Optimal Schedules for Asynchronous Transmission of Discrete Packets

Michael Langberg   Alexander Sprintson   Jehoshua Bruck

California Institute of Technology.
Email: {mikel,spalex,bruck}@caltech.edu

Abstract

In this paper we study the distribution of dynamic data over a broadcast channel to a large number of passive clients. Clients obtain the information by accessing the channel and listening for the next available packet. This scenario, referred to as *packet-based* or *discrete broadcast*, has many practical applications such as the distribution of weather and traffic updates to wireless mobile devices, reconfiguration and reprogramming of wireless sensors and downloading dynamic task information in battlefield networks.

The optimal broadcast protocols require a high degree of synchronization between the server and the wireless clients. However, in typical wireless settings such degree of synchronization is difficult to achieve due to the inaccuracy of internal clocks. Moreover, in some settings, such as military applications, synchronized transmission is not desirable due to jamming. The lack of synchronization leads to large delays and excessive power consumption. Accordingly, in this work we focus on the design of optimal broadcast schedules that are robust to clock inaccuracy. We present *universal* schedules for delivery of up-to-date information with minimum waiting time in asynchronous settings.

1 Introduction

In recent years, there has been a growing interest in systems for the distribution of dynamic data over a broadcast channel to a large number of passive clients. Such systems have numerous applications such as the distribution of weather and traffic updates to wireless mobile devices, reconfiguration and reprogramming of wireless sensors and downloading dynamic task information in battlefield networks.

The design of broadcast distribution systems is a complex task and poses many challenges. First, such systems must provide fast access to up-to-date information for all clients. Second, as the battery power is a scarce resource in wireless networks, the amount of energy needed to receive a data item should be minimized. Third, the systems must be *scalable*, i.e., be able to serve a large number of clients in an efficient way. Finally, the systems need to deal with the inherent inaccuracy of internal clocks and lack of synchronization among clients.

In order to achieve a high degree of scalability, a "*push-based*" approach has recently been proposed. With this approach, the server proactively transmits the data over the broadcast channel (see Fig. 1). When a client needs the information, it accesses the channel and listens for the next available packet. In contrast to the more traditional "*pull-based*" approach the clients do not send their requests to the server. This enables efficient service for a very large number of clients. Keeping the clients passive reduces energy consumption and keeps the location of the clients in secret.

Since typical devices used in wireless sensor and military networks have limited functionality, the broadcast protocol itself must be simple and easy to implement. Accordingly, in [8] we introduced a notion of *packet-based* or *discrete broadcast*. The main idea is to transmit data in the form of *discrete* packets. With this approach, clients receive all required information from just a single packet. Packets may contain the latest configuration of a wireless sensor or the up-to-date situation on a battlefield.

The major problem in the design of discrete broadcast systems is to devise efficient schedules that minimize the amount of time a client has to wait in order to receive the requested information, i.e., the *waiting time* of

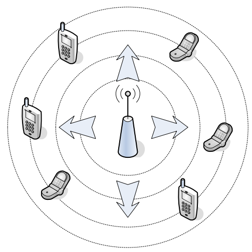

Figure 1: Wireless network with passive clients.

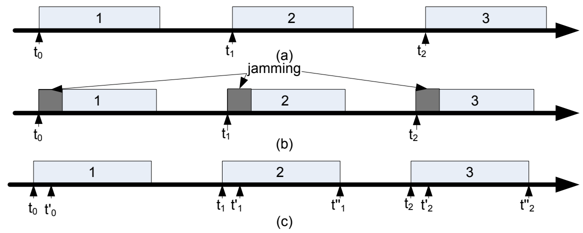

Figure 2: Broadcast schedules. (a) Synchronous schedule (b) Jamming (c) Worst-case example.

the client. This problem can be efficiently solved if all clients have accurate internal clocks and the system is perfectly synchronized. A simple solution in this case would be to transmit the packets at predefined time slots $t_0, t_1, t_2, \ldots$, as depicted in Fig. 2(a). However, such a synchronization is difficult to achieve due to the internal inaccuracy of local clocks. Moreover, in some settings, such as military applications, synchronized transmission is not desirable due to jamming. Indeed, an enemy can easily disrupt the broadcast by transmitting jamming messages at times $t_0, t_1, t_2, \ldots$, as depicted in Fig. 2(b). Furthermore, the synchronized transmission is not robust and highly sensitive to the inaccuracy of internal clocks. Indeed, suppose that the internal clock of a client is skewed by a small amount Δ_t. Such a client listens to the channel at times $t'_0, t'_1, t'_2, \ldots$, as depicted in Fig. 2(c), where $t'_i = t_i + \Delta_t$. Since the client misses the transmission of the current packet, it has to wait for the next packet, which incurs large delay and wastes power.

This Work

In this paper we present *universal* schedules that operate efficiently in asynchronous settings. We focus on schedules that guarantee low worst-case waiting time and, at the same time, deliver up-to-date information to all clients. We show that such schedules can be efficiently designed by adding redundancy and employing randomization techniques.

In order to evaluate the worst-case performance of a broadcast schedule, we use the notion of an *adversarial* client. The goal of such a client is to maximize waiting time by generating requests at times which are least desirable for the scheduler. A schedule that performs well against such an adversary will perform well in asynchronous settings and will be robust to clock skew and jamming.

The schedules we present are energy efficient and inherently less susceptible to jamming. While a rigorous analysis of energy efficiency and jamming is beyond the scope of this paper, our methods and tools lay the foundation for dealing with these problems.

Related Work

The design of efficient broadcast schedules for the push-based model has attracted a significant body of research (see e.g., [1–3, 5–7]). Prior work in this area typically assumes that client requests are distributed uniformly over time and focuses on minimizing the *average* waiting time. Such schedules, however, may incur large delays in the *worst case*. Furthermore, all of these works focus on deterministic schedules, whose performance is very sensitive to jamming and clock skew.

The adversarial model was first introduced and investigated in our previous work [8]. This work considers universal broadcast schedules in which each unit packet is transmitted as is, i.e. no redundancy is introduced, and certain restrictions are placed on the adversary. Under this model, a tight analysis of the worst-case expected waiting time is presented. In the current paper we consider a general framework that uses redundancy and does not impose any limitations on the adversary. It turns out that the framework considered in this paper requires new approaches and solution techniques.

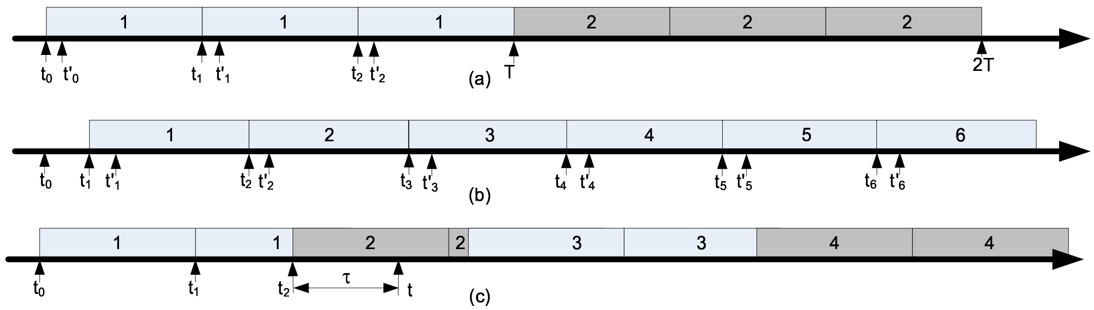

Figure 3: Examples of possible schedules. Each packet is marked by a number that specifies the order in which the packets were generated by the source.

Organization

The rest of the paper is organized as follows. In Section 2, we present the basic framework of scheduling packets over a discrete broadcast channel. In Section 3, we formally define our model. This section includes the reduction to i.i.d. schedules. In Section 4 we derive the expressions that tie our objective functions (waiting time and staleness) to the function F governing an i.i.d. schedule $\mathcal{S}$. In Sections 5 and 6 we present optimal and approximate broadcast schedules. In Section 7 we consider clients whose requests are distributed uniformly over the time. In section Section 8 we summarize our results and analyze important tradeoffs in the design of broadcast schedules. Finally, conclusions and directions for future work appear in Section 9.

2 Efficient Broadcast Schedules

In this section we consider the basic framework of scheduling packets with continuously updated content over a broadcast channel. In this framework, packets of equal length (say one unit) are being broadcasted periodically. Given a request, the client listens for the next available packet. The client must listen to at least one packet from beginning to end in order to satisfy a request. Our goal is to design schedules that minimize the time that passes between a request and the broadcast of a new packet (referred to as the waiting time of the client) in an asynchronous setting.

A basic example

As discussed in the Introduction, the simple approach of transmitting packets at predefined time slots can be very sensitive to the inaccuracy of internal clocks. Indeed, consider the schedule depicted in Fig. 2(c) and suppose that a client makes its request at time $t_1' = t_1 + \Delta_t$, where Δ_t is a small time period. Given this request the client begins to listen to the channel immediately. During the time interval $[t_1', t_1'']$, the client listens to the first packet, but receives only a part of it. However, such partial information is insufficient so the client must continue to listen to the channel until it receives a full packet (in time $t_2'' = t_2 + 1$). In this case we define the waiting time of the client to be $t_2 - t_1'$ (we do not count the additional unit of time required for the transmission of the final packet).

