Periodic Broadcast Scheduling for Data Distribution

Thesis by

Kevin Foltz

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2002
(Submitted May 17, 2002)

Acknowledgements

Thanks to everyone in Paradise Lab for a great grad student experience. In particular, thanks to Shuki for helping me learn what research is and how to go about doing it. He may claim to know nothing and talk nonsense, but sometimes that's just the kind of advice that gets the ideas flowing. Thanks to all my surfing/skiing/hiking buddies for helping me to get out and enjoy what the area outside of campus has to offer. And thanks to my parents for being there through many, many years of school.

Abstract

As wireless computer networks grow in size and complexity, we are faced with the problem of providing scalable, high-bandwidth service to their users. Wired networks typically use "data pull," where users send requests to a server and the server responds with the desired information. In the wireless domain, "data push" promises to provide better performance for many applications [1]. The broadcast domain that is typical of wireless communication is very effective in distributing information to large audiences.

The idea of broadcast disks has been around since the Teletext system [3]. There is now an interest in applying these ideas to wireless computer networks. There are some interesting research questions about scheduling for data distribution. Computing optimal schedules has been shown to be difficult [18]. The optimal schedules themselves, however, seem to be less complex, and often periodic [4]. Xu [24] looks at the scheduling of streaming data, which involves splitting the data into smaller pieces. The idea of error correction is also important for wireless transmission due to the noisy nature of the channel [6].

We look at scheduling data for broadcast. We compare time-division scheduling and frequency-division scheduling for data items of equal length. We show that time-division is better for sending dynamic data. We then find optimal time-division schedules for two items. We show how the freedom to split items into smaller pieces can give improvements in performance. With a single split, where each of two items is split in half, we find the optimal schedules for items of equal length.

We continue with the idea of splitting items, and show what happens when the number of splits is very large. Then, we examine what happens when we add streaming data to our broadcast. We compare time-division and frequency-division as before, and now also look at a mix of the two. We prove bounds on where the mix is the best broadcast method.

Contents

List of Figures

Chapter 1 Introduction

1.1 Wireless Networks

Computer networks have changed the way we store, process, and transfer information. The Internet has made it possible to store information in one part of the world and make it available anywhere in the world with an Internet connection. The Internet infrastructure is established and continues to grow. However, the access points leave much room for improvement. Many homes and offices have computers which can access this growing collection of information, but people often want information when they are away from these areas.

Wireless computer networks are becoming increasingly popular as a way to provide access to the Internet over broad areas through portable, wireless devices. As these networks grow in area and include more users, we will face new challenges in providing scalable, high-bandwidth access to these users. At first glance, wireless Internet connections seem very similar to wire connections, except that clients can move within a specified area instead of requiring a wire connection. However, as more users move to wireless connections, the resulting networks will face problems that wired networks do not have.

One fundamental difference is that in a wireless network, communication is done by broadcasts, while wired network communication is usually done along wires from one computer to another. In wireless networks everyone in a region must share the communication channel, making communication slower and less efficient. The difficulties of wireless communication are not just due to broadcasts. Ethernet also uses broadcasts, but with Ethernet, we can quickly determine if a collision has occurred, since everyone can hear everyone else. In a wireless network, a node may receive data from two different nodes that do not detect each other. This “hidden terminal problem” makes efficient, reliable communication difficult.

Although wireless communication has its drawbacks when it comes to point-to-point communication, it is very effective in distributing information to large audiences. Wired networks typically use “data pull,” where users send requests to a server and the server responds with the desired information. However, in many cases “data push,” where the server sends information with no user requests, provides better performance. For example, if certain Web pages are very popular and their requests by wireless clients would collide too often to make pull-based communication efficient, it is better to simply send the popular pages and have the clients listen for the pages they want. In this way there are no collisions, and each broadcast can serve many clients simultaneously.

This is the idea behind the broadcast disk. Data items are sent repeatedly in a periodic, or nearly periodic, manner with their frequency of broadcast determined by their demand and size. The broadcast disk idea has been used in the Teletext system, as discussed by Ammar and Wong [3]. It has a promising future with wireless communication, using the air to send data instead of wires. If portable computing devices become as popular as cellular phones, we will need techniques to make efficient use of the available bandwidth. Broadcasting information within cells is one way to help achieve this goal. Also, digital cable systems provide another application for intelligent broadcast techniques. As more channels send data in addition to video, scheduling this data will become increasingly important.

1.2 Scheduling

Scheduling sounds like an easy task at first. To send n data items, we can just send them one after the other, and then repeat. This works if all the data items are about the same size and have about equal demands. However, when this is not the case, we would still like to send information in a way that is efficient.

For example, consider the items shown in Figure 1.1. We assume that there are three data items to broadcast, with lengths $l_1 = 5$, $l_2 = 30$, and $l_3 = 2$. We assume that these items can change over time, so to assure consistent data, clients must

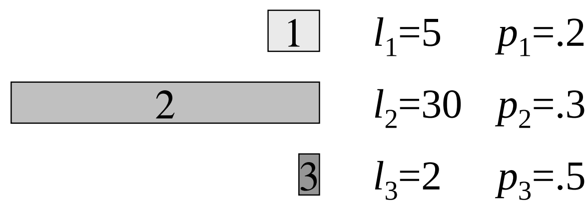

Figure 1.1: Three items, with their corresponding lengths and demand probabilities.

receive a data item from start to finish. If a client starts listening while the item it desires is being sent, it must wait until the beginning of the next broadcast of that item. We want to send these items to clients whose demands for the items are $p_1 = .2$, $p_2 = .3$, and $p_3 = .5$. If we send these items one after the other repeatedly, the clients will wait for item 1 for a time between 0 and $5 + 30 + 2 = 37$, with the expected waiting time being $\frac{37}{2}$. Since items 2 and 3 have the same spacing as item 1, a client will also have an expected waiting time of $\frac{37}{2}$ for these items. The overall expected waiting time is simply the average of these times, weighted by the demands for the items. This is $\frac{37}{2} \cdot .2 + \frac{37}{2} \cdot .3 + \frac{37}{2} \cdot .5 = 18.5$. For this particular schedule, all the times are the same, so the expected waiting time will be 18.5 for any values of the p_i's.

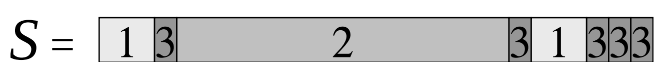

Figure 1.2: A schedule for three items.

Now, consider the periodic schedule, S, one period of which is shown in Figure 1.2. Figures 1.3, 1.4, and 1.5 show the waiting times, $T^{S,i}_{wait}$, for items $i = 1$, 2, and 3, respectively, as a function of $t^{S,i}_{init}$, the time at which a client starts listening for that item within the schedule. The expected waiting time for each item is shown in the figures as a dashed horizontal line. The overall expected waiting time is the average of these times, weighted by the p_i's.

For this example, we get expected waiting times of 16.42, 25, and 11.3 for items 1, 2, and 3, respectively, and an overall expected waiting time of $.2 \cdot 16.42 + .3 \cdot 25 + .5 \cdot 11.3 = 16.434$. We see that the expected waiting times for the items are not always decreased. Item 2 has a longer expected waiting time in the new schedule than in the simple schedule. However, the overall expected waiting time is lower. This is the idea of scheduling. Since we have a fixed bandwidth to share, we must find a way to

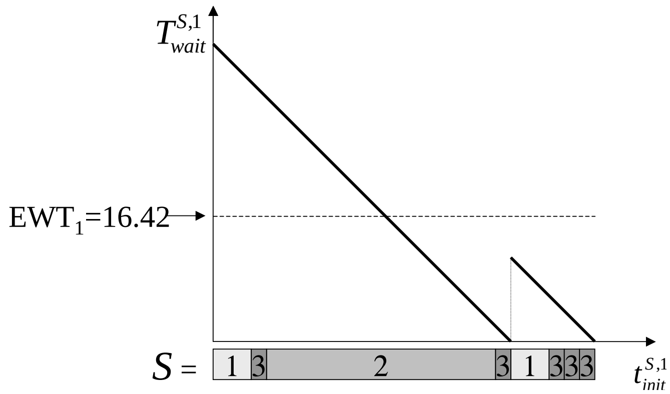

Figure 1.3: Waiting time for item 1 as a function of initial listening time within the schedule.

