Physical Communication 6 (2013) 100–113

Contents lists available at SciVerse ScienceDirect

Physical Communication

journal homepage: www.elsevier.com/locate/phycom

Full length article

Real-time delay with network coding and feedback

Eleni Drinea, Lorenzo Keller *, Christina Fragouli

École Polytechnique Fédérale de Lausanne (EPFL), CH-1015, Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 15 April 2011
Accepted 3 May 2012
Available online 11 May 2012

Keywords:
Network coding
Broadcast
Feedback
Complexity
Delay

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. The sender can broadcast a single message or a combination of messages at each timestep, through separate erasure channels. 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 in that it combines coding techniques with feedback information to the end of minimizing delay. We show that it allows $\Theta(\mu)$ benefits as compared to previous approaches for offline algorithms, while feedback allows online algorithms to achieve smaller delay compared to online algorithms without feedback. Our main complexity result is that the offline minimization problem is *NP*-hard both under scheduling and coding algorithms. However we show that coding does offer delay and complexity gains over scheduling. We also discuss online heuristics and evaluate their performance through simulations.

1. Introduction

Current and emerging applications, such as satellite imaging, roadside to vehicle communication, internet TV and 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 subset of 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 [1], or implicitly, by overhearing transmissions from the receiver nodes [2]. In satellite transmissions, a satellite might learn when a receiver goes in a deep fade (e.g., enters a tunnel), in which case it loses a sequence of packets. A similar approach to explicitly collect acknowledgments in wired networks, when the source multicasts the same content over a distribution tree in an overlay network appears in [3]. We will assume in this paper that the source has perfect feedback information.

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.

* Corresponding author.
E-mail addresses: edrinea@gmail.com (E. Drinea), lorenzo.keller@epfl.ch (L. Keller), christina.fragouli@epfl.ch (C. Fragouli).

1874-4907/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.phycom.2012.05.005

Multiple description is a well studied data compression technique which allows to provide robustness and graceful recovery in the presence of unknown channel conditions. Although the theoretical problem was already introduced in the 80's (see for example [4]), 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 (see for example [5–10] for a tutorial paper). The main idea is that we encode our file, for example an image, using a number μ of equally important descriptions, and each description is sent separately to the receiver. Depending on the channel conditions, the receiver may receive a different number of descriptions. These descriptions are constructed to have the following property: if a receiver receives a single description (any one of them), it gets a coarse version of the image that is within some appropriately defined distortion guarantees from the original. If a receiver gets *any* two descriptions, it can reconstruct the image more accurately. Generally, the larger the number of descriptions received, the smaller the distortion distance of the reconstructed image from the original. Reception of all descriptions results in the most accurate reconstruction. Note that in this construction, it is only the number of different received descriptions that defines the reconstruction accuracy; the ordering at which descriptions are received plays no role.

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 descriptions, and thus the basestation has μ descriptions to deliver. When communicating towards a single receiver, simple sequential transmission of the descriptions suffices: the underlying multiple description code will determine the image quality experienced by the receiver, as a function of the number of different descriptions collected.

The problem becomes much more challenging when the image needs to be broadcasted to a number of receivers, each of which receives information over its own erasure channel. The sender may use a *scheduling* algorithm to decide which image description to broadcast next. In this paper, we propose instead to use a *coding* algorithm, that encodes the descriptions we need to transmit to the receivers. Both in the case of scheduling and coding, the algorithm may decide on the current transmission by using the feedback information it has collected, i.e., which receivers received the previous transmissions. Note that our proposed coding is additional to the multiple description data compression: it decides which and how many image descriptions it will combine together, and falls in the area of network coding (see [11–13] for introductory tutorials), as its main purpose is to better share the network resources among the contending receivers.

In order to reconstruct the original image, every time a receiver receives, it wants to learn some missing piece of information, namely *any* image description it does not know yet. This motivates us to increment the delay d_j of receiver r_j by one every time r_j successfully receives a transmission of the following type: (i) an image description r_j already knows, or (ii) an encoding of image descriptions which, when combined with r_j's successful receptions so far, does not allow r_j to extract some image description it does not know yet. This definition allows us to disengage delay from the erasure frequency as we only count delay when a transmission is successful. 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 message upon 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 benefits in terms of delay, and (ii) how to design coding schemes that minimize average and maximum delay, and what is the complexity of this task. We focus in the case where all receivers are interested in the same content and believe that this simple model will provide insight for variations where receivers may demand different subsets of the messages or request the messages in a specific order. It is worth noting that the popular solution of employing rate-less erasure correcting codes at the source such as LT or Raptor codes [14,15] for reliable broadcasting over erasure channels, performs very poorly in terms of delay (see also Section 1.1).

Our contributions include the following. Concerning the complexity of the offline problem, we show that minimizing the average and maximum 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. We examine classes of erasure instances where coding offers significant benefits in terms of delay, and give a simple inapproximability result for maximum offline delay. Finally, we discuss heuristic online algorithms where the erasures of different receivers are independent and i.i.d. distributed. We evaluate the performance of our heuristics through simulations. The latter verify our observation that coding can significantly reduce delay compared to scheduling.

The importance of our work lies perhaps in that, to the best of our knowledge, it was the first to examine the complexity and algorithmic aspects of the joint use of coding and feedback information for delay-optimal content delivery. Erasures are inherent in many realistic networks, and we believe that the trade-off between rate and delay that arises in our setting is worth exploring further.

The remainder of this paper is organized as follows. Section 2 introduces our model and notation. Section 3 examines the complexity of offline broadcasting with scheduling, while Section 4 examines the complexity when coding at the source is allowed. Section 5 discusses online results. Section 6 concludes the paper.