Adding redundancy

Robustness of a broadcast schedule can be improved by adding redundancy. Such a redundancy can be achieved by transmitting each packet several times over long time intervals. Specifically, each packet is periodically transmitted over a time interval of length $T > 1$. Such an encoding which is, in fact, a repetition code, suffices for a noiseless channel; in the presence of noise a more involved transmission scheme can be considered.

We depict a schedule that uses the repetition code in Fig. 3(a). Note that this schedule improves waiting time for certain requests. Indeed, suppose that a client arrives at time $t_1' = t_1 + \Delta_t$. In this case, the client obtains the first part of the first packet during the time interval $[t'_1, t_2]$ and the second part of the same packet during the interval $[t_2, t'_2]$, where $t'_2 = t_2 + \Delta_t$. Thus, the waiting time for a request that arrives at time t'_1 is zero, which is a significant improvement over the schedule presented on Fig. 2(c). However, the worst-case waiting time has not been improved. Indeed, for a request that arrives at time t'_2, the waiting time is $1 - \Delta_t$, which is prohibitively high. We observe that (almost) every deterministic schedule has worst-case waiting time close to 1. The only exception is the schedule that transmits the same packet over and over again (its corresponding interval will be of length $T = \infty$). However, such a schedule is not useful, because it cannot deliver new information.

Adding randomness

In this section we show that the worst-case waiting time can be improved by adding randomness. We begin by presenting a simple random schedule depicted on Fig. 3(b). The transmission begins at time t_1 which is drawn uniformly from the interval $[0, 1]$. Then next packets are transmitted at times $t_2, t_3, \ldots$, such that $t_{i+1} = t_i + 1$.

Notice, that once we introduced randomness, the server has a distribution over schedules instead of a single schedule. Instead of considering the waiting time of any request given at time t, we consider its *expected* waiting time. The expectation we consider is taken over the randomness of the server, as opposed to the standard scheduling models in which the expectation is taken over the assumed distribution of the clients. Our objective in the design of universal random schedules is now to minimize the worst-case expected waiting time. Accordingly, the adversary's goal is to pick a time t, for which the expected waiting time is large. With this new schedule, it is not hard to verify that the expected waiting time is equal to 0.5 time units for any request time t. This is a significant improvement over the deterministic case, and demonstrates that the introduction of randomness is crucial for designing universal schedules.

However, once randomness is introduced, one must take into consideration the adaptiveness of the assumed adversary. Namely, in the discussion above we have assumed that the adversary is *oblivious* to its view of the schedule so far. This implies that the quality analysis of the schedule assumes that the clients requests do not depend on the information previously obtained by the client (through previous requests). However, in the most general setting, the adversary may be *adaptive*, generating its requests based on the previously broadcasted information.

We observe that for an adaptive adversary, the worst-case expected waiting time of the schedule depicted in Fig. 3(b) is also arbitrary close to one. Indeed, consider the case in which a client requested the initial information at time 0. Such a client will wait until time t_1 for the transmission of the first packet. Now consider a (malicious) second request of the client at time $t'_2 = t_2 + \Delta_t$, for some small $\Delta_t > 0$. In this case the clients waiting time on the second request is arbitrarily close to one.

It is important to note that in this case the adversarial approach captures the situation that can occur in cases where client's clock is inaccurate. For example, consider a client which after an initial request at time 0, is interested in an update after some time has passed (say after 5 time units). Based on the information gained by the first request, the client decides to place an additional request at time $t_5 = t_1 + 5$ (so as to minimize waiting time). However, unfortunately, the client's clock is inaccurate and the actual request is at time $t_5 + \Delta_t$, for some small $\Delta_t > 0$. In this case the client's expected waiting time on the second request is $1 - \Delta_t$, which is again prohibitively long. We note that the adaptive adversarial approach permits the design of schedules that are less susceptible to jamming. Indeed, the enemy can easily adapt its jamming strategy by listening to the channel.

Combining randomness and redundancy

The natural question that arises in this context is whether it is possible to construct schedules that achieve lower waiting time in the presence of an (adaptive) adversary. As we show in this paper, the answer to this question is positive. We achieve this goal by introducing both *randomness* and *redundancy* in the design of broadcast schedules.

Specifically, we employ schedules that transmit each packet several times over intervals or random length T. We begin by presenting a simple schedule whose worst-case expected waiting time is strictly less than 1 (in the presence of an adaptive adversary). As we will see, while the adversary knows the distribution governing the schedule, it does not know the outcome of future coin-tosses, which makes it hard to decide when to place a malicious request.

Example 1 Consider the schedule in which the length of each interval is uniformly distributed on $[1, 2]$. An example of an execution of such a schedule is depicted on Fig. 3(c). We analyze the worst-case expected waiting time of this schedule. Consider a request placed at time t. Let τ represent the time passed between the beginning of the interval transmitted at the time of the request and the time t (for example let $\tau = t - t_2$ as depicted in Fig. 3(c)). We consider two cases: (1) If $\tau \leq 1$, then with probability τ the waiting time will be nonzero (this happens when the length of the interval is less than $1 + \tau$). Thus, a simple computation reveals an expected waiting time of $\tau(1 - \frac{\tau}{2})$. (2) If $1 < \tau \leq 2$, then the adversary knows that the length of the interval is at least τ. Hence, by conditioning on this event, the adversary can deduce that the length of the interval is uniformly distributed on $[\tau, 2]$. Again, simple computations show that the expected waiting time is now $\frac{2-\tau}{2}$.

We conclude that the worst-case expected waiting time of this schedule is just 0.5. This follows by the fact that any client request must fall at most two time units after the beginning of some transmitted interval. Moreover, the maximum expected waiting time is obtained when a request is placed one time unit after the beginning of an interval. We conclude that the simple randomization strategy presented above reduces the worst-case waiting time by 50% compared to a deterministic schedule.

Introducing Staleness

Until now we have shown that, by the addition of randomness and redundancy, one can obtain schedules with low expected waiting time, no mater when the request is placed, or what the viewed history of the channel is before the request. However, this result comes at a price: there may be requests in which the information received by the client is not "fresh". Namely, due to the addition of redundancy, the client may receive information which was sampled by the server long before the client's request. Since, the state of the information source changes all the time, this received information may no longer be relevant.

The construction of schedules with low waiting time which guarantee "fresh" information is the central problem studied in this work. Accordingly, we introduce a new notion, the *staleness* of the information, that captures the age of the information delivered to the client. The staleness is defined to be the amount of time passed between the time the information was sampled by the server and the time at which the client begins to receive the information, or equivalently, between the moment a transmission of a new encoded packet starts and the time at which the client begins to receive the information.

Let us analyze the expected staleness of the random schedule described in Example 1. Again, we analyze what happens if the adversary puts its request τ units after the beginning of an interval. We consider two cases. If $\tau \leq 1$, then with probability $1-\tau$ the length of the transmitted packet is greater than $1+\tau$ and the staleness is equal to τ; while with probability τ the staleness equals to zero (as the client must wait for a new packet, which when received will be fresh). Thus, the expected staleness is $\tau(1 - \tau)$. If $\tau > 1$, then the client always waits until the beginning of a new interval, in which case the expected staleness is zero. We stress that in the latter case, the client receives fresh information after its wait (i.e., information with zero staleness). We conclude that the worst-case expected staleness of this schedule is 0.25 and it occurs if the adversary puts its request 0.5 time units after the transmission of a new packet begins.

The main goal of this paper is to identify a probability distribution over schedules that yields both low worst-case expected waiting time and low worst-case expected staleness. Namely, we seek to find schedules that obtain the minimum worst-case expected waiting time under a given staleness constraint. As we shall see, it is possible to construct a better schedule distribution than that presented in Example 1.

3 Model

As mentioned in the Introduction, our goal is to design a schedule that achieves low waiting time and staleness in the presence of adaptive adversarial clients. We assume that the transmission of a packet takes one time unit. However, to obtain our goals, we add redundancy by repeatedly transmitting the packet over a time interval whose length is more than one unit. A schedule specifies for each packet the length of the time interval allocated for its transmission.

Definition 1 (Schedule $\mathcal{S}$) *A* schedule *is a sequence of random variables $\{X_1, X_2, \dots\}$, $X_i \geq 0$, such that $X_i + 1$ specifies the length of the interval allocated for packet i.*

A schedule $\mathcal{S} = \{X_1, X_2, \dots\}$ can also be defined by its *transmission sequence* $\{T_1, T_2, \dots\}$, where T_n represents the beginning of the n^{th} interval, that is, $T_1 = 0$ and $T_n = \sum_{i=1}^{n-1} X_i + n - 1$ for all $n > 1$.