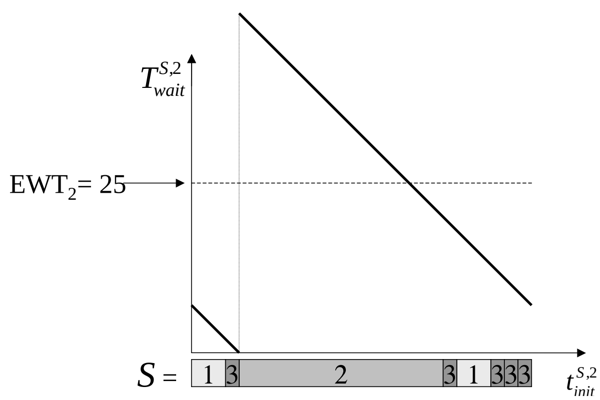

Figure 1.4: Waiting time for item 2 as a function of initial listening time within the schedule.

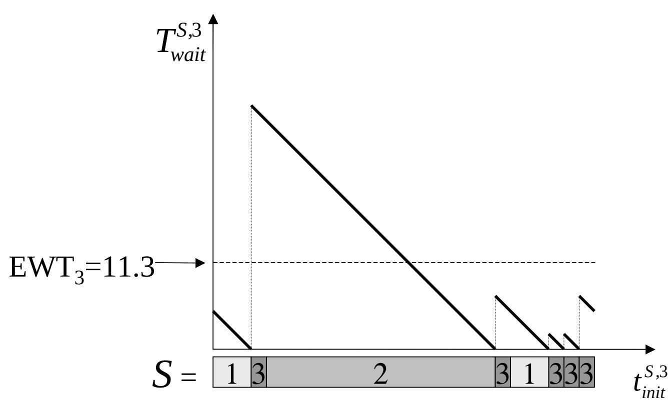

Figure 1.5: Waiting time for item 3 as a function of initial listening time within the schedule.

share it that makes some waits longer and others shorter, but makes the average wait as short as possible.

We see that this schedule, although non-optimal, provides improved performance over the simple schedule of repeatedly sending the items sequentially. In this case, the improvement is modest. For some cases, we can prove that the simple schedule is actually optimal. For others, intelligent scheduling provides large improvements.

1.3 Research Questions and Related Work

There are many research questions related to broadcast schedules. Here we list a few of them.

- **Complexity of Computing Optimal Schedules.** Any set of items and demand probabilities has an optimal schedule. How computationally difficult is it to find these optimal schedules as a function of the data items we wish to broadcast and their demands by the clients?

- **Complexity of Optimal Schedules.** We would like to know how easy the optimal schedules are to describe, in terms of length of a schedule's period, frequency of changing which item is sent, or other metrics. Before we answer this, we should also ask whether the optimal schedules are always periodic. In general, how complicated are the optimal schedules?

- **Time-Division vs. Frequency-Division.** We can schedule using frequency-division, time-division, a mixture of the two, or some other method. How does the performance of frequency-division scheduling compare with that of time-division scheduling? How does mixing these two compare with using either alone? Do the optimal schedules always fall into one of these classes, or is there some other method that can perform better?

- **Data Types.** We can send static data, dynamic data, data that is updated regularly, or data that requires some constant average bandwidth. How do properties of the data affect how we should schedule it? Are the same schedules optimal for both dynamic and static data? When mixing data types, do new types of scheduling become optimal?

- **Splitting.** Sometimes a data item consists of many independent smaller pieces combined together. Instead of sending the data item, we might want to split it into its components and send them independently. What advantages does the ability to split data into smaller pieces provide?

- **Errors.** How can we combine error detection and correction with scheduling? Can k out of n codes provide increased performance or increase error-handling abilities? When is it better to add parity bits to the items, and when is it better to have clients wait for the next broadcast of that item if they receive it in error?

- **Robustness.** Knowing the optimal schedule is good, but sometimes system parameters change, or parameters are not known precisely. Are there situations when certain schedules are optimal over a broad range of system parameters? Which schedules are robust with respect to small changes or errors in system parameters?

There has been work on some of these areas, and some questions have been answered. Vaidya and Hameed [13, 14, 23] looked at finding optimal schedules. They found a theoretical bound on how well a schedule can perform, and then proposed an algorithm whose performance approximates this bound. Su and Tassiulas [21] also examine broadcast scheduling. In [22], they also discuss memory management for client caching. Liberatore [20] also discusses caching, but in the context of on-line scheduling algorithms. Jiang and Vaidya [16] discuss performance metrics combining the mean and variance of the response time, and show how to adjust schedules based on these metrics. Khanna and Zhou [17] look at combining tuning time and waiting time for clients and minimizing them together.

Aksoy et al. [1] and Franklin and Zdonik [12] discuss the general issues related to broadcast disks and push-based technology. Hassin and Megiddo [15] look at the problem of scheduling inventory replenishment. This is very different than the broadcast problem, but the scheduling problem behind each is similar. Bar-Noy, Bhatia, Naor, and Schieber [4] look at scheduling in general and show that there is an optimal cyclic schedule for a broadcast disk and that finding it is NP-hard. Kenyon and Schabanel [18] examine the broadcast of multiple items with different lengths and transmission costs. They show that finding optimal schedules is NP-hard, and the optimal schedules seem very different structurally than schedules for equal lengths. Bar-Noy, Nisgav, and Patt-Shamir [5] discuss perfectly periodic scheduling, where each item has constant spacing within a periodic schedule.

Aksoy and Franklin [2] discuss scheduling the broadcast of information based on client requests. They consider such metrics as average and worst case performance, scheduling overhead, and robustness in the presence of environmental changes. Leong and Si [19] also discuss how to choose which items to broadcast, using ideas of cache management. This does not address scheduling itself, but looks at choosing the broadcast data to put in the schedule. Bestavros [6] describes a way to add fault tolerance to broadcast disks by sending parity information in addition to data. Xu [24] discusses splitting data into pieces to reduce waiting time, in the context of streaming data. Franklin, Zdonik, Acharya, and Alonso [11, 25] also discuss aspects of broadcast disks.

1.4 Contributions

In this thesis, we examine different ways to schedule the broadcast of information to multiple clients. We address the following areas:

- **Time-Division vs. Frequency-Division.** We [8] prove the optimality, with respect to minimal expected waiting time for the clients, of a certain type of frequency-division schedule, for both static and dynamic data. We provide a constructive proof that time-division schedules are better than the optimal frequency-division schedules for broadcasting equal-length dynamic data.

- **Data Types.** We show that optimal schedules for frequency-division are the same whether the data is static or dynamic. We examine dynamic data and find some optimal time-division schedules. We look at combining streaming video data with dynamic data, and find where a combination of time-division and frequency-division is better than either alone.

- **Splitting.** We show how splitting data items in half can improve performance, and we [7] provide optimal schedules for two dynamic items of equal length. We examine multiple splits and prove some facts about when and how much to split data items, and how to schedule their pieces.

- **Robustness.** We [9] show an example where a simple schedule for two items is optimal and robust to small changes in demands for the items.

The following chapters examine the topics above and provide answers to some specific questions in this area. In Chapter 2, we first give definitions for the basic ideas used in the rest of the paper. We analyze frequency-division scheduling, finding the optimal schedules for both static and dynamic data. We then show that for dynamic data items of equal length, we can always find a better time-division schedule.

