

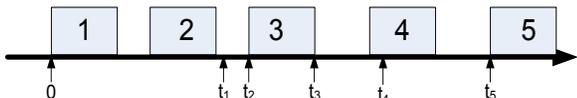
Optimal Universal Schedules for Discrete Broadcast¹

Michael Langberg, Alexander Sprintson, and Jehoshua Bruck
California Institute of Technology, Pasadena, CA 91125
Email: {mikel,spalex,bruck}@paradise.caltech.edu

In this paper we investigate efficient schedules for sending dynamic data over lossless broadcast channels. We consider a system in which the server periodically transmits highly dynamic data to a number of passive clients. We study the case in which the data consists of *discrete* packets, where each update is sent in a separate packet. Each client listens to the channel in order to obtain the most recent data.

Our goal is to allow for each client to access the most recent data as soon as possible. In particular, we want to minimize the time elapsed since the client started to listen to the channel until it received the information. Designing efficient broadcast schedules (with respect to this objective) attracted a large body of research (see e.g., [1, 2]). To the best of our knowledge, all previous studies assumed that clients' requests are distributed uniformly over time. However, in general settings, the clients' behavior is difficult to predict. In this paper we focus on *universal* schedules; that is to say, schedules that must perform well for *any* possible request sequence.

We consider the basic framework of scheduling packets with continuously updated content over a single broadcast channel. In this framework, packets of equal length (of say one unit) are being broadcasted periodically. The client must listen to at least one packet from beginning to end in order to satisfy a request. Our objective is to design universal schedules that minimize the time that passes between a client's request and the broadcast of a new item (referred to as the *waiting time* of the client), independently of the client's behavior. A schedule example is depicted in the following figure. In this example if a client's request arrives at time t_1 the client will wait until time t_2 which is the arrival of the next packet (number 3); the request will be completed at time t_3 . If a request arrives at time t_4 , the client must wait until the transmission of packet 5 that begins at time t_5 .

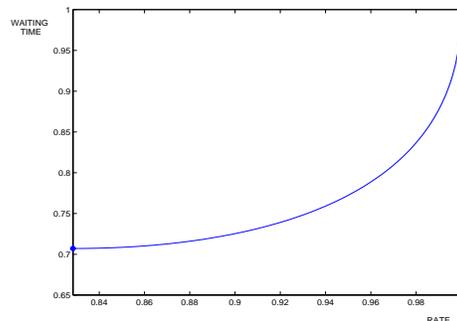


Design of universal schedules is an interesting task and poses many major challenges. First, consider the trivial schedule, in which the packets are sent immediately one after another. We observe that if a client issues a request shortly after the beginning of the transmission of the current item, then it will need to wait almost an entire unit of time until it will start receiving a new message. We observe that this also holds for any *deterministic* schedule. We thus turn to consider *random* scheduling strategies of the server. In the random setting, the server has a distribution over schedules instead of a single scheduling. Instead of considering the waiting time of any request given at time t , we consider its *expected* waiting time. Notice that in this case the expectation is taken over the randomness of the server, as apposed to the standard scheduling models in which the expectation is taken over the assumed distribution of the clients.

Our results. In our work we consider *adaptive* clients, i.e., clients whose behavior depends on information gained (via

previous requests) on the history of the schedule. It turns out that our ability to construct good schedules depends on the *degree of adaptivity*, i.e., amount of time elapsed between the information on the history of the schedule and the time a request is placed. We focus on clients whose degree of adaptivity is one time unit. We present a scheduling strategy which promises an expected waiting time strictly lower than 1, no matter when the request was placed, or what the viewed history of the channel was before the request. Specifically, our scheduling strategy guarantees a worst case expected waiting time of no more than $1/\sqrt{2} \simeq 0.7$. Moreover, we show that this is the best strategy possible. Namely, we show that no matter what scheduling strategy is used, it is impossible to obtain a universal waiting time of value less than $1/\sqrt{2}$.

We also consider the *transmission rate* of a schedule, which is defined to be the average number of packets sent over a period of time. The transmission rate of the schedule is important for clients that need to receive as many updates as possible. It turns out that there exists a trade-off between the transmission rate and minimum worst-case waiting time. Our optimal schedule has a high rate of $r = \frac{2}{1+\sqrt{2}} \simeq 0.82$. Moreover, for any larger rate r , we present a schedule of rate r with expected waiting time of no more than $\frac{2-r-\sqrt{2-2r}}{r}$ time units, no matter when the request was placed, or what the viewed history of the channel was before the request. We show that this is the optimal schedule possible under certain restrictions on the server. The tradeoff between the transmission rate and worst case expected waiting time is depicted in the following figure.



The full version of this paper appears in [3].

ACKNOWLEDGMENTS

We would like to thank Kevin Foltz for useful discussions.

REFERENCES

- [1] K. Foltz and J. Bruck, "Splitting Schedules for Internet Broadcast Communications," *IEEE Transactions on Information Theory*, vol. 48, no. 2, pp. 345–358, 2002.
- [2] M.H. Ammar, "Response Time in a Teletext System: An Individual User's Perspective," *IEEE Trans. Comm.*, vol. COM-35, no. 11, pp. 1159–1170, 1987.
- [3] M. Langberg, A. Sprintson, J. Bruck, "Optimal Universal Schedules for Discrete Broadcast," ETR057, 2003, available on-line at www.paradise.caltech.edu/ETR.html.

¹This work was supported in part by the Caltech Lee Center for Advanced Networking and by NSF grant ANI-0322475.