Let $\mathcal{S}$ be a schedule, and suppose that a client request is placed at time t. Also, let n be the *current* interval, i.e., the interval for which it holds that $T_n \leq t < T_{n+1}$. The waiting time depends on the amount of time remaining in the current interval, i.e., $T_{n+1} - t$. Specifically, if $T_{n+1} - t \geq 1$ then the client request can be satisfied by listening to the current interval, hence the waiting time is zero. Otherwise, the client must wait until the beginning of the next interval, hence its waiting time is $T_{n+1} - t$.

Definition 2 (Waiting Time, $WT(\mathcal{S}, t)$) *The* Waiting Time *$WT(\mathcal{S}, t)$ for a request at time t using a schedule $\mathcal{S} = \{X_1, X_2, \dots\}$ is defined as follows.*

$$WT(\mathcal{S}, t) = \begin{cases} T_{n+1} - t & \textbf{if} \quad T_{n+1} - t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where n is the current interval, i.e., the interval for which it holds that $T_n \leq t < T_{n+1}$.

The staleness captures the age of the information delivered to the client. As before, for a client request at time t let n be the current interval. The staleness depends on both the amount of time that has passed since the beginning of the current interval, i.e., $t - T_n$, and the amount of time remaining in the current interval, i.e., $T_{n+1} - t$. Specifically, if $T_{n+1} - t \geq 1$ then the client request can be satisfied by listening to the current interval, hence the staleness is $t - T_n$. Otherwise, the client must wait to the beginning of the next interval, and the information it receives will be fresh, i.e., the staleness will be zero.

Definition 3 (Staleness, $ST(\mathcal{S}, t)$) *The* Staleness *$ST(\mathcal{S}, t)$ for a request at time t using a schedule $\mathcal{S} = \{X_1, X_2, \dots\}$ is defined as follows.*

$$ST(\mathcal{S}, t) = \begin{cases} 0 & \textbf{if} \quad T_{n+1} - t < 1 \\ t - T_n & \text{otherwise.} \end{cases} \tag{2}$$

where n is the current interval, i.e., the interval for which it holds that $T_n \leq t < T_{n+1}$.

A few remarks are in place. First notice that as the schedule $\mathcal{S}$ is a random process, both $WT(\mathcal{S}, t)$ and $ST(\mathcal{S}, t)$ are random variables. Secondly, notice that the schedule $\mathcal{S}$ as defined above is closely related to so called *renewal* processes (e.g., [4]). Moreover, using the notation above, both $t - T_n$ and $T_{n+1} - t$ are random variables well studied in the theory of renewal processes. Nevertheless, to the best of our knowledge, the questions of our interest (regarding the worst-case expected value of $WT(\mathcal{S}, t)$ and $ST(\mathcal{S}, t)$) have not been addressed in the literature. Furthermore, notice the duality between the notion of staleness and waiting time. Namely, for any specific value of t it is the case that exactly one of the two is non-zero. This duality is not only syntactic but also intuitive, as the design of schedules with very low waiting time will necessarily have a high value of staleness (and visa versa). Finally, notice that the staleness is defined to be zero if $T_{n+1} - t < 1$. This follows from the fact that although the client may wait for information, the information (when received) will be "fresh".

Expected staleness and waiting time in the adaptive setting

In this section we define the expected staleness and waiting time for adaptive adversarial clients. The behavior of an adaptive adversarial client at time t depends on the *history* of the schedule up to time t. Therefore, for such clients we condition the probability distribution of a given random schedule $\mathcal{S}$ on the history of $\mathcal{S}$ up to time t. Intuitively, the history of a schedule can be described by the lengths of the intervals transmitted up to time t.

Definition 1 *A history* $H = (t, x_1, x_2, \ldots, x_l)$ *of a random schedule* $\mathcal{S} = \{X_1, X_2, \ldots\}$ *at time* t *is the event in which (a) For all* $i, 1 \le i \le l$*, it holds that* $X_i = x_i$*; and (b)* $\sum_{i=1}^{l} X_i + l \le t < \sum_{i=1}^{l+1} X_i + l + 1$.

In other words, $H = (t, x_1, x_2, \ldots, x_l)$ is the event in which (a) For the first l random variables in $\mathcal{S}$ it holds that $X_i = x_i$, and (b) The number of intervals that are completely broadcasted up to time t is l.

We assume that adaptive clients know at any time t the history H of a particular invocation of the schedule $\mathcal{S}$ (at time t). Thus, adaptive clients (almost always) have more information about the future behavior of $\mathcal{S}$ than oblivious clients. In particular, adaptive clients may base their future requests on the schedule $\mathcal{S}$ conditioned on the event H.

Formally, let H be a history event. H is said to be admissible if the probability that it occurs is non-zero. For admissible histories H, let $\mathcal{S}|H$ be the schedule obtained by conditioning $\mathcal{S}$ on the event H. Notice that $\mathcal{S}|H$ is also a (typically random) schedule. For any request time t, our objective is to obtain bounds on the expected waiting time and staleness of a schedule *no matter* what the viewed history of the schedule was before the request. Accordingly, the maximum expected staleness and waiting time for adaptive clients are defined as follows: $MWT(\mathcal{S}) = \sup_{H,t} E[WT(\mathcal{S}|H, t)]$; $MST(\mathcal{S}) = \sup_{H,t} E[ST(\mathcal{S}|H, t)]$. Here, the expectation is over the schedule distribution $\mathcal{S}|H$, and we are maximizing over admissible history events H.[1]

I.i.d. schedules

In the reminder of this study we focus on schedules $\mathcal{S} = \{X_1, X_2, \ldots\}$ in which all random variables X_i are independent and identically distributed (i.i.d.), that is $X_i = X$ for all i. Such schedules are referred to as i.i.d. schedules. As show below, for any schedule $\mathcal{S}$ there exists an i.i.d. schedule $\mathcal{S}'$ which is at least as good as $\mathcal{S}$, i.e., $MWT(\mathcal{S}') \le MWT(\mathcal{S})$ and $MST(\mathcal{S}') \le MST(\mathcal{S})$. This implies that our i.i.d. assumption does not result in any loss of generality.

We observe that if $\mathcal{S}$ is an i.i.d. schedule, then a history $H = (t, x_1, x_2, \ldots, x_n)$ of $\mathcal{S}$ at time t can be summarized by a single parameter $\tau = t - \sum_{i=1}^{n}(x_i + 1)$. Here, τ is the time that passed since the beginning of the current interval. Indeed, consider the schedule $\mathcal{S}|H$. Since the length of the current interval is at least τ, it is distributed according to the random variable $X|H = X\ |\{X + 1 > \tau\}$. The length of any subsequent interval in $\mathcal{S}|H$ is distributed according to the random variable X.

We define $EWT(S, \tau)$ and $EST(S, \tau)$ to be the expected waiting time and staleness of a request that arrives τ units of time after the beginning of the current transmitted interval, where the expectation is taken over the distribution of the schedule S, conditioned on the event that the current interval has length at least τ time units. From the above discussion it follows that for any i.i.d. schedule $\mathcal{S}$ with history $H = (t, x_1, x_2, \ldots, x_n)$ it holds that $E[WT(\mathcal{S}|H, t)] = EWT(S, \tau)$, where $\tau = t - \sum_{i=1}^{n}(x_i + 1)$. Further, the maximum expected waiting time and staleness of the schedule $\mathcal{S}$ are $MWT(\mathcal{S}) = \sup_\tau EWT(S, \tau)$ and $MST(\mathcal{S}) = \sup_\tau EST(S, \tau)$, respectively. These expressions will be used throughout our work.

An i.i.d. schedule $\mathcal{S} = \{X, X, \ldots\}$ can be completely described by the cumulative distribution function (c.d.f.) $F(x)$ of X, i.e., $F(x) = P(X \le x)$. Accordingly, in this paper the terms "schedule" and "distribution" are used interchangeably. We say that a function $F(x)$, and in turn a schedule $\mathcal{S}$, has support T if for $x \ge T$ it holds that $F(x) = 1$. We also denote $F^-(x) = P(X < x)$.

We proceed to prove that i.i.d. schedules are at least as good as general schedules. Namely, we show that any random schedule $\mathcal{S} = \{X_1, X_2, \ldots\}$ in which the random variables X_i are not necessarily i.i.d. and in which it may be the case that X_i depends on X_j for $j < i$, can be reduced to a random schedule $\mathcal{S}' = \{X_1', X_2', \ldots\}$ in which X_i' are i.i.d. Specifically, $\mathcal{S}'$ will have worst case expected waiting time and staleness no larger than $\mathcal{S}$. The reduction is rather simple and considers the schedule $\mathcal{S}'$ in which for each i, X_i' is set to be X_1 (of the schedule $\mathcal{S}$). Correctness follows from the fact that the expected waiting time and staleness of a query placed at time t depends exclusively on the distribution of the current interval that t is assumed to fall into. Thus considering only a single random variable X_1 which is repeated many times will yield an overall worst case waiting time and staleness no larger than that obtained when considering many different random variables.

[1]We have ignored a slightly subtle issue regarding the ability of an adversary to place a request at time t based on the information gained on the schedule up to time t' which is *strictly less* that t. In our setting (worst-case analysis), it is not hard to verify that such a scenario can be ignored w.l.o.g.

