback channels and that the source receives K_j^t before the end of time slot t. Thus the source can use this information to generate the next packet. We assume that during time slot t, the receivers can receive $p(t)$ and decode it using previously received packets. A receiver who has decoded all

μ messages is no longer interested in the source transmissions which continue until all receivers have decoded all messages.

We can think of the μ source messages as defining a μ-dimensional space over a finite field $\mathbb{F}_q$, where each message corresponds to one of the orthonormal basis vectors $\{e_1, \ldots, e_\mu\}$. In this work we are interested in linear schemes, where $p(t)$ has the form (c, x) with $c \in \mathbf{F}_q^\mu$ and $x = \sum_j c_j e_j$; the choice of the coefficient vector c determines x, so we leave x implied in what follows. Operations over a finite field $\mathbf{F}_q$ of size say $q = 2^\ell$ in practice means that we divide the binary packets the source produces into contiguous sets of ℓ bits, and treat each such set as a symbol of $\mathbf{F}_q$. Linear combining of the packets occurs symbol-wise.

Let Π_j^t be the subspace collected by r_j *at the end* of time slot t and E_j^t the set of vectors $e_\ell \in \Pi_j^t$. We say that a received vector (packet) brings *innovative information* to r_j if it increases $\dim(\Pi_j)$ by one. Schemes where every successfully received packet brings innovative information to its receiver are called *rate-optimal*. In such schemes, if r_j has received ℓ packets, it has an ℓ-dimensional subspace Π_j of the μ-dimensional space. For $\ell = \mu$, r_j can successfully decode all source messages. (for more properties see [19], [12]).

Definition 1: The delay d_j^T experienced by r_j under transmission scheme T is the number of packets that, although successfully received, did not allow r_j to instantly decode a new message. [1] In symbols, $d_j^T \triangleq 1+\sum_{t:|E_j^t|<\mu} \mathbf{1}(E_j^t = E_j^{t-1}) \cdot K_j^t$, where $\mathbf{1}(\cdot)$ is the indicator function.

Let D_a^T, D_w^T denote the average, worst case delay under T respectively. Our goal is to compute $\min_T \mathcal{D}_a^T$, $\min_T \mathcal{D}_w^T$ and find the possibly different (see [12]) schemes that achieve them. Observe that if a scheme achieves the minimum delay of one for a given broadcasting instance, then the scheme is rate-optimal. On the other hand, any non rate-optimal scheme incurs average and maximum delay strictly larger than one.

Since the delay of any online scheme is lower bounded by the delay of the optimal offline scheme, we first investigate the offline model in Sections III and IV. [2] The offline problem has 4 inputs: μ; ρ; a time τ by which all receivers must have decoded all messages; and a $\tau \times \rho$ symbolic matrix P with entries from $\{\surd,\text{x}\}$ such that $P(t, j) = \surd$ **iff** r_j received $p(t)$. [3] A broadcasting scheme for the source *completes* an offline instance (μ, ρ, τ, P) if by time τ all receivers have decoded all messages. For example, if t_i is the first slot by which r_i has μ successful receptions, and $\tau \geq \max_{1 \leq i \leq \rho} t_i$, any rate-optimal scheme completes the instance (regardless of the delay).

III. Minimizing Scheduling Delay is NP-hard

Given an offline broadcasting instance (μ, ρ, τ, P), we want to minimize the average (maximum) delay under any scheduling scheme that completes the instance. A priori this appears to be an easier problem than the one studied in [6] since our notion of delay is relaxed: all receivers need all messages instead of specific subsets of messages, and the order of reception does not matter. The decision version of our minimization problem has an extra integer input $d \geq 1$, and answers "yes" **iff** there is a *scheduling* scheme that completes (μ, ρ, τ, P) with total (maximum) delay at most d [4] . An algorithm that solves the minimization problem must answer the decision problem for every value of d, including its minimum value which is one for both average and maximum delay. Thus showing that it is hard to decide if the average delay is one proves that both minimization problems are NP-hard. This is the main result of this section and it is summarized in the following theorem.

Theorem 1: Minimizing average and maximum offline delay under scheduling schemes is NP-hard.

In the rest of this section we sketch the proof of Theorem 1. We will reduce 3SAT to average delay (henceforth referred to simply as delay) of one.

Given a formula ϕ in CNF on n variables $x_1, \ldots, x_n$, and m clauses $c_1, \ldots, c_m$, where each clause consists of disjunctions of exactly 3 literals, we want to decide if there is an assignment of truth values to the variables that satisfies all clauses.

