Theoretical Computer Science 530 (2014) 12–22

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Competitive router scheduling with structured data ☆

Yishay Mansour [a,1], Boaz Patt-Shamir [b,2], Dror Rawitz [b,*]

[a] *School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel*
[b] *School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel*

ARTICLE INFO

Article history:
Received 27 January 2012
Received in revised form 28 June 2013
Accepted 17 February 2014
Communicated by T. Erlebach

Keywords:
Competitive analysis
Packet fragmentation
Randomized algorithms

ABSTRACT

We consider the task of transmitting structured information over bounded-capacity links. Our information model is a stream of basic units called *superpackets* that are broken into k packets each. To model the possible structure and redundancy of the superpackets, we assume that for each superpacket there is a collection of minimal subsets of packets whose delivery makes the superpacket *useful*. This very general model encompasses, for example, MPEG streams, where one can think of a group of pictures (GoP) as a superpacket. The fundamental difficulty is that networks can forward only the primitive packets, but applications can use only superpackets, and thus if no minimal subset is delivered, the whole superpacket becomes useless. Our aim is to maximize goodput (number of useful superpackets) in the face of overloaded communication links, where we are forced to drop some packets.

Specifically, we assume that an arbitrary stream of packets arrives at a router with multiple bounded-capacity outgoing links. An online algorithm needs to decide, for each superpacket, which outgoing link to use (all packets of the same superpacket must use the same link) and, in case of an overload at a link, which packets to drop and which to transmit so as to maximize goodput. We analyze a simple randomized competitive algorithm for the general case and provide a nearly matching lower bound on the competitive ratio of any randomized online algorithm.

1. Introduction

Consider a video stream encoded in MPEG-2 [11]. Grossly oversimplifying, the structure of the stream is as follows. The stream is broken into *Groups of Pictures* (GoP), which may last a few minutes each. A GoP consists of a single *I-frame*, a few *P-frames*, and many *B-frames*. An I-frame is a stand-alone picture that requires no other information for decoding; decoding a P-frame requires its preceding I-frame; and decoding a B-frame requires its preceding "reference frame" (be it I- or P-frame).[3] The implication of this structure is that if an I-frame is lost, then the whole GoP is lost, and if a B-frame

☆ A preliminary version was presented at the 9th Workshop on Approximation and Online Algorithms, 2011. Research supported in part by the Next Generation Video (NeGeV) Consortium, Israel (www.negev-initiative.org).

* Corresponding author.
E-mail addresses: mansour@cs.tau.ac.il (Y. Mansour), boaz@eng.tau.ac.il (B. Patt-Shamir), rawitz@eng.tau.ac.il (D. Rawitz).

[1] Supported in part by a grant from the Israel Science Foundation, a grant from the United States–Israel Binational Science Foundation (BSF), a grant by Israel Ministry of Science and Technology and the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11).

[2] Supported in part by the Israel Science Foundation (Grant 1372/09) and by Israel Ministry of Science and Technology.

[3] In fact, P-frames depend on the previous reference frame; and B-frames depend on both their immediate surrounding frames. In addition, MPEG partitions frames into "slices," which are transmitted in network packets.

http://dx.doi.org/10.1016/j.tcs.2014.02.019
0304-3975/

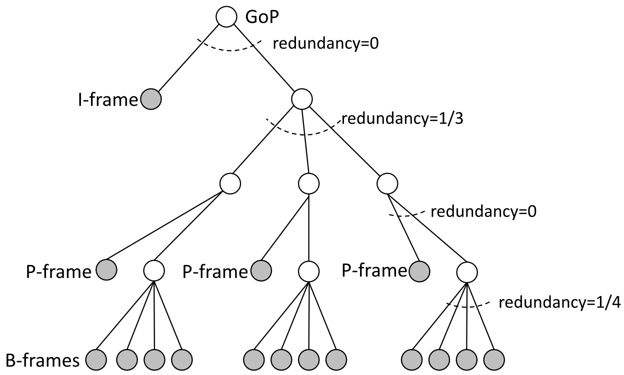

Fig. 1. A tree representation of a GoP. Gray nodes represent data. A node with redundancy β is deemed useful if no more than a fraction β of its children is non-useful.

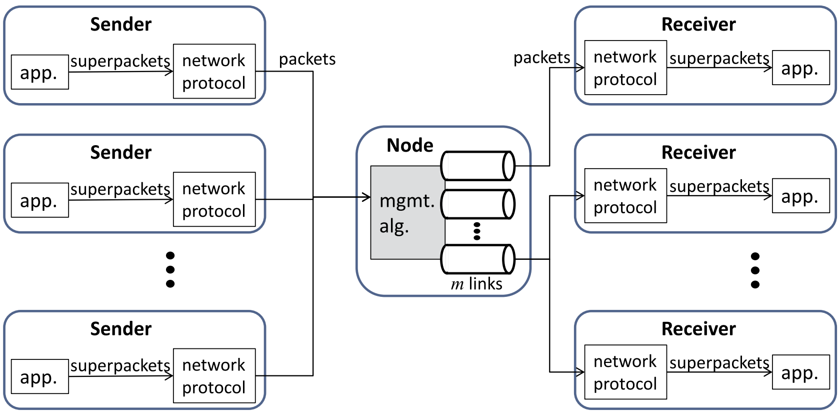

Fig. 2. Basic system setup. Our focus is on the link management algorithm (shaded). All packets belonging to the same superpacket must use the same link.

is lost, then only a fraction of a second is lost. But then again, if too many B-frames are lost (where "too many" is defined subjectively), the GoP should be considered again worthless. This structure can be modeled by a tree. Fig. 1 illustrates a simple example.

The root represents the GoP; if either the I-frame (left child) or the other data (right subtree) are lost, then the GoP is lost; however, the right subtree may be considered useful even if one of its children is lost; and similarly, each of these (depth 2) nodes is useful only if both its P-frame child and at least 3/4 of its B-frames are not lost. While this is not an accurate description of MPEG, we note that the hierarchical tree structure is very natural and appears in many other formats (e.g., XML documents [17]), with or without redundancy. Conceivably, more complex forms of redundancy are also used.

Bearing this traffic model in mind, suppose that we need to manage a router that delivers multiple video streams, such that each stream may use any of a number of outgoing links (see Fig. 2). At every step, some packets arrive at the router, and the router needs to decide which outgoing link is used for each packet, and, in case of an overflow in that link, which packets to discard. Note that in our example, if we drop an I-frame from each GoP, then all GoP's are useless at the receiving ends, even if the link has delivered all P- and B-frames (this is an instance of a high throughput, low goodput situation). In this paper we study, from the theoretical viewpoint, algorithms that decide which packets to drop so as to maximize the goodput of bounded-capacity links.

To this end, we consider the following abstract model. Senders generate basic information units, called *superpackets*, that are broken into packets by the network protocol at the senders. The router needs to decide which link is used by each new superpacket: all subsequent packets of that superpacket must use the same link.[4] If the number of packets assigned to a link exceeds its capacity, the management algorithm needs to decide which packet to drop and which to forward. To allow for arbitrary structure and redundancy, we assume that each superpacket is associated with a collection of *feasible subsets* that is closed under set inclusion (i.e., if $S \supset S'$ and S' is feasible, then so is S). A superpacket is considered useful only if the set of its delivered packets is one of its feasible subsets. The goal of the algorithm is to maximize the number (or weight) of useful superpackets at the receivers.

[4] This requirement, referred to as "stickiness" or "persistence" is typical in communication protocols, e.g. TCP.