Problem formulation

In this paper we investigate the problem of finding schedules $\mathcal{S}$ which guarantee both low worst case expected waiting time and low worst case expected staleness in the presence of an adaptive adversary. We identify a tradeoff between the attainable worst case expected waiting time of a schedule and its worst case expected staleness. The main problem considered in this paper can be formulated as follows.

Problem OS *(Optimum Schedule) Given a staleness constraint s, find a schedule $\mathcal{S}$ whose worst-case expected waiting time $MWT(\mathcal{S})$ is minimal subject to the staleness constraint $MST(\mathcal{S}) \leq s$.*

We denote the minimum worst-case expected waiting time obtainable under the staleness constraint s by $OPT(s)$.

4 Maximum waiting time and staleness

Let X be a random variable, and let F be its cumulative distribution function. In this section we represent the expected waiting time and staleness yielded by an i.i.d. schedule $\mathcal{S}$ defined by X in terms of F.

Theorem 1 *Let $\mathcal{S}$ be an i.i.d. schedule with distribution $F(x)$. Fix $\tau \geq 0$ such that $F(\tau - 1) < 1$. Then, the expected waiting time $EWT(\mathcal{S}, \tau)$ of a request that arrives τ units of time after the beginning of the current interval is*

$$EWT(\mathcal{S}, \tau) = \begin{cases} F^{-}(\tau) - \int_0^{\tau} F(x)dx & \textbf{if} \quad \tau < 1 \\ \frac{F^{-}(\tau) - \int_{\tau-1}^{\tau} F(x)dx}{1 - F(\tau - 1)} & otherwise. \end{cases} \tag{3}$$

Proof: Consider any history event H at time t in which the length of the current transmitted interval is τ. By definition of $EWT(\mathcal{S}, \tau)$, it holds that $EWT(\mathcal{S}, \tau) = E[WT(\mathcal{S}|H, t)]$, where $\mathcal{S}|H$ is the schedule obtained by conditioning $\mathcal{S}$ on H. Since the length of the current interval in $\mathcal{S}|H$ is more than τ, its length is distributed according to the random variable $\hat{X} = X|H = X\ |\{X + 1 > \tau\}$. Since $\mathcal{S}$ is an i.i.d. schedule, the length of any subsequent interval in $\mathcal{S}|H$ is distributed according to the random variable X.

First, we determine the cumulative distribution function of $\hat{F}(x)$ of $\hat{X}$, i.e., $\hat{F}(x) = P(\hat{X} \leq x)$. We consider the two following cases:

1. $\tau < 1$. Then, since $X + 1 > \tau$ it holds that $\hat{X} = X$, that is $\hat{X}$ is distributed according to the probability distribution function $F(x)$, i.e., $\hat{F}(x) = F(x)$.

2. $\tau \geq 1$ Then, $\hat{X} = X \mid \{X + 1 > \tau\}$, hence $\hat{X}$ is distributed according to the following distribution:

$$\hat{F}(x) = \begin{cases} 0 & \textbf{if} \quad x \leq \tau - 1 \\ \frac{F(x) - F(\tau-1)}{1 - F(\tau - 1)} & otherwise. \end{cases} \tag{4}$$

The waiting time $WT(\mathcal{S}|H, \tau)$ depends on the time remained in the current interval, i.e., $\hat{X} + 1 - \tau$. We consider the two following cases:

1. **Case 1:** $\hat{X} - \tau + 1 \geq 1$ or $\hat{X} \geq \tau$. In this case the remaining time in the interval is one unit of time or more. Thus, according to Equation (1), the waiting time is zero, i.e., $WT(\mathcal{S}|H, \tau) = 0$.

2. **Case 2:** $\hat{X} + 1 - \tau < 1$ or $\hat{X} < \tau$. In this case, the client has to wait to the beginning of the next interval. According to Equation (1), the waiting time in this case is $WT(\mathcal{S}|H, \tau) = \hat{X} + 1 - \tau$.

We proceed to derive the expression for the expected waiting time $EWT(\mathcal{S}, \tau)$. We begin with the case $\tau < 1$ and identify the distribution $F_{WT}(x)$ of $WT(\mathcal{S}, \tau)$. We consider the following cases:

1. x=0. We note that the waiting time is zero if and only if $\hat{X} \geq \tau$. Thus, $F_{WT}(0) = P(WT(\mathcal{S}|H,\tau) = 0) = P(\hat{X} \geq \tau) = 1 - \hat{F}^{-}(\tau) = 1 - F^{-}(\tau)$.
2. $0 < x < 1-\tau$. We note that for $\tau < 1$ the waiting time is either zero or at least $1-\tau$. Hence for $0 < x < 1-\tau$ it holds that $F_{WT}(x) = F_{WT}(0) = 1 - F^{-}(\tau)$.
3. $1-\tau \leq x < 1$. Then, $F_{WT}(x) = P(WT(\mathcal{S}|H,\tau) \leq x) = P(1-\tau \leq WT(\mathcal{S}|H,\tau) \leq x) + P(WT(\mathcal{S}|H,\tau) < 1-\tau) = P(1-\tau \leq \hat{X}+1-\tau \leq x) + F_{WT}(0) = P(0 \leq \hat{X} \leq x+\tau-1) + F_{WT}(0) = 1 + F(x+\tau-1) - F^{-}(\tau)$.

Notice that for $x \geq 1$, $F_{WT}(x) = 1$. We are ready to compute the expected waiting time $EWT(\mathcal{S},\tau)$.

$$EWT(\mathcal{S},\tau) = \int_0^\infty (1 - F_{WT}(x))dx = F^{-}(\tau) - \int_{1-\tau}^{1} F(x+\tau-1)dx = F^{-}(\tau) - \int_0^\tau F(x)dx$$

Now we consider the case in which $\tau \geq 1$. There are two possible cases.

1. $x = 0$. In this case $F_{WT}(0) = P(\hat{X} \geq \tau) = 1 - \hat{F}^{-}(\tau) = 1 - \frac{F^{-}(\tau) - F(\tau-1)}{1-F(\tau-1)}$.
2. $0 < x < 1$. In this case $F_{WT}(x) = P(WT(\mathcal{S}|H,\tau) \leq x) = P(0 < WT(\mathcal{S}|H,\tau) \leq x) + P(WT(\mathcal{S}|H,\tau) = 0) = P(0 < \hat{X} + 1 - \tau \leq x) + F_{WT}(0) = P(\tau - 1 < \hat{X} \leq x + \tau - 1) = \hat{F}(\tau + x - 1) - \hat{F}(\tau - 1) + F_{WT}(0) = \frac{F(\tau+x-1) - F(\tau-1)}{1-F(\tau-1)} + F_{WT}(0)$.

We compute now the expected waiting time $EWT(\mathcal{S},\tau)$:

$$EWT(\mathcal{S},\tau) = \int_0^\infty (1 - \hat{F}(x))dx = \frac{F^{-}(\tau) - \int_0^1 F(\tau+x-1)dx}{1-F(\tau-1)} = \frac{F^{-}(\tau) - \int_{\tau-1}^{\tau} F(x)dx}{1-F(\tau-1)}.$$

■

Remark 1 For the purpose of computing the *maximum* waiting time, we can use the following expression instead of (3) (which allows us to consider only the function F without considering the function F^{-}).

$$EWT(\mathcal{S},\tau) = \begin{cases} F(\tau) - \int_0^\tau F(x)dx & \textbf{\textit{if}} \quad \tau < 1 \\ \frac{F(\tau) - \int_{\tau-1}^{\tau} F(x)dx}{1-F(\tau-1)} & otherwise. \end{cases} \tag{5}$$

Namely, it can be seen that $MWT(\mathcal{S}) = \sup_\tau EWT(S,\tau)$ is identical no matter which definition of $EWT(\mathcal{S},\tau)$ is used ((3) or (5)). Correctness follows from the properties of F. As Equation 5 is easier to deal with, we use this equation for computing the maximum waiting time throughout our work.

Theorem 2 *Let $\mathcal{S}$ be an i.i.d. schedule. Fix $\tau \geq 0$ such that $F(\tau-1) < 1$. The expected staleness $EST(\mathcal{S},\tau)$ of a request that arrives τ units of time after the beginning of the current interval is*

$$EST(\mathcal{S},\tau) = \begin{cases} \tau(1 - F^{-}(\tau)) & \textbf{if} \quad \tau < 1 \\ \tau\left(\frac{1-F^{-}(\tau)}{1-F(\tau-1)}\right) & otherwise. \end{cases} \tag{6}$$

Proof: Consider any history event H at time t in which the length of the current transmitted interval is τ. By definition of $EWT(\mathcal{S},\tau)$ it holds that $EWT(\mathcal{S},\tau) = E[WT(\mathcal{S}|H,t)]$, were $\mathcal{S}|H$ is the schedule $\mathcal{S}$ conditioned on H. We observe, that the length of the current interval in $\mathcal{S}|H$ is more than τ and its length is distributed according to the random variable $\hat{X} = X|H = X\,|\{X+1 > \tau\}$. Let $\hat{F}(x) = P(\hat{X} \leq x)$. We consider the two following cases:

1. **Case 1:** The next interval begins in one unit of time or more, that is $\hat{X} \geq \tau$. According to Equation (2), the staleness in this case is $ST(\mathcal{S}|H, \tau) = \tau$.
2. **Case 2:** The next interval begins in less than one unit of time, that is $\hat{X} < \tau$. According to Equation (1) the staleness in this case is zero, i.e., $ST(\mathcal{S}|H, \tau) = 0$.