We will construct an offline broadcasting instance $B(\phi) = (\mu, \rho, \tau, P(\phi))$ such that ϕ is satisfiable **iff** there is a scheduling scheme that completes $B(\phi)$ with delay one. In our instance, the source has $\mu = 2n$ messages, there are $\rho = n + 2m$ receivers, and $\tau = 4n + 5m$ slots. In our construction each receiver has exactly $\mu = 2n$ successful receptions by time τ. This suffices to decide if a delay-one scheme for our instance exists: any such scheme must be rate-optimal (see Section II), thus must deliver all μ messages after μ successful receptions.

In more detail, for every variable x_i, $1 \leq i \leq n$ we introduce 2 messages, e_i, $\overline{e}_i$, and one receiver D^i (we will discuss the role of D^i after the construction of $P(\phi)$ is complete). Also, two receivers, C_1^j and C_2^j are introduced for every clause c_j, $1 \leq j \leq m$. Thus $P(\phi)$ has $\rho = n + 2m$ columns.

We now discuss the number of rows τ in $P(\phi)$. For every variable x_i, we introduce 4 consecutive time slots, which we call the *variable period* β_i; β_i starts at time $4i - 3$, and ends at time $4i$. Following the n-th variable period, we introduce m consecutive *clause* periods: the j-th clause period, denoted by γ_j, consists of 5 time slots, starts at time $4n + 5j - 4$, and ends at time $4n + 5j$. Thus $P(\phi)$ has $\tau = 4n + 5m$ rows.

To complete our construction, we must assign values to the $\tau \cdot \rho$ entries of $P(\phi)$. We will do this sequentially in time, i.e., by first considering the *variable* and then the *clause* periods.

Time	C_1^j	C_2^j
$4i-3$	$\surd$	x
$4i-2$	x	x
$4i-1$	x	$\surd$
$4i$	x	x

Time	C_1^j	C_2^j
$4i-3$	$\surd$	x
$4i-2$	x	x
$4i-1$	x	x
$4i$	x	$\surd$

Time	C_1^j	C_2^j
$4i-3$	$\surd$	$\surd$
$4i-2$	$\surd$	$\surd$
$4i-1$	x	x
$4i$	x	x

TABLE I

Receptions of C_1^j, C_2^j during β_i: as on the left, if clause c_j contains x_i; middle, if c_j contains $\overline{x}_i$; right, otherwise.

[1]The +1 is introduced for technical reasons and may be interpreted as setup time: e.g., $t = 0$ is used by the source to identify the number of receivers ρ.

[2]Besides serving as benchmarks for online performance, offline problems can be particularly interesting on their own (e.g., Index Coding).

[3]We use $P(t, j)$ for the offine model to distinguish from K_j^t used online.

[4]For μ independent of ρ, minimizing total or average delay are equivalent.

Time	D^1	$\dots$	D^i	$\dots$	D^n
$4i-3$	$\surd$	$\dots$	x	$\dots$	$\surd$
$4i-2$	$\surd$	$\dots$	x	$\dots$	$\surd$
$4i-1$	x	$\dots$	$\surd$	$\dots$	x
$4i$	x	$\dots$	$\surd$	$\dots$	x

Time slot	C_1^j	C_2^j
$4n+5j-4$	$\surd$	$\surd$
$4n+5j-3$	x	$\surd$
$4n+5j-2$	x	$\surd$
$4n+5j-1$	$\surd$	x
$4n+5j$	$\surd$	x

TABLE II
LEFT: $D^1, \dots, D^n$ DURING β_i. RIGHT: C_1^j, C_2^j DURING γ_j.

During variable period β_i, for all $1 \leq j \leq m$, receivers C_1^j, C_2^j corresponding to clause c_j receive as shown in Table I depending on whether x_i, $\overline{x}_i$ or none of them appears in c_j. Also, during β_i, receivers D^ℓ for $1 \leq \ell \leq n$, receive as in Table II: for $\ell \neq i$, D^ℓ receives during the first two time slots of β_i, while *only* D^i receives during the last two time slots.

During clause period γ_j, receivers C_1^j, C_2^j corresponding to clause c_j receive as shown in the right Table II. All other receivers experience erasures during γ_j.

The above completes our construction. It is easy to check that the reduction can be carried out by a deterministic Turing machine in logarithmic space, and that every receiver has exactly μ successful receptions. So a priori there could be a scheduling scheme T'_S completing $B(\phi)$ with delay one. Receivers D^i ensure the following property of all such T'_S.

Proposition 1: For all β_i, any T'_S that satisfies $B(\phi)$ with delay one, sends two *new* messages in the first two slots, and resends these messages in some order in the next two slots.