In Chapters 3, 4, and 5 we look at finding optimal time-division schedules. We first find the optimal schedules for two dynamic data items. Then, we consider splitting two equal-length items into halves and scheduling these smaller pieces. We find the optimal schedules and present a proof of their optimality. We then generalize to arbitrary splits. We make a proposition regarding the optimal schedules for this model and prove some lemmas that support this proposition. One of these lemmas shows the robustness of a simple schedule with respect to small changes in the demands for the items.

In Chapter 6, we consider scheduling for streaming data and dynamic data simultaneously. We find that a mixture of time-division and frequency-division is sometimes best. In Chapter 7, we summarize the results and propose future directions for research.

Chapter 2 Comparing Frequency-Division and Time-Division Scheduling

We examine frequency-division scheduling and time-division scheduling. In Section 2.1, we introduce the ideas of frequency-division and time-division, and define terms relating to scheduling and expected waiting times. In Section 2.2, we define the square root rule frequency-division schedule and prove its optimality for both dynamic and static data. Then, in Section 2.3, we compare time-division and frequency-division for equal-length data. We show that for dynamic data time-division is always better. For static data we consider the scheduling of 2 or 3 items, and show that for these cases time-division is better.

2.1 Introduction

We consider a set of n items, numbered 1 through n, where each item i has a length, l_i, and a demand probability, p_i. The length, l_i, is simply a measure of the size in bytes of data item i. The demand probability, p_i, is the probability that the next item that any client wants is item i. We assume that item requests are Poisson processes, so the p_i values are constant over time.

Since we will be talking about scheduling, let us define some of the terms we will be using.

Definition 2.1 *A schedule for n items and frequency interval $[f_1, f_2]$ is a function $S : \mathbb{R} \times [f_1, f_2] \to \{1, 2, \ldots, n\}$, where $S(t, f)$ is the number of the item sent at time t over frequency f.*

Essentially, a schedule is an assignment of our two resources, time and bandwidth, to the n items we wish to broadcast. We will examine two special types of schedules:

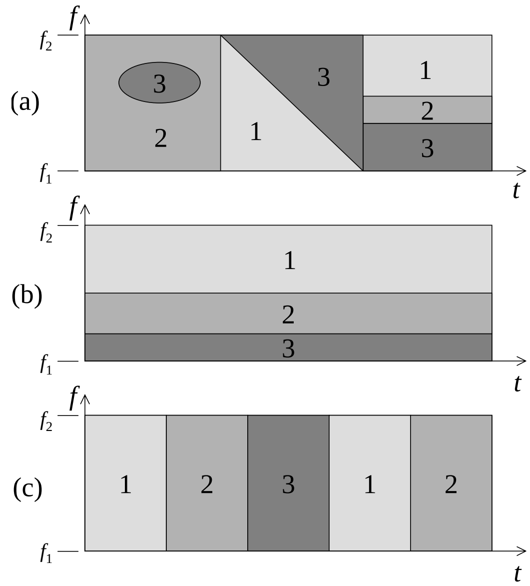

Figure 2.1: Examples of different types of schedules: (a) general schedule (b) frequency-division schedule, and (c) time-division schedule.

Definition 2.2 *A frequency-division schedule for n items and frequency interval $[f_1, f_2]$ is a schedule S for n items and frequency interval $[f_1, f_2]$, where for all $t \in \mathbb{R}$ and $f \in [f_1, f_2]$, $S(t, f) = f_F(f)$, for some function $f_F : [f_1, f_2] \to \{1, 2, \ldots, n\}$.*

Definition 2.3 *A time-division schedule for n items and frequency interval $[f_1, f_2]$ is a schedule S for n items and frequency interval $[f_1, f_2]$, where for all $t \in \mathbb{R}$ and $f \in [f_1, f_2]$, $S(t, f) = f_T(t)$, for some function $f_T : \mathbb{R} \to \{1, 2, \ldots, n\}$.*

Figure 2.1 represents some schedules graphically. We will generally consider periodic schedules S with $[f_1, f_2] = [0, B]$, where B is the bandwidth we have available for broadcast. We will refer to these as schedules for n items and bandwidth B. Using frequency-division, we essentially divide the bandwidth into n channels and repeatedly send one item per channel. With time-division, we simply send items one after another in some specified order using the full bandwidth available.

A schedule tells us when to send each item, but it does not tell us which part of the item to send. We assume that we send bits of each item sequentially within their bandwidth allocation. When we must get an item from start to finish, the starting and ending points for the items are important to know. We define starting points and ending points as follows:

Definition 2.4 *The delta function, $\delta(a, b)$, is 1 if $a = b$, 0 otherwise.*

Definition 2.5 *The k^{th} starting point for item i using schedule S, for $k \in \mathbb{Z}$, is the largest time t for which $\int_0^t \int_{f_1}^{f_2} \delta(S(t, f), i)\, df\, dt = (k - 1) \cdot l_i$*

Definition 2.6 *The k^{th} ending point for item i using schedule S, for $k \in \mathbb{Z}$, is the smallest time t for which $\int_0^t \int_{f_1}^{f_2} \delta(S(t, f), i)\, df\, dt = k \cdot l_i$*

We assume that at time zero we are ready to send the first bit of each item. The k^{th} starting point of item i is the time when we start sending the first bit of item i for the k^{th} time. The k^{th} ending point of item i is the time when we have just completed sending item i for the k^{th} time.

We consider two types of data. The first is dynamic data. This data can change from one broadcast to the next. If item 1 is dynamic, then each broadcast of item 1 would refer to the same basic information, but the actual content could change with time. For example, stock quotes and sports scores would be dynamic data. You could make two requests for the same information and receive different data each time, since stock prices fluctuate with trading, and sports scores change as the games progress.

The second type of data we consider is static data. This is the more traditional type of data. If item 2 is static, then each broadcast of item 2 will contain the same sequence of bits. An example is photographs of Mars from a recent exploration. Many people will want to access these images, but the data will never change once it is posted. There are other types of data, such as information that is updated regularly, or information that is updated irregularly and infrequently.

We define the idea of waiting time for both dynamic and static data.

Definition 2.7 *The waiting time for dynamic item i of length l_i with schedule S (for n items and frequency interval $[f_1, f_2]$) and initial listening time t_1 is a function $WT_i(S, t_1) = t_2 - t_1 - \frac{l_i}{f_2 - f_1}$, where t_2 is the smallest ending point for item i such that*

$$\int_{t_1}^{t_2} \int_{f_1}^{f_2} \delta(S(t, f), i) df\, dt \geq l_i.$$

Definition 2.8 *The waiting time for static item i of length l_i with schedule S (for n items and frequency interval $[f_1, f_2]$) and initial listening time t_1 is a function $WT_i(S, t_1) = t_2 - t_1 - \frac{l_i}{f_2 - f_1}$, where t_2 is the smallest time such that*

$$\int_{t_1}^{t_2} \int_{f_1}^{f_2} \delta(S(t, f), i) df\, dt \geq l_i.$$

For dynamic data, we must start at a starting point and end at an ending point, since we receive data from start to finish. For static data, we have more freedom. We can start receiving data when we start listening to the channel. If we start listening during the transmission of the data we want, we can collect the last part of the current transmission and then finish when we hear the beginning of the next transmission. So, our restriction of starting at a starting point and ending at an ending point is eliminated.

In each case, we subtract the term $\frac{l_i}{f_2 - f_1}$ because we will always listen at least this long for item i. Even in the best possible case, where we immediately start receiving item i from its beginning and continue receiving it over the full available bandwidth, we still listen a time $\frac{l_i}{f_2 - f_1}$. Since we are interested in studying waiting times due to scheduling, we subtract this constant "transmission time" when computing the waiting time.