It follows that $EST(\mathcal{S}, \tau) = \tau P(\hat{X} \geq \tau)$. Again, we consider two cases. Using the expressions derived in the proof of Theorem 1 for $\hat{F}(x)$ we conclude that

1. If $\tau < 1$, then

$$EST(\mathcal{S}, \tau) = \tau(1 - \hat{F}^{-}(\tau)) = \tau(1 - F^{-}(\tau)) \tag{7}$$

2. If $\tau \geq 1$ then

$$EST(\mathcal{S}, \tau) = \tau(1 - \hat{F}^{-}(\tau)) = \tau \left(1 - \frac{F^{-}(\tau) - F(\tau - 1)}{1 - F(\tau - 1)}\right) = \tau \left(\frac{1 - F^{-}(\tau)}{1 - F(\tau - 1)}\right). \tag{8}$$

■

Remark 2 Using the same ideas as in Remark 1, we can show that using the following equation for the purpose of computing the maximum staleness is equivalent to using Equation (6).

$$EST(\mathcal{S}, \tau) = \begin{cases} \tau(1 - F(\tau)) & \textbf{\textit{if}} \quad \tau < 1 \\ \\ \tau \left(\frac{1 - F(\tau)}{1 - F(\tau - 1)}\right) & otherwise, \end{cases} \tag{9}$$

As Equation (9) is easier to deal with, we use this equation for computing the maximum staleness throughout our work.

5 Approximation algorithm for Problem OS

In this section we turn to study arbitrary values of s. We present an approximation algorithm for Problem OS. The algorithm receives as input, a staleness constraint s and an approximation ratio ε and returns a schedule $\mathcal{S}$ whose worst case waiting time is at most $OPT(s) + \varepsilon$.

We use two approximations techniques. First, we show that for any $\varepsilon_1 > 0$ there exists a schedule $\mathcal{S}_1$ such that $MST(\mathcal{S}_1) \leq s$, $MWT(\mathcal{S}_1) \leq OPT(s) + \varepsilon_1$ and the support of $\mathcal{S}_1$ is at most $\frac{s}{\varepsilon_1}$. In other words, the optimal distribution can be approximated by a distribution with bounded support. Second, we show that for any $\varepsilon_2 > 0$ the schedule $\mathcal{S}_1$ can be approximated by a schedule $\mathcal{S}_2$ whose distribution is a piecewise-constant function that includes at most $\frac{s}{\varepsilon_1 \varepsilon_2}$ segments. This schedule satisfies the staleness constraint, i.e., $MST(\mathcal{S}_2) \leq s$, and its maximum waiting time is more than that of $\mathcal{S}_1$ by at most ε_2 , i.e., $MWT(\mathcal{S}_2) \leq MWT(\mathcal{S}_1) + \varepsilon_1 \leq OPT(s) + \varepsilon_1 + \varepsilon_2$. Moreover, we present a Linear Program (LP) that allows to compute $\mathcal{S}_2$ in time $O(poly(\frac{s}{\varepsilon_2}))$. As a result, for any $\varepsilon > 0$ we can compute a schedule that satisfies the staleness constraint s and whose maximal waiting time is at most $OPT(s) + \varepsilon$. Indeed, by setting $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2}$, we ensure that the schedule $\mathcal{S}_2$ satisfies the above requirements.

5.1 Approximation by a fixed support distribution

In this section we prove that an optimal solution $F(x)$ to Problem OS can be approximated by a distribution $\hat{F}(x)$ that has a bounded support.

Lemma 1 *Let s be a staleness constraint and let $\varepsilon_1 > 0$ be an approximation parameter. Then, there exists a schedule $\hat{\mathcal{S}}$ with support bounded by $\frac{s}{\varepsilon_1}$ such that $MST(\hat{\mathcal{S}}) \leq s$ and $MWT(\hat{\mathcal{S}}) \leq OPT(s) + \varepsilon_1$.*

Proof: Let $\mathcal{S}$ be the optimum i.i.d. schedule that satisfies the staleness constraint of value s and let $F(x)$ be its distribution. We define a new distribution $\hat{F}(x)$ as follows:

$$\hat{F}(x) = \begin{cases} F(x) & \text{if} \quad x < \frac{s}{\varepsilon_1} \\ 1 & \text{if} \quad x \geq \frac{s}{\varepsilon_1} \end{cases} \tag{10}$$

Let $\hat{\mathcal{S}}$ be the schedule defined by $\hat{F}(x)$. It is not hard to verify that $\hat{F}(x)$ satisfies the staleness constraint s.

We proceed to prove that the expected worst case waiting time for $\hat{F}(x)$ is at most $OPT(s) + \varepsilon_1$. Let $\tau^* = \frac{s}{\varepsilon_1}$. Clearly, for $\tau \leq \tau^*$, it holds that $EWT(\hat{\mathcal{S}}, \tau) = EWT(\mathcal{S}, \tau)$. Suppose that $\tau^* \geq 1$ and let us compute $EWT(\hat{\mathcal{S}}, \tau)$ for $\tau^* < \tau \leq \tau^* + 1$. By Equation (5),

$$EWT(\hat{\mathcal{S}}, \tau) = \frac{\hat{F}(\tau) - \int_{\tau-1}^{\tau} \hat{F}(x)dx}{1-\hat{F}(\tau-1)} = \frac{1 - \int_{\tau-1}^{\tau^*} F(x)dx - \int_{\tau^*}^{\tau} dx}{1-F(\tau-1)} = \frac{F(\tau) + (1-F(\tau)) - \int_{\tau-1}^{\tau} F(x)dx - \int_{\tau^*}^{\tau}(1-F(x))dx}{1-F(\tau-1)} \leq$$
$$\leq EWT(\mathcal{S}, \tau) + \frac{1-F(\tau)}{1-F(\tau-1)}. \tag{11}$$

Since $F(\tau)$ satisfies the staleness constraint, Equation (9) implies that $\tau \frac{1-F(\tau)}{1-F(\tau-1)} \leq s$. We conclude that $EWT(\hat{\mathcal{S}}, \tau) \leq EWT(\mathcal{S}, \tau) + \frac{s}{\tau} \leq EWT(\mathcal{S}, \tau) + \frac{s}{\tau^*} = EWT(\mathcal{S}, \tau^*) + \varepsilon_1$.

The proof for the case in which $\tau^* < 1$ follows similar lines. ■

5.2 Approximation algorithm

In this section we show that an optimal solution to Problem OS can be approximated by a step function. Let $\mathcal{S}$ be the schedule whose existence is guaranteed by Lemma 1. We denote by $F(x)$ the distribution of $\mathcal{S}$. The support of $F(x)$ is bounded by $T = \frac{s}{\varepsilon_1}$. We show that for any $\varepsilon_2 > 0$ there exists a distribution $\hat{F}(x)$ which is a step (piecewise constant) function on $N = \frac{2}{\varepsilon_2}T$ intervals, that satisfies the staleness constraint s and whose maximum waiting time exceeds that of $\mathcal{S}$ by at most ε_2 (in what follows we assume that both T and N are integers, minor modifications are needed in the proof otherwise). Thus, the distribution $\hat{F}(x)$ yields a schedule $\hat{\mathcal{S}}$ such that $MST(\hat{\mathcal{S}}) \leq s$ and $MWT(\hat{\mathcal{S}}) \leq MWT(\mathcal{S}) + \varepsilon_2 \leq OPT(s) + \varepsilon_1 + \varepsilon_2 \leq OPT(s) + \varepsilon$. Further, we show that such a function can be identified by solving a Linear Program of N variables. We note that $N = O(\frac{s}{\varepsilon_1 \varepsilon_2})$.

We represent a step function $\hat{F}(x)$ by a set of discontinuity points $\{(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)\}$:

$$\hat{F}(x) = \begin{cases} y_0 & \textbf{if} \quad x_0 \leq x < x_1 \\ y_1 & \textbf{if} \quad x_1 \leq x < x_2 \\ \cdots & \\ y_N & \textbf{if} \quad x \geq x_N. \end{cases} \tag{12}$$

In order for a function $\hat{F}(x)$ to be a distribution, we require that $y_0 \geq 0$, $y_N = 1$ and that $y_i \geq y_{i-1}$ (for $i = 1 \ldots N$). We choose the x-coordinates of discontinuity points to be uniformly distributed between 0 and T, i.e., $x_i = \frac{T}{N}i$. For clarity of presentation we denote by $M = \frac{N}{T} = \frac{2}{\varepsilon_2}$ the number of discontinuity points in an interval of length one unit. Further, we denote by $\delta = \frac{1}{M}$ the time between two subsequent discontinuity points.