In effect, this flexibility in the scheduling of the messages during the last two slots of each β_i is our choice gadget. Our consistency gadget is that during β_i, C_2^j receives a different message from C_2^ℓ if x_i appears in c_j and $\overline{x}_i$ in c_ℓ. Our clause constraint gadget is the simultaneous reception of the two receivers corresponding to c_j during the first slot of γ_j.

We now move to showing that ϕ is satisfiable **iff** $B(\phi)$ admits delay one. Before, we introduce two useful schedulings.

Scheduling 1, 2 for variable period β_i: the ordered sequence of messages $\{e_i, \overline{e}_i, e_i, \overline{e}_i\}$, $\{e_i, \overline{e}_i, \overline{e}_i, e_i\}$, respectively.

Proposition 2: If ϕ is satisfiable, there is a scheduling scheme T_S that satisfies $B(\phi)$ with delay one.

Conversely, let T'_S be any scheduling scheme that satisfies $B(\phi)$ with delay one. W.l.o.g., assume that T'_S transmits $\{e_{x_i}, e_{y_i}\}$ during the first two slots of β_i. By Proposition 1, these messages will not be rescheduled before time $4n$, so for the sake of clarity, we may relabel them as $e_i, \overline{e}_i$ respectively. We define the following truth assignment. For $1 \leq i \leq n$, if T'_S applied Scheduling 1 for β_i, set x_i to true, else if T'_S used Scheduling 2 for β_i, set x_i to false. By Proposition 1, any T'_S indeed applied one of these two schedulings during β_i. The following show that the above truth assignment satisfies ϕ.

Proposition 3: Let $c_j = (\ell_i \vee \ell_a \vee \ell_b)$ be *any* clause. Under any T'_S that satisfies $B(\phi)$ with delay one, C_2^j has received at least one of e_i, e_a, e_b by time $4n$.

Corollary 1: If T'_S is a scheduling scheme that satisfies $B(\phi)$ with delay one then ϕ is satisfiable.

IV. BENEFITS AND LIMITS FROM CODING

We start here by attempting to understand the structural properties of instances where offline scheduling results in delay greater than one. We then show that coding across messages can offer two types of benefits: (i) Reduce the delay: we exhibit instances where coding achieves delay one, while scheduling cannot. (ii) Reduce the complexity of solving the problem: for example, with scheduling, it is NP-hard to compute the delay for the erasure pattern in Section III, while it is trivial to achieve the minimum delay of one with coding: during β_i, send $e_i, \overline{e}_i, e_i, \overline{e}_i$, while during β_j, for clause $c_j = (\ell_i \vee \ell_a \vee \ell_b)$, send $e_i + \overline{e}_i$, then whatever is missing from C_2^j, and finally $\overline{e}_a, \overline{e}_b$. The main purpose of this section is to examine whether and how much coding can help.

We use the following notation: B_t denotes the set of messages the source has transmitted up to time t, $\overline{B}_t$ the remaining messages. Similarly, E_j^t denotes the set of messages from B_t received by r_j, while $\overline{E}_j^t = B_t \backslash E_j^t$ for all r_j.

For the case of one receiver, trivially, scheduling achieves delay one. For two receivers, a simple algorithm ensures delay one: if at time t (i) both r_1 and r_2 receive, transmit a message from $\overline{B}_t$ (ii) only r_j receives, if $\bar{E}_j^t \neq \emptyset$ transmit a message from $\overline{E}_j^t$, else a message from $\overline{B}_t$. This scheme ensures that at each t either $\overline{E}_1^t = \emptyset$ or $\overline{E}_2^t = \emptyset$, and $\overline{B}_t = \emptyset$ only when at least one of the two receivers has received all messages.

For three receivers, offline scheduling can result in worst case delay $\mathcal{O}(\mu)$. Indeed, delay is introduced when the transmission scheme cannot be rate optimal. For the pattern in Table III, where each line is repeated $\frac{\mu}{2}$ times, rate optimality for r_3 implies that at $t = \mu + 1$, $\overline{E}_1^t \cap \overline{E}_2^t = \emptyset$ (r_3 is necessary to ensure $\overline{E}_1^t \cap \overline{E}_2^t = \emptyset$). Then the transmissions at $t = \mu + 1, \dots, \frac{3\mu}{2}$ incur sum delay $\frac{\mu}{2}$ for r_1 and r_2. Thus:

time-slots	r_1	r_2	r_3
$1, \dots, \frac{\mu}{2}$	$\surd$	x	$\surd$
$\frac{\mu}{2}+1, \dots, \mu$	x	$\surd$	$\surd$
$\mu+1, \dots, \frac{3\mu}{2}$	$\surd$	$\surd$	x

time-slots	r_1	r_2	r_3	r_4
$1, \dots, \frac{\mu}{2}$	$\surd$	x	$\surd$	x
$\frac{\mu}{2}+1, \dots, \mu$	x	$\surd$	$\surd$	x
$\mu+1, \dots, \frac{3\mu}{2}$	$\surd$	$\surd$	x	$\surd$

TABLE III
LEFT: SCHEDULING DELAYS $\mathcal{O}(\mu)$, CODING 1; RIGHT: CODING DELAY $\mathcal{O}(\mu)$.