Definition 2.9 *The expected waiting time for item i is a function, EWT_i, of a schedule, S, where*

$$EWT_i(S) = \lim_{T \to \infty} \frac{1}{T} \int_0^T WT_i(S, t) dt.$$

For a periodic schedule with period T, where the starting points and ending points are also periodic with period T, we can eliminate the limit and compute $EWT_i(S)$ as $\frac{1}{T} \int_0^T WT_i\,(S, t)\, dt$.

Definition 2.10 *Expected waiting time (EWT) is a function of a schedule, S, and a*

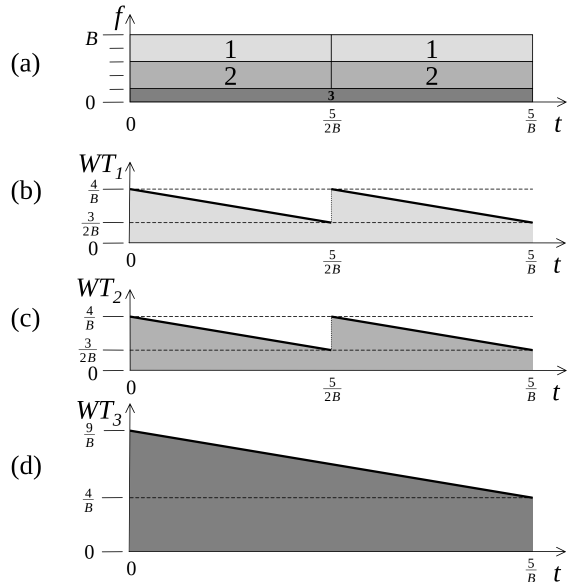

Figure 2.2: Sample WT calculation for a frequency-division schedule for dynamic data: (a) the schedule S, (b) $WT_1(S,t)$, (c) $WT_2(S,t)$, and (d) $WT_3(S,t)$.

vector of demand probabilities $\vec{p} = (p_1, p_2, \ldots, p_n)$*, where*

$$EWT(S, \vec{p}) = \sum_{i=1}^{n} p_i EWT_i(S).$$

When we have $n = 2$ items, we will often say $EWT(S, p_1)$ instead of $EWT(S, \vec{p})$, where $\vec{p}$ is assumed to be $(p_1, 1 - p_1)$.

Example 2.1

Figures 2.2 and 2.3 show sample WT computations for frequency-division and time-division schedules for dynamic data. Since each of the schedules is periodic with period $\frac{5}{B}$, we can compute $EWT_i(S)$ by simply computing the average value of $WT_i(S,t)$ for $t \in \left[0, \frac{5}{B}\right]$. For the frequency-division schedule, we get $EWT_1 = \frac{11}{4B}$, $EWT_2 = \frac{11}{4B}$, and $EWT_3 = \frac{13}{2B}$. For the time-division schedule, we get $EWT_1 = \frac{13}{10B}$, $EWT_2 = \frac{13}{10B}$, and $EWT_3 = \frac{5}{2B}$. For $p_1 = \frac{9}{20}$, $p_2 = \frac{7}{20}$, and $p_3 = \frac{1}{5}$, we get expected waiting times of $\frac{7}{2B}$ for the frequency-division schedule and $\frac{77}{50B}$ for the time-division schedule.

We see in the example above that the time-division schedule has a lower expected waiting time than the frequency-division schedule. In this chapter, we show that for

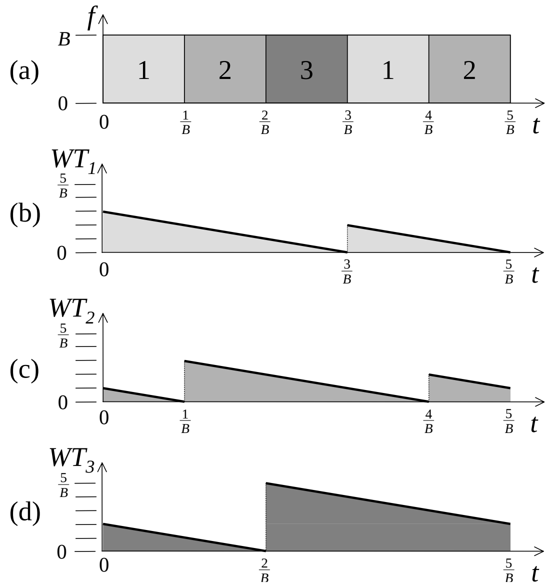

Figure 2.3: Sample WT calculation for a periodic time-division schedule for dynamic data: (a) one period of the schedule S, (b) $WT_1(S,t)$, (c) $WT_2(S,t)$, and (d) $WT_3(S,t)$.

equal-length, dynamic data there is always a time-division schedule that has a lower expected waiting time than any given frequency-division schedule. In Section 2.2, we discuss optimal frequency-division scheduling. In Section 2.3.1, we show how to find better time-division schedules for equal-length, dynamic data. Then in Section 2.3.2, we discuss static data and some cases where time-division is provably better.

2.2 Optimal Frequency-Division Scheduling

2.2.1 The SRR-FD Schedule

We want to use frequency-division scheduling to send n items over a broadcast channel of bandwidth B. This is shown for $n = 3$ in Figure 2.4. When waiting for item i, a client listens to channel i of bandwidth $\alpha_i \cdot B$. We define a specific frequency-division schedule and prove its optimality for both dynamic and static data:

Definition 2.11 *The square root rule frequency-division schedule (SRR-FD schedule) for n items of lengths l_i, $1 \leq i \leq n$, frequency interval $[f_1, f_2]$, and demand probability vector $\vec{p} = (p_1, p_2, \ldots, p_n)$ is a frequency-division schedule S for n items*

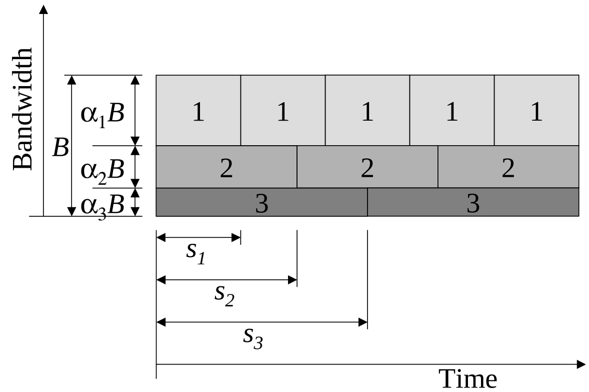

Figure 2.4: Frequency-division scheduling.

and frequency interval $[f_1, f_2]$*, with* $S(t, f) = k$*, if*

$$f_1 + (f_2 - f_1) \sum_{j=k+1}^{n} \alpha_j \leq k < f_1 + (f_2 - f_1) \sum_{j=k}^{n} \alpha_j,$$

where

$$\alpha_i = \frac{\sqrt{p_i l_i}}{\sum_{j=1}^{n} \sqrt{p_j l_j}} \ \forall i, 1 \leq i \leq n.$$

2.2.2 Dynamic Data

We first look at scheduling of dynamic data.