We are interesting in finding a distribution $\hat{F}(x)$ that satisfies the staleness constraint s and has a maximum waiting time of at most c, where c is an approximation for $OPT(s)$. In order for the function $\hat{F}(x)$ to satisfy the two above conditions, it suffices that the values of $\{y_i\}_{i=0}^{N}$ satisfy the following set of linear inequalities:

$$\begin{array}{ll} y_0 \leq c & \\ y_i - \delta \sum_{j=0}^{i-1} y_j \leq c & \textbf{for } i = 1, \cdots, M-1; \\ y_i - \delta \sum_{j=i-M}^{i-1} y_j \leq c(1 - y_{i-M}) & \textbf{for } i = M, \cdots, N; \\ x_{i+1}(1 - y_i) \leq s & \textbf{for } i = 1, \cdots, M-1; \\ x_{i+1}(1 - y_i) \leq s(1 - y_{i-M}) & \textbf{for } i = M, \cdots, N-1; \\ y_i \geq y_{i-1} & \textbf{for } i = 2, \cdots, N; \\ y_0 \geq 0, \; y_N = 1. & \end{array} \tag{13}$$

The first three lines ensure that the maximum waiting time of $\hat{F}(x)$ is bounded by c. The next two lines ensure that the staleness of $\hat{F}(x)$ is at most s. The purpose of the last two lines is to ensure that the function $\hat{F}(x)$ is a cumulative distribution function.

A step function that satisfies given staleness and maximum waiting time constraints, s and c, can be found by solving the Linear Program defined by (13). This leads to an approximation algorithm for Problem OS. The algorithm finds the value of c that is sufficiently close to $OPT(s)$ by performing binary search (over c) on the interval $[0, 1]$. Specifically, we begin by setting a lower bound LB to 0 and an upper bound UB to be 1. At each iteration we attempt to solve the LP for $c = \frac{UB+LB}{2}$. If there is no feasible solution to LP, we set $LB = c$, otherwise we set $UB = c$. The algorithm continues until $UB - LB \le \frac{\varepsilon_2}{4}$. The algorithm returns the distribution function that corresponds to the last solution of the LP.

5.3 Analysis of the approximation algorithm

We proceed to prove the correctness of the approximation algorithm (namely that there exists a step function *close* to being optimal).

Lemma 2 *Let $\mathcal{S}$ be the schedule whose existence is guaranteed by Lemma 1 and $F(x)$ the distribution of $\mathcal{S}$. Then, the LP defined by (13) has a feasible solution for $c = MWT(\mathcal{S}) + \frac{\varepsilon_2}{2}$.*

Proof: We set $y_i = F(x_{i+1})$ for $0 \le i < N$ and $y_N = 1$ and show that $\{y_i\}_{i=0}^N$ is a feasible solution to the LP, i.e., it satisfies inequalities (13) with $s = MST(\mathcal{S})$ and $c = MWT(\mathcal{S}) + \frac{\varepsilon_2}{2}$ (this implies that the distribution defined by the pairs (x_i, y_i) has $MST = s$ and $MWT = c$).

This paragraph deals with line 1 of LP (13). Since $EWT(\mathcal{S}, x_1) \le MWT(\mathcal{S})$ and since $EWT(\mathcal{S}, x_1) \ge F(x_1) - \delta F(x_1)$ it holds that $y_0 = F(x_1) \le MWT(\mathcal{S}) + F(x_1)\delta$. Further, we note that $F(x_1) \le 1$ and $\delta \le \frac{\varepsilon_2}{2}$. Thus, we conclude that $y_0 \le MWT(\mathcal{S}) + \frac{\varepsilon_2}{2} = c$.

We proceed to prove that the inequalities defined in line 2 of (13) hold. Fix $i, 1 \le i \le M - 1$. We note that $EWT(\mathcal{S}, x_{i+1}) \ge F(x_{i+1}) - \delta \sum_{j=1}^{i+1} F(x_j)$. Since $EWT(\mathcal{S}, x_{i+1}) \le MWT(\mathcal{S})$, it holds that $F(x_{i+1}) - \delta \sum_{j=1}^{i+1} F(x_j) \le MWT(\mathcal{S})$, and, in turn, $y_i - \delta \sum_{j=0}^{i} y_j \le MWT(\mathcal{S})$. We conclude that $y_i - \delta \sum_{j=0}^{i-1} y_j \le MWT(\mathcal{S}) + F(x_{i+1})\delta \le MWT(\mathcal{S}) + \delta \le MWT(\mathcal{S}) + \frac{\varepsilon_2}{2} = c$.

Now, we turn to the inequalities defined in line 3 of (13). Fix $i, M \le i \le N$. We note that $EWT(\mathcal{S}, x_{i+1}) \ge \frac{F(x_{i+1}) - \delta \sum_{j=i-M+2}^{i+1} F(x_j)}{1 - F(x_{i-M+1})}$, where $x_i = \delta i$. Since $EWT(\mathcal{S}, x_{i+1}) \le MWT(\mathcal{S})$, it holds that

$$\frac{F(x_{i+1}) - \delta \sum_{j=i-M+2}^{i+1} F(x_j)}{1 - F(x_{i-M+1})} \le MWT(\mathcal{S}),$$

and, in turn,

$$\frac{y_i - \delta \sum_{j=i-M+1}^{i} y_j}{1 - y_{i-M}} \le MWT(\mathcal{S}).$$

We conclude that $\frac{y_i - \delta \sum_{j=i-M}^{i-1} y_j}{1 - y_{i-M}} \le MWT(\mathcal{S}) + \delta \cdot \frac{y_i - y_{i-M}}{1 - y_{i-M}} \le MWT(\mathcal{S}) + \delta \frac{1 - y_{i-M}}{1 - y_{i-M}} \le MWT(\mathcal{S}) + \delta \le c$.

Next, we turn to the inequalities defined in lines 4 and 5 of (13). Fix $i, 1 \le i \le M - 1$. Since $F(x)$ satisfies the staleness constraint it holds that $x_{i+1}(1 - F(x_{i+1})) \le s$. This implies that $x_{i+1}(1 - y_i) \le s$. Now, fix $i, M \le i \le N$. Again, since $F(x)$ satisfies the staleness constraint we have $x_{i+1} \frac{1 - F(x_{i+1})}{1 - F(x_{i-M+1})} \le s$. Thus, it follows that $x_{i+1}(1 - y_i) \le s(1 - y_{i-M})$.

The inequalities in line 6 are satisfied because $F(x)$ is a monotonically increasing function. Finally, the inequality in line 7 is satisfied because the distribution $F(x)$ has support T. Thus, $y_N = F(x_{N+1}) = F(\delta(N + 1)) = F((N + 1)\frac{T}{N}) = 1$. ■

We conclude by the following theorem.

Theorem 3 *The approximation algorithm presented in Section 5.2 identifies, for a given staleness constraint s and approximation ratio $\varepsilon > 0$, a schedule $\hat{\mathcal{S}}$ that satisfies $MST(\hat{\mathcal{S}}) \le s$ and $MWT(\hat{\mathcal{S}}) \le OPT(s) + \varepsilon$. The computational complexity of the algorithm is polynomial in $\frac{s}{\varepsilon}$.*

Proof: We prove that during the execution of the algorithm it holds that $LB \leq OPT(s) + \frac{3\varepsilon}{4}$ and $UB \geq OPT(s)$.

Lemma 1 implies that there exists a schedule $\hat{\mathcal{S}}$ with support bounded by $\frac{s}{\varepsilon_1}$ such that $MST(\hat{\mathcal{S}}) \leq s$ and $MWT(\hat{\mathcal{S}}) \leq OPT(s) + \varepsilon_1 = OPT(s) + \frac{\varepsilon}{2}$. By Lemma 2, the LP defined by (13) has a feasible solution when $c \geq OPT(s) + \frac{\varepsilon}{2} + \frac{\varepsilon_2}{2} = OPT(s) + \frac{3\varepsilon}{4}$. Thus, if the algorithm does not find a solution to the LP for a given c, then it holds that $c < OPT(s) + \frac{3\varepsilon}{4}$. Hence, LB remains valid during the execution of the algorithm. On the other hand, if the algorithm finds a feasible solution $\hat{\mathcal{S}}$ to the LP, then, $MST(\hat{\mathcal{S}}) \leq s$ and $MWT(\hat{\mathcal{S}}) \leq c$. Thus, in this case, $OPT(s) \leq c$, hence UB remains valid during the execution of the algorithm.

When the algorithm stops it holds that $UB - LB \leq \frac{\varepsilon}{4}$. Hence, $UB \leq OPT(s) + \frac{3\varepsilon}{4} + \frac{\varepsilon}{4} = OPT(s) + \varepsilon$. We conclude that the algorithm returns a schedule $\hat{\mathcal{S}}$ that satisfies $MST(\hat{\mathcal{S}}) \leq s$ and $MWT(\hat{\mathcal{S}}) \leq OPT(s) + \varepsilon$.