Proposition 4: If at time t there are receivers r_i and r_j such that $\overline{E}_i^t \cap \overline{E}_j^t = \emptyset$, and after time t, for the next D timeslots with $D \triangleq \min\{|\overline{E}_i^t|, \quad |\overline{E}_j^t|\}$, both r_i and r_j succesfully receive, then offline scheduling results in delay $\mathcal{O}(D)$.

Use of coding allows to make all source transmissions rate optimal (e.g., for the left pattern in Table III, it suffices to transmit $\frac{\mu}{2}$ messages from $\overline{E}_1^t + \overline{E}_2^t$ at $t = \mu + 1, \dots, \frac{3\mu}{2}$), but delay is now introduced, if a receiver cannot decode a received linear combination, as shows the right pattern in Table III (see also [12]). It is easy to see that, at $t = \mu + 1$, $\overline{E}_1^t \cap \overline{E}_2^t = \emptyset$, and additionally, $\overline{E}_1^t \subset \overline{E}_4^t$, $\overline{E}_2^t \subset \overline{E}_4^t$. To be rate optimal for r_1 and r_2 the source must transmit from $\overline{E}_1^t + \overline{E}_2^t$. However, these transmissions cannot be decoded by r_4. Thus:

Proposition 5: If at time t there are receivers r_i, r_j and r_k with $\overline{E}_i^t \cap \overline{E}_j^t = \emptyset$, $\overline{E}_i^t \subset \overline{E}_k^t$, $\overline{E}_j^t \subset \overline{E}_k^t$, and after t, for the next D timeslots with $D \triangleq \min\{|\overline{E}_i^t|, \quad |\overline{E}_j^t|\}$, both r_j and r_k successfully receive, then offline coding incurs delay $\mathcal{O}(D)$.

Clearly, coding achieves delay one for a larger set of erasure patterns than scheduling. Some such patterns are given below.

Proposition 6: Coding achieves delay one when each transmission is: 1. Successfully received by at most two receivers (high erasure probability scenario). 2. Not received by at most one receiver (low erasure probability scenario).

Given that there are instances that become easier with coding, the next question is, whether the general problem, when we are allowed to use coding, becomes polynomial time, or remains NP-hard. Note that the problem of maximizing the throughput when multicasting over graphs becomes polynomial time if coding at intermediate network nodes is allowed [15], while it is NP-hard otherwise. However the following results show that this not the case in our problem. The proof of Theorem 2 builds on the ideas in the proof of Theorem 1.

Theorem 2: Minimizing average or maximum offline delay when the source uses (linear or nonlinear) coding is NP-hard.

Proposition 7: Unless $P = NP$, no $(2-\epsilon)$-factor approximation algorithm exists for maximum offline delay and $\epsilon > 0$ when the source uses (linear or nonlinear) coding.

V. Online Algorithms

We start by discussing the competitive ratio of a natural class of online algorithms for minimizing average and maximum delay in the case of arbitrary erasures. We then suggest an online heuristic that improves significantly on the average delay of the best heuristic from [12] for i.i.d. erasures.

A systematic rate-optimal online algorithm first transmits all μ messages once uncoded, then sends combinations of all messages. Such schemes have smaller delay than their non-systematic variants where linear combinations of all messages are always sent. However they perform poorly even in the presence of a deterministic adversary who injects arbitrary erasures but does not observe any channel.