Theorem 2.1 *Among all frequency-division schedules for* n *dynamic items of lengths* l_i*,* $1 \leq i \leq n$*, and frequency interval* $[f_1, f_2]$*, for demand probability vector* $\vec{p} = (p_1, p_2, \ldots, p_n)$*, the corresponding SRR-FD schedule has minimal expected waiting time.*

Proof of Theorem 2.1

First, we note that the waiting times depend only on the total bandwidth allocated to each item, and not on how it is distributed in S. So, we are free to assume that any frequency-division schedule S consists of exactly n different "subchannels," where item i is sent on subchannel i of bandwidth $\alpha_i B$, and $\sum_{i=1}^{n} \alpha_i = 1$. The waiting time, using the notation of Definition 2.7, is $WT_i = t_2 - t_1 - \frac{l_i}{f_2 - f_1}$. We see that $t_2 - t_1$ is the total time spent listening to the channel. This is just $t_{partial} + t_{item}$, where t_{item} is the wait to receive the item from start to finish, starting at the next starting point, and $t_{partial}$ is the wait until the next starting point, which ranges uniformly from 0 to t_{item}. So $t_2 - t_1 = t_{partial} + \frac{l_i}{\alpha_i B}$, and $WT_i(S,t) = t_{partial} + \frac{l_i}{\alpha_i B} - \frac{l_i}{B}$. We now compute $EWT_i(S)$.

$$
\begin{aligned}
EWT_i(S) &= \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(t_{partial} + \frac{l_i}{\alpha_i B} - \frac{l_i}{B} \right) dt \\
&= \lim_{T\to\infty} \frac{1}{T} \int_0^T t_{partial} dt + \frac{l_i}{\alpha_i B} - \frac{l_i}{B} \\
&= \frac{1}{2} \frac{l_i}{\alpha_i B} + \frac{l_i}{\alpha_i B} - \frac{l_i}{B} \\
&= \frac{3}{2} \frac{l_i}{\alpha_i B} - \frac{l_i}{B}
\end{aligned}
$$

This gives us an overall expected waiting time of

$$
\begin{aligned}
EWT(S, \vec{p}) &= \sum_{i=1}^{n} \left(\frac{3}{2} \frac{l_i}{\alpha_i B} - \frac{l_i}{B} \right) p_i \\
&= \frac{3}{2B} \sum_{i=1}^{n} \frac{p_i l_i}{\alpha_i} - \frac{1}{B} \sum_{i=1}^{n} p_i l_i
\end{aligned}
$$

This is minimized when $\sum_{i=1}^{n} \frac{p_i l_i}{\alpha_i}$ is minimized, or when

$$\alpha_i = \frac{\sqrt{p_i l_i}}{\sum_{j=1}^{n} \sqrt{p_j l_j}}.$$

These are the values of the α_i's used by the SRR-FD schedule. It follows that the SRR-FD schedule has minimal expected waiting time among all frequency-division schedules.

Using these values of α_i, we get the minimal expected waiting time of

$$EWT_{min} = \frac{1}{B} \sum_{i=1}^{n} \sqrt{p_i l_i} \left(\frac{3}{2} \sum_{j=1}^{n} \sqrt{p_j l_j} - \sqrt{p_i l_i} \right)$$

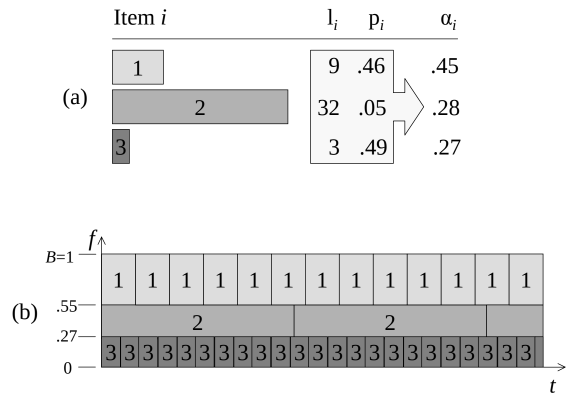

Figure 2.5: Optimal frequency-division scheduling: (a) computing α_i's for the items, (b) the resulting schedule.

Example 2.2

As an example of optimal frequency-division scheduling, we consider scheduling three items with lengths $l_1 = 9$, $l_2 = 32$, and $l_3 = 3$, and demand probabilities $p_1 = .46$, $p_2 = .05$, and $p_3 = .49$. We compute $p_1 l_1 = 4.14$, $p_2 l_2 = 1.6$, and $p_3 l_3 = 1.47$. Taking square roots and normalizing, we get $\alpha_1 = .45$, $\alpha_2 = .28$, and $\alpha_3 = .27$, giving an expected waiting time of 23.33. Figure 2.5 shows the resulting schedule for $B = 1$.

2.2.3 Static Data

We now look at optimal scheduling for static data.

Theorem 2.2 *Among all frequency-division schedules for n static items of lengths l_i, $1 \leq i \leq n$, and frequency interval $[f_1, f_2]$, for demand probability vector $\vec{p} = (p_1, p_2, \ldots, p_n)$, the corresponding SRR-FD schedule has minimal expected waiting time.*

Proof of Theorem 2.2

Again, we note that the waiting times depend only on the total bandwidth allocated to each item, and not on how it is distributed in S. So, we are free to assume that any frequency-division schedule S consists of exactly n different "subchannels," where item i is sent on subchannel i of bandwidth $\alpha_i B$, and $\sum_{i=1}^{n} \alpha_i = 1$. The waiting time, using the notation of Definition 2.8, is $WT_i(S, t_1) = t_2 - t_1 - \frac{l_i}{f_2 - f_1}$. We see that $t_2 - t_1$ is the total time spent listening to the channel. This is just t_{item}, the wait to receive the item from start to finish. So $t_2 - t_1 = \frac{l_i}{\alpha_i B}$, and $WT_i(S, t_1) = \frac{l_i}{\alpha_i B} - \frac{l_i}{B}$. We now compute $EWT_i(S) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(\frac{l_i}{\alpha_i B} - \frac{l_i}{B} \right) dt = \frac{l_i}{\alpha_i B} - \frac{l_i}{B}$.

This gives us an overall expected waiting time of

$$\begin{aligned} EWT &= \sum_{i=1}^{n} \left(\frac{l_i}{\alpha_i B} - \frac{l_i}{B} \right) p_i \\ &= \frac{1}{B} \sum_{i=1}^{n} \frac{p_i l_i}{\alpha_i} - \frac{1}{B} \sum_{i=1}^{n} p_i l_i \end{aligned}$$

This is minimized when $\sum_{i=1}^{n} \frac{p_i l_i}{\alpha_i}$ is minimized, or when

$$\alpha_i = \frac{\sqrt{p_i l_i}}{\sum_{j=1}^{n} \sqrt{p_j l_j}}.$$

As it was for scheduling dynamic data, these are the values of the α_i's used by the SRR-FD schedule. It follows that the SRR-FD schedule has minimal expected waiting time among all frequency-division schedules.

Using these values of α_i, we get the minimal expected waiting time of

$$EWT_{min} = \frac{1}{B} \sum_{i=1}^{n} \sqrt{p_i l_i} \left(\sum_{j=1}^{n} \sqrt{p_j l_j} - \sqrt{p_i l_i} \right)$$

2.3 Better Time-Division Scheduling

We now consider time-division schedules, as in Figure 2.6. Finding the optimal time-division schedules as we did for frequency-division schedules is difficult. Instead, we consider the case when all items have the same length, and for any frequency-division schedule we look for a time-division schedule that has lower expected waiting time.

2.3.1 Dynamic Data

We first consider dynamic data.

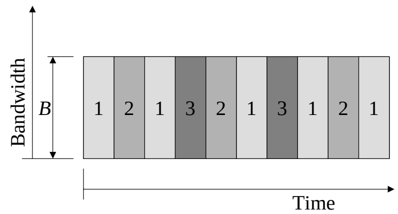

Figure 2.6: Time-division scheduling

Theorem 2.3 *For any frequency-division schedule, S_{FD}, for n dynamic items of length l and frequency interval $[f_1, f_2]$, there is a time-division schedule, S_{TD}, for n dynamic items of length l and frequency interval $[f_1, f_2]$ such that $EWT\left(S_{TD}, \vec{p}\right) \le EWT\left(S_{FD}, \vec{p}\right)$ for any vector $\vec{p} = (p_1, p_2, \ldots, p_n)$ of demand probabilities.*

Proof of Theorem 2.3

The idea of the proof is to consider an arbitrary frequency-division schedule, S_{FD}, and construct a better time-division schedule. We will assume WLOG that the length of the items is $l = 1$, the bandwidth is $B = 1$, and the α_i's are ordered as follows: $\alpha_1 \ge \alpha_2 \ge \alpha_3 \ge \ldots \ge \alpha_n$. We let $s_i = \frac{1}{\alpha_i}$, the spacing between instances of item i in the frequency-division schedule. For the frequency-division schedule, the expected waiting time is $\sum_{i=1}^{n} p_i \left(\frac{3}{2\alpha_i} - 1\right)$. We will construct a time-division schedule that has a lower expected waiting time. We choose our construction based on the value of α_1.