We proceed to prove the computational complexity of the algorithm. Notice that our LP has $O(\frac{s}{\varepsilon^2})$ variables and the same order of equations. Thus, solving the LP requires time polynomial in $\frac{s}{\varepsilon}$ (e.g., [9]). Since the LP is solved $O(\log \frac{1}{\varepsilon})$ times, the total computational complexity of our algorithm remains $poly\left(\frac{s}{\varepsilon}\right)$. ■

6 Optimal solution for small values of staleness

In this section we present an optimal solution for Problem OS for the special case in which the values of the staleness constraint are small. More specifically, let $C(s) = (1 - \frac{2s}{s+\sqrt{s(4+s)}})e^{-\frac{1}{2}(s+\sqrt{s(4+s)})}$ and let $c_{\max} = \sqrt{-\frac{1}{2}\text{ProductLog}(-\frac{2}{e^2})}$. We prove that for $s \in [0, C^{-1}(c_{\max})]$, $F(x) = \min\{1, C(s)e^x\}$ is an optimal solution for Problem OS with staleness constraint s, where $C^{-1}(x)$ is the inverse function of $C(s)$. The approximate values for $c_{\max}$ and $C^{-1}(c_{\max})$ are 0.450764 and 0.13206, respectively.

The proof includes several steps. First, we show that if the expected waiting time yielded by a schedule $\mathcal{S}$ is at most c, then it holds that $F(x) \leq ce^x$ for all $x < 1$, where $F(x)$ is the distribution of $\mathcal{S}$.

Lemma 3 *Let $\mathcal{S}$ be a schedule that satisfies the waiting time constraint c and let $F(x)$ be the distribution of $\mathcal{S}$. Then, for each $x \in [0, 1)$ it holds that $F(x) \leq ce^x$.*

Proof: By way of contradiction, assume that there exists a function $F(x)$ such that (1) $F(x)$ satisfies the waiting time constraint c (2) there exists a time $x < 1$ for which it holds that $F(x) > ce^x$.

Suppose that $F(x)$ is a continuous function. We note that $F(0) \leq c$, otherwise according to Equation (5), $MWT(\mathcal{S}) > c$, which contradicts our assumption. Thus, there exists a value $x < 1$ such that $F(x) = ce^x$. Let x^* be the smallest value that satisfies $F(x^*) = ce^{x^*}$. By Equation (5) $EWT(\mathcal{S}, x^*) = F(x^*) - \int_0^{x^*} F(y)dy$.

Since for each $x \in [0, x^*)$ it holds that $F(x) < ce^x$, we have $\int_0^{x^*} F(y)dy < \int_0^{x^*} ce^x dy = ce^{x^*} - c$. Thus, $EWT(\mathcal{S}, x^*) > ce^{x^*} - ce^{x^*} + c = c$, which contradicts the fact that $EWT(x) \leq c$ for each $x \in [0, 1)$.

If $F(x)$ is not a continuous function, a similar argument can be used. ■

Next, we observe that if the expected staleness yielded by a schedule $\mathcal{S}$ is at most s, then it holds that $F(x) \geq 1 - \frac{s}{x}$ for all $x < 1$, where $F(x)$ is the distribution of $\mathcal{S}$. Indeed, suppose that $F(x^*) < 1 - \frac{s}{x^*}$ for some $x^* \leq 1$, then, by Equation (9), $EST(\mathcal{S}, x^*) = x^*(1 - F(x^*)) > s$, a contradiction.

This observation, together with Lemma 3, allows to establish a lower bound on $OPT(s)$.

Lemma 4 *Let $\mathcal{S}$ be a schedule that satisfies a staleness constraint s. Then, the maximal waiting time of $\mathcal{S}$ is at least $C(s)$, i.e., $OPT(s) \geq C(s)$, where $C(s) = (1 - \frac{2s}{s+\sqrt{s(4+s)}})e^{-\frac{1}{2}(s+\sqrt{s(4+s)})}$.*

Proof: Let $F(x)$ be the distribution of $\mathcal{S}$ and let c be the worst case expected waiting time of $\mathcal{S}$. Fix a staleness constraint s. As we showed above, $F(x) \geq 1 - \frac{s}{x}$ for all $x < 1$. By Lemma 3 it also holds that $F(x) \leq ce^x$ for all $x < 1$. Thus, it must be the case that $1 - \frac{s}{x} \leq ce^x$, otherwise such a function $F(x)$ does not exist. Simple algebra shows that this is the case only when $c \geq (1 - \frac{2s}{s+\sqrt{s(4+s)}})e^{-\frac{1}{2}(s+\sqrt{s(4+s)})}$. ■

Next, we show that for $s \in [0, C^{-1}(c_{\max})]$, the lower bound proven in Lemma 4 is tight. Namely, we show that for $c \in [0, c_{\max}]$ the distribution $F(x) = \min\{1, C(s)e^x\}$ satisfies the staleness constraint s and has maximal waiting time $C(s)$.

Lemma 5 *Let $s \in [0, C^{-1}(c_{\max})]$. Then, the distribution $F(x) = \min\{1, C(s)e^x\}$ satisfies the staleness constraint s and has maximal expected waiting time $C(s)$, where $C(s) = (1 - \frac{2s}{s+\sqrt{s(4+s)}})e^{-\frac{1}{2}(s+\sqrt{s(4+s)})}$.*

Proof: First, we prove that $F(x)$ satisfies the waiting time constraint. We denote $c = C(s)$. By Equation (5), for $x < 1$ it holds that $EWT(\mathcal{S}, x) \leq F(x) - \int_0^x F(y)dy \leq c$. For $x \geq 1$, it is easy to see that the maximum waiting time is achieved at $x = 1$ and is equal to $\frac{-\ln c}{1-c} - 1$. It can be verified that for $c \in [0, \sqrt{-\frac{1}{2}\mathrm{ProductLog}(-\frac{2}{\mathrm{e}^2})}]$ it holds that $\frac{-\ln c}{1-c} - 1 \leq c$.

We proceed to show that the function $F(x) = \min\{1, C(s)e^x\}$, satisfies the staleness constraint s. By Equation (9), any distribution that satisfies $F(x) \geq 1 - \frac{s}{x}$ for $x \in [s, 1]$ and $F(x) = 1$ for $x \geq 1$ yields the expected staleness less that s. Thus, since $F(x) \geq 1 - \frac{s}{x}$ for $x \in [s, 1]$ and since for $s \in [0, C^{-1}(c_{\max})]$ it holds that $F(x) = 1$ for $x \geq 1$, we conclude that $F(x)$ satisfies the staleness constraint s. ■

Our results can be summarized by the following theorem.

Theorem 4 *Let $s \leq C^{-1}(c_{\max})$ be a staleness constraint. Then, $F(x) = \min\{1, C(s)e^x\}$ is an optimal solution for Problem OS, where $C(s) = (1 - \frac{2s}{s+\sqrt{s(4+s)}})e^{-\frac{1}{2}(s+\sqrt{s(4+s)})}$. In addition, $C(s)$ is the minimum possible worst case expected waiting time under the staleness constraint $s \leq C^{-1}(c_{\max})$.*

Proof: Follows from Lemmas 3-5. ■

7 Uniform Clients

In this section, we consider the tradeoff between the worst case expected waiting time and worst case expected staleness in the presence of uniform clients (i.e., whose request times are distributed uniformly over time).

The expected waiting time and staleness of such clients is defined over both the distribution of request times and the distribution of the schedule $\mathcal{S} = \{\mathcal{S}_\omega\}_{\omega \in \Omega}$.

$$\begin{aligned} EWT^{unif}(\mathcal{S}) &= E_{t,\omega}[WT(S_\omega, t)]; \\ EST^{unif}(\mathcal{S}) &= E_{t,\omega}[ST(S_\omega, t)]. \end{aligned} \tag{14}$$

As we shall see, in this case, random schedules do not have an advantage over deterministic ones. We begin by the following observation:

Lemma 6 *Let S be a deterministic schedule whose staleness is at most s, i.e., $EST^{unif}(S) \leq s$, and whose expected waiting time $EWT^{unif}(S)$ is minimal. Then, $EWT^{unif}(S) = \frac{1+s-\sqrt{2s+s^2}}{2}$.*

Proof: We first show that $EWT^{unif}(S) \geq \frac{1+s-\sqrt{2s+s^2}}{2}$. We denote by T_i the length of the time interval during which packet i is transmitted in schedule S. Then,

$$EWT^{unif}(S) = \lim_{n\to\infty} \frac{n}{2\sum_{i=1}^n T_i}$$

and

$$EST^{unif}(S) = \lim_{n\to\infty} \frac{\sum_{i=1}^n (T_i - 1)^2}{2\sum_{i=1}^n T_i}$$

Let $\alpha_n = \frac{n}{2\sum_{i=1}^n T_i}$ and $\beta_n = \frac{\sum_{i=1}^n (T_i-1)^2}{2\sum_{i=1}^n T_i}$. We note that for each n it holds that

$$\beta_n = \frac{\sum_{i=1}^n (T_i - 1)^2}{2\sum_{i=1}^n T_i} = \frac{\sum_{i=1}^n T_i^2 - 2\sum_{i=1}^n T_i + n}{2\sum_{i=1}^n T_i} \geq \frac{\frac{1}{n}(\sum_{i=1}^n T_i)^2 - 2\sum_{i=1}^n T_i + n}{2\sum_{i=1}^n T_i} = \frac{1}{4\alpha_n} - 1 + \alpha_n, \qquad (15)$$

where the inequality follows from the fact that $\sum_{i=1}^n x_i^2 \geq \frac{(\sum_{i=1}^n x_i)^2}{n}$, which, in turn, follows from Cauchy's inequality.