Proposition 8: For $\mu = O(\rho)$ and arbitrary erasures, the competitive ratio of a systematic rate-optimal online algorithm is $\mu - O(1)$, $\mu - 1$ for average, maximum delay respectively. Proposition 8 motivates us to look at algorithms that are not necessarily rate-optimal in the online scenario. We focus on the case where all ρ channels experience i.i.d. erasures with common constant erasure probability q. Our new heuristic first sends the μ messages once uncoded. Then for all $t > \mu$, $p(t)$ is created as follows. First, a set of messages S is built by going over every source message e in a random order and setting $S = S \cup \{e\}$ if $S \cup \{e\}$ is instantly decodable (not necessarily innovative) for every receiver. Then, for every $e \notin S$, it updates $S = S \cup \{e\}$ if more receivers will delay upon reception of S than of $S \cup \{e\}$. Finally, the transmitted $p(t)$ is a random linear combination over the messages in S. Figure 1 compares the performance of this algorithm (Cost driven 2) with scheduling (always transmit the most needed message), and two heuristics from [12] (Systematic FEC and Cost driven 1), also discussed in [19]. Although our new heuristic is clearly suboptimal, it can improve by even 50% the performance of systematic FEC (a rate optimal algorithm with expected delay μq that does not use feedback) as q, ρ (graphs below) and μ (simulations not shown here) increase, and achieves more than 78% of the maximum rate for each point in the graphs. Variations aiming to weigh more cleverly the delays from useless and non instantly decodable packets are the subject of current work.

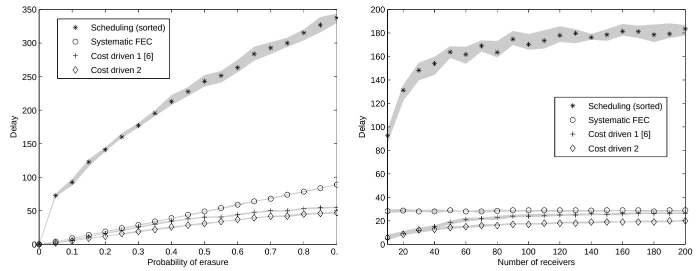

Fig. 1. Median delay; the gray area corresponds to confidence interval; $\rho = 150$ on the left, $q = 0.3$ on the right graph; $\mu = 100$ for both graphs

VI. Conclusions

We consider the problem of minimizing average and maximum delay when broadcasting with erasures. We show that the general offline problem is NP-hard under scheduling schemes, and remains NP-hard even under (linear or non-linear) coding schemes. However we demonstrate that coding offers delay and complexity gains offline, and that feedback information allows online algorithms specifically designed for delay-sensitive applications to outperform both scheduling and standard FEC (no feedback) schemes.

Acknowledgments

This work is supported by EU Project NetReFound FP6-IST-034413, SNSF Award No PP002–110483 and Hasler Foundation ManCom Project No 2072.

References

[1] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, pp. 1204–1216, July 2000.
[2] N. Alon, A. Hassidim, E. Lubetzky, U. Stav, A. Weinstein, ”Broadcasting with side information”, FOCS 2008.
[3] J. Apostolopoulos, M. Trott. Path diversity for enhanced media streaming. *IEEE Communications Magazine*, 42(8): 80–87, August 2004.
[4] Y. Birk, T. Kol, ”ISCOD over broadcast channels”, INFOCOM 1998.
[5] Y. Cassuto, J. Bruck, “Netw. Coding for Nonuniform Demands”, ISIT’05.
[6] J. Chang, T. Erlebach, R. Gailis, S. Khuller, “Broadcast scheduling: algorithms and complexity”, SODA 2008, pages 473–482.
[7] M.Durvy, C.Fragouli, P.Thiran, “On feedback for netw. coding”, ISIT’07
[8] K. Foltz, L. Xu, J. Bruck, “Coding and Scheduling for Efficient Loss-Resilient Data Broadcasting”, ISIT 2003.
[9] A. El Gamal and T. M. Cover, “Achievable rates for multiple descriptions,” *IEEE Trans. Information Theory*, vol. 28:6, pp. 851–857, 1982.
[10] S. El Rouayheb, A. Sprintson, C. Georghiades, “On the Relation between Index Coding and Network Coding Problems”, ISIT 2008.
[11] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, “XORs in the Air: Practical Wireless Network Coding”, Sigcomm06.
[12] L. Keller, E. Drinea, C. Fragouli, “Online broadcasting with Network Coding”, Network Coding 2008.
[13] M. Luby, “LT codes”, FOCS 2002.
[14] D. Lucani, M. Stojanovic, M. Médard, “Random Linear Network Coding for Time Division Duplexing”, Infocom’09.
[15] D. Lun, M. Médard, T. Ho, R. Koetter, ”Network coding with a cost criterion”, ISIT 2004.
[16] A. Shokrollahi, “Raptor Codes”, IEEE Trans. Inf. Theory, vol. 52, 2006.
[17] C. Su, L. Tassiulas, V. J. Tsotras, “Broadcast scheduling for information distribution”, Wireless Networks, Springer 2004.
[18] J.Sundararajan, D. Shah, M. Médard, “ARQ for Netw. Coding”, ISIT’08.
[19] http://infoscience.epfl.ch/record/126035