Case 1 $\left(0 < \alpha_1 \le \frac{1}{2}\right)$

When all α_i's are at most $\frac{1}{2}$, we see that for the frequency-division schedule S_{FD} we have

$$\begin{aligned} EWT_i\left(S_{FD}\right) &= \frac{3}{2\alpha_i} - 1 \\ &= \left(\frac{3}{2} - \alpha_i\right) \cdot s_i \\ &\ge s_i \end{aligned}$$

Let $s_i' = 2^{\lceil \log_2 s_i \rceil}$, the next power of 2 greater than or equal to s_i. Since $\sum_{i=1}^{n} \frac{1}{s_i} = 1$ and $s_i' \ge s_i$ for $1 \le i \le n$, it follows that $\sum_{i=1}^{n} \frac{1}{s_i'} \le 1$. Since $\sum_{i=1}^{n} \frac{1}{s_i'}$ is the bandwidth

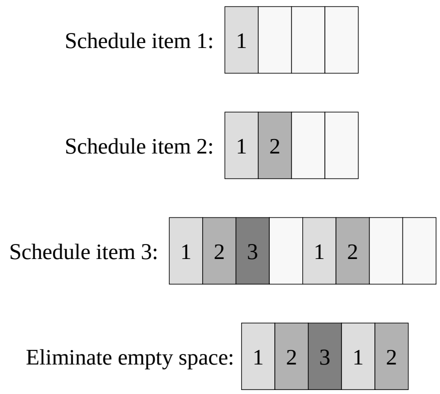

Figure 2.7: Generating a time-division schedule better than the frequency-division schedule with $\alpha_1 = .41$, $\alpha_2 = .36$, and $\alpha_3 = .23$.

requirement for scheduling with spacing s'_i, and 1 is the total bandwidth available, we have enough bandwidth to schedule with these spacings between items. We start with item 1 and schedule it with spacing s'_1. We then repeat for items 2 through n. Since the spacings are increasing powers of 2, we can always schedule the items with the appropriate spacing. Each s'_i is less than twice the corresponding s_i, and the expected waiting time for item i using the time-division schedule is $\frac{1}{2}s'_i < \frac{1}{2}(2s_i) = s_i$. Thus, for this time-division schedule, S_{TD}, we get $EWT_i(S_{TD}) < s_i \leq EWT_i(S_{FD})\ \forall i, 1 \leq i \leq n$. It follows that time-division is better than frequency-division when $\alpha_1 < \frac{1}{2}$.

Example 2.3

To see how to schedule items when $\alpha_1 < \frac{1}{2}$, we consider an example. Let $n = 3$, $\alpha_1 = .41$, $\alpha_2 = .36$, and $\alpha_3 = .23$. Then we assign $s'_1 = 4$, $s'_2 = 4$, $s'_3 = 8$, and schedule the items as shown in Figure 2.7. Using the frequency-division schedule, we get $EWT_1 = 2.66$, $EWT_2 = 3.17$, and $EWT_3 = 5.52$. With the time-division schedule, we get $EWT_1 = 1.4$, $EWT_2 = 1.4$, and $EWT_3 = 2.5$. Since the time-division schedule gives lower expected waiting times for each item, it follows that the time-division schedule has a lower expected waiting time for any values of the demand probabilities p_1, p_2, and p_3.

Case 2 $\left(\frac{1}{2} < \alpha_1 \leq \frac{3}{4}\right)$

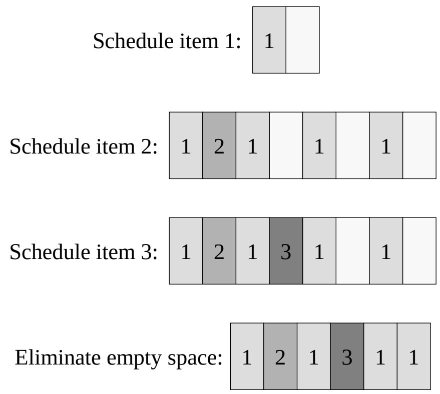

Figure 2.8: Generating a time-division schedule better than the frequency-division schedule with $\alpha_1 = .61$, $\alpha_2 = .23$, and $\alpha_3 = .16$.

For the frequency-division schedule, S_{FD}, with $\frac{1}{2} < \alpha_1 \leq \frac{3}{4}$, we get $EWT_1\left(S_{FD}\right) = \frac{3}{2\alpha_1} - 1 \geq 1$. For our time-division schedule, S_{TD}, we let $s_1' = 2$. This gives $EWT_1\left(S_{TD}\right) = 1$, so $EWT_1\left(S_{TD}\right) \leq EWT_1\left(S_{FD}\right)$. For the other items, the sum of the α_i's is at most $\frac{1}{2}$, and we have $\frac{1}{2}$ the schedule to fit them in. So, we proceed as in Case 1 and round each s_i up to the next power of 2. We then schedule these items as in Case 1. As before, $EWT_i\left(S_{TD}\right) < EWT_i\left(S_{FD}\right)\ \forall i, 1 \leq i \leq n$. So, time-division is better than frequency-division when $\frac{1}{2} < \alpha_1 \leq \frac{3}{4}$.

Example 2.4

To see how to schedule items when $\frac{1}{2} < \alpha_1 < \frac{3}{4}$, we consider an example. Let $n = 3$, $\alpha_1 = .61$, $\alpha_2 = .23$, and $\alpha_3 = .16$. Then we assign $s_1' = 2$, $s_2' = 8$, $s_3' = 8$, and schedule the items as shown in Figure 2.8. Using the frequency-division schedule, we get $EWT_1 = 1.46$, $EWT_2 = 5.52$, and $EWT_3 = 8.38$. With the time-division schedule, we get $EWT_1 = .83$, $EWT_2 = 3$, and $EWT_3 = 3$. Since the time-division schedule gives lower expected waiting times for each item, it follows that the time-division schedule has a lower expected waiting time for any values of the demand probabilities p_1, p_2, and p_3.

Case 3 $\left(\frac{3}{4} < \alpha_1 < 1\right)$

Consider the time-division schedule, S_{TD}, where we broadcast item 1 consecutively r times and then leave an empty spot. This has $EWT_1\left(S_{TD}\right) = \frac{r+3}{2(r+1)}$. For the time-division schedule to have a lower value of EWT_1 than the frequency-division schedule, we need $\frac{r+3}{2(r+1)} < \frac{3}{2\alpha_1} - 1 \Longrightarrow r > \frac{2}{3} \cdot \frac{\alpha_1}{1-\alpha_1} - 1$, so we let $r = \left\lfloor \frac{2}{3} \cdot \frac{\alpha_1}{1-\alpha_1} \right\rfloor$. This uses a fraction $\frac{r}{r+1}$ of the schedule for item 1. This is

$$\begin{aligned}
\frac{r}{r+1} &= \frac{\left\lfloor \frac{2}{3} \cdot \frac{\alpha_1}{1-\alpha_1} \right\rfloor}{\left\lfloor \frac{2}{3} \cdot \frac{\alpha_1}{1-\alpha_1} \right\rfloor + 1} \\
&< \frac{\frac{2}{3} \cdot \frac{\alpha_1}{1-\alpha_1}}{\frac{2}{3} \cdot \frac{\alpha_1}{1-\alpha_1} + 1} \\
&= \frac{2\alpha_1}{2\alpha_1 + 3 - 3\alpha_1} \\
&= \frac{2\alpha_1}{2 + (1 - \alpha_1)} \\
&< \frac{2\alpha_1}{2} \\
&= \alpha_1
\end{aligned}$$