Simple algebra shows that (15) holds only when $\alpha_n \geq \frac{1+\beta_n-\sqrt{2\beta_n+\beta_n^2}}{2}$. Since $EWT^{unif}(S) = \lim_{n\to\infty} \alpha_n$ and $EST^{unif}(S) = \lim_{n\to\infty} \beta_n$ we have

$$EWT^{unif}(S) \geq \frac{1 + EST^{unif}(S) - \sqrt{2EST^{unif}(S) + EST^{unif}(S)^2}}{2} \geq \frac{1+s-\sqrt{2s+s^2}}{2}.$$

Second, we show that there exists a schedule S' such that $EST^{unif}(S) \leq s$ and $EWT^{unif}(S) = \frac{1+s-\sqrt{2s+s^2}}{2}$. Let S' be a schedule that transmits each packet over an interval of length $T = \frac{1+s+\sqrt{2s+s^2}}{2}$. Then, the expected waiting time $EWT^{unif}(S)$ of S is

$$EWT^{unif}(S) = \frac{1}{2T} = \frac{1}{2(1+s+\sqrt{2s+s^2})} = \frac{1+s-\sqrt{2s+s^2}}{2}.$$

It is also easy to verify that $EST^{unif}(S) = \frac{(T-1)^2}{2T} = s$. ■

The next lemma shows that deterministic schedules are just as good as random ones for uniform clients.

Lemma 7 *For any random schedule $\mathcal{S}$ there exists a deterministic schedule $\mathcal{S}'$ such that $EST^{unif}(\mathcal{S}') = EST^{unif}(\mathcal{S})$ and $EWT^{unif}(\mathcal{S}') \leq EWT^{unif}(\mathcal{S})$.*

Proof: The random schedule $\mathcal{S}$ is a probability distribution over deterministic schedules $\{S_\omega\}_{\omega\in\Omega}$. We denote $s = EST^{unif}(\mathcal{S})$ and $c = EWT^{unif}(\mathcal{S})$. Further, we denote by s_ω and c_ω the expected waiting time and staleness of the deterministic schedule S_ω. Then, according to Equation (14), $EWT^{unif}(\mathcal{S}) = c = E_\omega[c_\omega]$ and $EST^{unif}(\mathcal{S}) = s = E_\omega[s_\omega]$, where the expectations are taken over the probability distribution of the schedule $\mathcal{S}$.

Lemma 6 implies that

$$EWT^{unif}(\mathcal{S}) \geq E\left[\frac{1+s_\omega-\sqrt{2s_\omega+s_\omega^2}}{2}\right] \geq \frac{1+E[s_\omega]-\sqrt{2E[s_\omega]+E[s_\omega]^2}}{2} = \frac{1+s-\sqrt{2s+s^2}}{2},$$

where the last inequality follows from the convexity of the function $\frac{1+s-\sqrt{2s+s^2}}{2}$. By Lemma 6, there exists a deterministic schedule whose expected staleness and waiting time are s and $\frac{1+s-\sqrt{2s+s^2}}{2}$, respectively. This completes the proof of the lemma. ■

Lemmas 6 and 7 establish a tradeoff between staleness and waiting time for uniformly distributed clients.

8 Our Results

We used the optimal and approximation algorithm presented in the previous sections in order to compute the attainable values of worst-case waiting time for a broad range of staleness constraints. Our results establish a tradeoff between the staleness and waiting time of universal broadcast schedules. The tradeoff is depicted on Fig. 4 (series A). This tradeoff has a surprising behavior we refer to as the "knee" phenomenon: for small values of staleness (typically below 0.3) the minimum waiting time decreases drastically with only a minor increase in the staleness constraint; however, for large values of the staleness constraint (above 0.3), any increase in the staleness constraint results in only a minor decrease of waiting time. A direct result of the knee phenomenon is the existence of a schedule that has small maximum expected waiting time (0.31) and whose worst-case

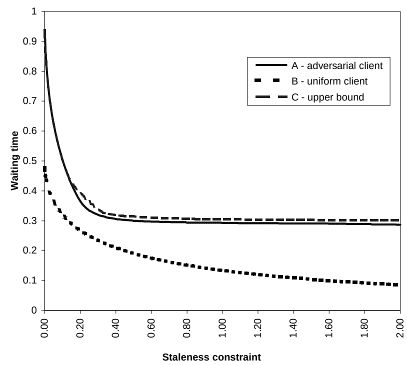

Figure 4: Tradeoff between staleness and waiting time for adaptive adversarial clients (A), uniform clients (B) and an upper bound on this tradeoff using the explicit distributions (C).

expected staleness is also small (at most 0.3). This point represents a reasonable tradeoff between waiting time and staleness. The corresponding schedule reduces the worst-case waiting time by 70% compared to a deterministic schedule while ensuring that the distributed information is up-to-date.

We also considered uniform clients (clients whose requests are assumed to be distributed uniformly over time). We study the staleness/waiting time tradeoff in this setting and find optimal schedules S (for arbitrary values of the staleness constraint). These are found explicitly (the governing distribution function F is presented analytically). Our results on uniform clients are depicted in Figure 4 (series B).

The study of analytical (closed form) approximate solutions to Problem OS gave rase to the following empirical observation. For arbitrary values of s, the distribution function $G_s(x) = 1 - \Gamma(s + a(s))\frac{s^{x+1-s-a(s)}}{\Gamma(x+1)}$ yields worst-case waiting time which is very close to optimum. Here $a(s)$ is a constant between 0 and 1, and Γ represents the standard Gamma function. The staleness/waiting time tradeoff of our schedules defined by F_s are depicted in Figure 4 (series C).

9 Conclusion

In this paper, we have studied the design of optimal schedules in asynchronous settings. We have defined the notion of staleness, and have presented a tight characterization of the obtainable waiting time given a staleness constraint. Our results are optimal for small values of staleness, and arbitrarily close to being optimal for general staleness values.

We would like to note that the model of universal discrete broadcast and the objective functions of waiting time and staleness studied in this work have applications in other fields as well. For example consider the problem of scheduling updates in a dynamic database. Namely, consider the scenario in which it takes one time unit to update a database with highly dynamic data, and during this time the database is inaccessible to clients. The objective in this case will be to schedule the updates as to minimize the time a client has to wait in order to obtain the queried data, and to ensure the data obtained is “fresh”.

Many questions in the setting of universal broadcasting remain open. First, in our framework we have assumed that the packets are transmitted over a channel without errors. However, one may consider lossy channels. In such a case changing the repetition encoding of packets to one that admits error correction seems to yield a good staleness/waiting time tradeoff. Second, one may consider multiple information sources as opposed to the single source studied in this work. The tools presented in this work can be used to study this extended model as well. Finally, the jamming problem must be rigorously studied and analyzed. Introducing randomness to the schedule and adversarial analysis are promising tools for designing broadcast schedules that have provable performance in the presence of jamming.

References

[1] A. Bar-Noy, R. Bhatia, J. Naor, and B. Schieber. Minimizing Service and Operation Costs of Periodic Scheduling. In *Proceedings of SODA*, pages 11–20, 1998.

[2] A. Bar-Noy, J. Naor, and B. Schieber. Pushing Dependent Data in Clients-Providers-Servers Systems. *Wireless Networks*, 9(5):421–430, 2003.

[3] Y. Bartal and S. Muthukrishnan. Minimizing Maximum Response Time in Scheduling Broadcasts. In *Proceedings of SODA*, pages 558–559, 2000.

[4] W. Feller. *An Introduction to Probability Theory and Its Applicatitons*. John Wiley & Sons, Inc., 1966.

[5] K. Foltz and J. Bruck. Time Division is Better than Frequency Division for Periodic Internet Broadcast of Dynamic Data. In *Proceedings of ISIT*, page 158, 2001.

[6] K. Foltz and J. Bruck. Splitting Schedules for Internet Broadcast Communication. In *Proceedings of IEEE Transactions on Information Theory*, page 48(2):345–358, 2002.

[7] C. Kenyon and N. Schabanel. The Data Broadcast Problem with Non-Uniform Transmission Times. *Algorithmica*, 35:146–175, 2003.

[8] M. Langberg, A. Sprintson, and J. Bruck. Optimal Universal Schedules for Discrete Broadcast. In *Proceedings of ISIT, page 111*, 2004.

[9] P. M. Vaidya. An Algorithm for Linear Programming Which Requires $o(((m+n)n^2+(m+n)^{1.5}n)L)$ Arithmetic Operations. In *Proceedings of STOC*, pages 29–38, 1987.