So, we have used less than α_1 of the schedule for item 1, and we have kept $EWT_1(S_{TD}) \leq EWT_1(S_{FD})$. For items $i = 2$ through n, we let $s_i' = (r+1)\, 2^{\left\lceil \log_2 \frac{s_i}{r+1} \right\rceil}$, the next value larger than or equal to s_i of the form $(r+1)\, 2^m, m \in \mathbb{Z}$. If $\alpha_i' = \frac{1}{s_i'}$, then $\sum_{i=2}^{n} \alpha_i' < \sum_{i=2}^{n} \alpha_i$, so our bandwidth requirement is reduced for time-division. We schedule items 2 through n as before, doubling the schedule's length as necessary. For $i \geq 2$, we have $s_i' \leq 2s_i \Longrightarrow EWT_i(S_{TD}) = \frac{1}{2}(s_i') \leq \frac{1}{2}(2s_i) = s_i \leq \frac{3}{2\alpha_i} - 1 = EWT_i(S_{FD})$. So, $EWT_i(S_{TD}) \leq EWT_i(S_{FD})\ \forall i, 1 \leq i \leq n$, and time-division is better than frequency-division when $\frac{3}{4} < \alpha_1 < 1$.

Example 2.5

To see how to schedule items when $\alpha_1 > \frac{3}{4}$, we consider an example. Let $n = 3$, $\alpha_1 = .88$, $\alpha_2 = .07$, and $\alpha_3 = .05$. Then we group item 1 in blocks of $r = 4$, let $s_2' = (4+1) \cdot 4 = 20$ and $s_3' = (4+1) \cdot 4 = 20$, and schedule the items as shown in Figure 2.9. Using the frequency-division schedule, we get $EWT_1 = .70$, $EWT_2 = 20.4$, and $EWT_3 = 29.0$. With the time-division schedule, we get $EWT_1 = .61$, $EWT_2 = 9$,

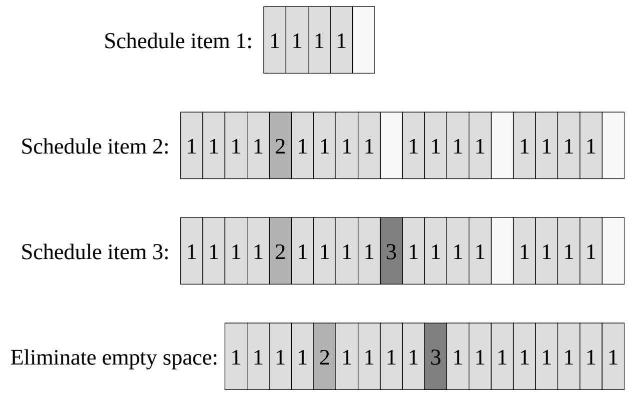

Figure 2.9: Generating a time-division schedule better than the frequency-division schedule with $\alpha_1 = .88$, $\alpha_2 = .07$, and $\alpha_3 = .05$.

and $EWT_3 = 9$. Since the time-division schedule gives lower expected waiting times for each item, it follows that the time-division schedule has a lower expected waiting time for any values of the demand probabilities p_1, p_2, and p_3.

We see that for any value of α_1 and any frequency-division schedule we can find a time-division schedule with a lower expected waiting time. Since all the EWT_i's are lower for time-division, it follows that the expected waiting time (EWT) for the time-division schedule is lower for any values of the demand probabilities, and the theorem is proved.

2.3.2 Static Data

For static data, we do not have a proof like we do for dynamic data. So, instead of showing that time-division is always better than frequency-division, we will show that this is true when we have $n = 2$ or 3 items. These techniques seem to generalize to any particular $n \geq 4$, but the proof for arbitrary n is elusive. First, the theorem for 2 items.

Theorem 2.4 *For any frequency-division schedule, S_1, for 2 static items of length l and frequency interval $[f_1, f_2]$, there is a time-division schedule, S_2, for 2 static items of length l and frequency interval $[f_1, f_2]$ such that $EWT(S_2, \vec{p}) \leq EWT(S_1, \vec{p})$ for any vector $\vec{p} = (p_1, p_2)$ of demand probabilities.*

Proof of Theorem 2.4

We will prove this by showing that the time-division schedule 12^n is better than the optimal frequency-division schedule, with the appropriate choice of n. In the proof of Theorem 2.2, we showed that the optimal frequency-division schedule has expected waiting time $2\sqrt{p_1 p_2} = 2\sqrt{p_1 - p_1^2}$. Consider the schedule $S^n = 12^n$ where $l_1 = l_2 = 1$. We compute the expected waiting time for each item and then the overall expected waiting time.

$$\begin{aligned}
EWT_1(S^n) &= \frac{1}{n+1}\left[1\,(n) + n\left(\frac{n}{2}\right)\right] \\
&= \frac{n^2+2n}{2\,(n+1)} \\
EWT_2(S^n) &= \frac{1}{n+1}\left[1\left(\frac{1}{2}\right) + (n-1)\,(0) + 1\,(1)\right] \\
&= \frac{3}{2\,(n+1)} \\
EWT(S^n, \vec{p}) &= EWT_1(S^n)p_1 + EWT_2(S^n)\,(1-p_1) \\
&= \frac{n^2+2n}{2\,(n+1)}p_1 + \frac{3}{2\,(n+1)}\,(1-p_1) \\
&= \frac{3}{2\,(n+1)} + \frac{n^2+2n-3}{2\,(n+1)}p_1
\end{aligned}$$

Figure 2.10 shows the performance of the optimal frequency-division schedule and some of the time-division schedules of the form 12^n. We compute the values of p_1 for which $EWT(S^n, \vec{p}) = EWT(S^{n+1}, \vec{p})$.

$$\begin{aligned}
EWT(S^n, \vec{p}) &= EWT(S^{n+1}, \vec{p}) \iff \\
\frac{3}{2\,(n+1)} + \frac{n^2+2n-3}{2\,(n+1)}p_1 &= \frac{3}{2\,(n+2)} + \frac{(n+1)^2 + 2\,(n+1) - 3}{2\,(n+2)}p_1 \iff \\
p_1 &= \frac{\frac{3}{2(n+1)} - \frac{3}{2(n+2)}}{\frac{n^2+4n}{2(n+2)} - \frac{n^2+2n-3}{2(n+1)}} \iff \\
p_1 &= \frac{3}{n^2+3n+6}
\end{aligned}$$

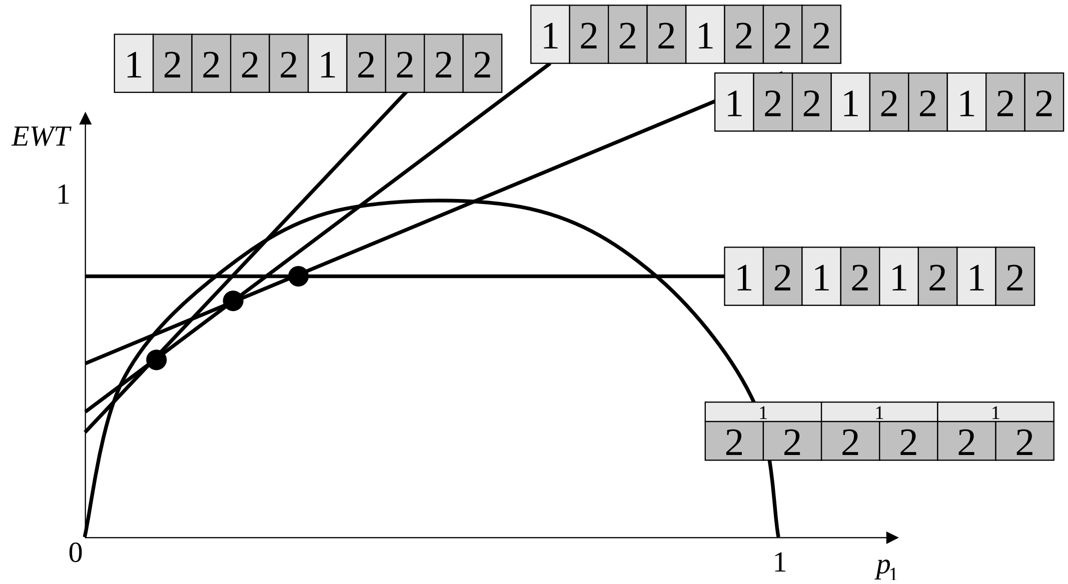

Figure 2.10: Comparing the optimal frequency-division schedule to time-division schedules of the form 12^n.

We now compute the expected waiting time for both S^n and the optimal frequency-division schedule at these values of p_1. We let $\vec{p} = (p_1, p_2)$, where

$$\begin{aligned} p_1 &= \frac{3}{n^2+3n+6} \\ p_2 &= 1 - \frac{3}{n^2+3n+6} \end{aligned}$$

$$\begin{aligned} t_{time}^{n=n+1} = EWT(S^n, \vec{p}) &= \frac{3}{2\,(n+1)} + \frac{n^2+2n-3}{2\,(n+1)} \frac{3}{n^2+3n+6} \\ &= \frac{3}{2} \frac{2n+3}{n^2+3n+6} \\ t_{freq}^{n=n+1} = EWT(S_{FDopt}, \vec{p}) &= 2\sqrt{\frac{3}{n^2+3n+6} - \frac{9}{(n^2+3n+6)^2}} \\ &= \frac{2}{n^2+3n+6}\sqrt{3n^2+9n+9} \end{aligned}$$

We compare the expected waiting times for the time-division and frequency-division schedules and see that the time-division schedules are always better.

$$
\begin{aligned}
t_{freq}^{n=n+1} \geq t_{time}^{n=n+1} &\iff \frac{2}{n^2+3n+6}\sqrt{3n^2+9n+9} \geq \frac{3}{2}\frac{2n+3}{n^2+3n+6} \\
&\iff 4\sqrt{3n^2+9n+9} \geq 3\left(2n+3\right) \\
&\iff 16\left(3n^2+9n+9\right) \geq 9\left(4n^2+12n+9\right) \\
&\iff 12n^2+36n+63 \geq 0 \longleftarrow \text{ true } \forall n \geq 1
\end{aligned}
$$

We see that we can always find a time-division schedule of the form 12^n that is better than the optimal frequency-division schedule. Hence, time-division is better than frequency-division for broadcasting 2 static data items.

We now consider 3 static data items.

Theorem 2.5 *For any frequency-division schedule, S_1, for 3 static items of length l and frequency interval $[f_1, f_2]$, there is a time-division schedule, S_2, for 3 static items of length l and frequency interval $[f_1, f_2]$ such that $EWT\left(S_2, \vec{p}\right) \leq EWT\left(S_1, \vec{p}\right)$ for any vector $\vec{p} = (p_1, p_2, p_3)$ of demand probabilities.*

Proof of Theorem 2.5

We will prove this by showing that one of the time-division schedules 123, 1^n213, or $1^n21^n21^n21^n213$ is better than the optimal frequency-division schedule, with the appropriate choice of $n \geq 1$. In the proof of Theorem 2.2, we showed that the optimal frequency-division schedule has expected waiting time $2\sqrt{p_1p_2} + 2\sqrt{p_1p_3} + 2\sqrt{p_2p_3}$.

Consider the schedule $S^n = 1^n213$, where $l_1 = l_2 = l_3 = 1$. We compute the expected waiting times for each item and then the overall expected waiting time. In this proof, we will assume that $p_1 \geq p_2 \geq p_3$. By symmetry, we need only show this case to show the theorem holds for all values of p_1, p_2, and p_3.

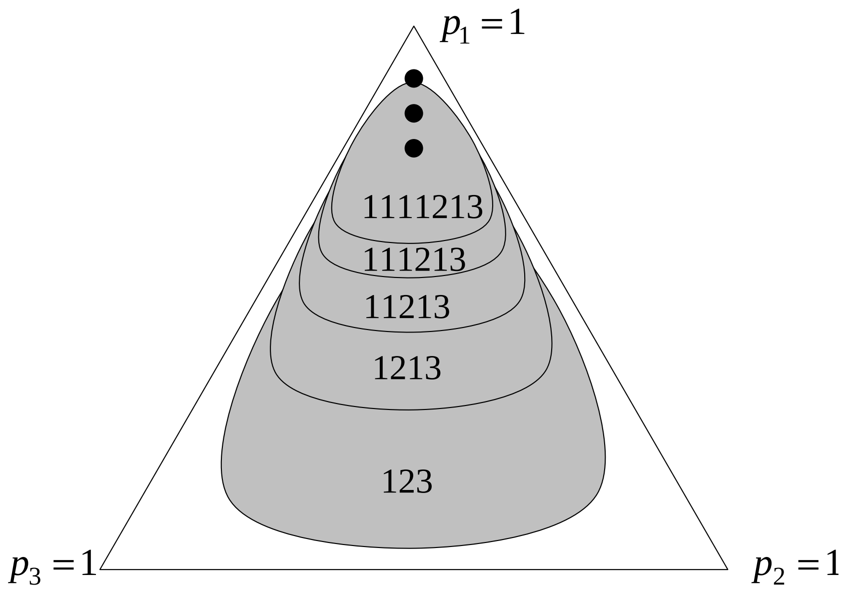

Figure 2.11: Regions where time-division schedules of the form 1^n213 have lower expected waiting time than the optimal frequency-division schedules.

$$
\begin{aligned}
EWT_1(S^n) &= \frac{1}{n+3}\left[(n-1)(0) + 1(1) + 1\left(\frac{1}{2}\right) + 1(1) + 1\left(\frac{1}{2}\right)\right] \\
&= \frac{3}{n+3} \\
EWT_2(S^n) &= \frac{1}{n+3}\left[1(n+2) + (n+2)\left(\frac{n+2}{2}\right)\right] \\
&= \frac{n^2+6n+8}{2(n+3)} \\
EWT_3(S^n) &= \frac{1}{n+3}\left[(n+2)\left(\frac{n+2}{2}\right) + 1(n+2)\right] \\
&= \frac{n^2+6n+8}{2(n+3)} \\
EWT(S^n, \vec{p}) &= EWT_1(S^n)p_1 + EWT_2(S^n)p_2 + EWT_3(S^n)(1-p_1-p_2) \\
&= \frac{n^2+6n+8}{2(n+3)} - \frac{n^2+6n+2}{2(n+3)}p_1
\end{aligned}
$$

Figure 2.11 shows the regions where each time-division schedule is better than the optimal frequency-division schedule. The triangle represents all possible values of the demands p_1, p_2, and p_3, and the shaded regions are where the time-division schedules are better. We compute the values of p_1 for which $EWT(S^n, \vec{p}) = EWT(S^{n+1}, \vec{p})$.

$$\begin{aligned}
EWT(S^n,\vec{p}) &= EWT(S^{n+1},\vec{p}) \Longleftrightarrow \\
\frac{n^2+6n+8}{2(n+3)} - \frac{n^2+6n+2}{2(n+3)}p_1 &= \frac{(n+1)^2+6(n+1)+8}{2(n+4)} \\
&\quad - \frac{(n+1)^2+6(n+1)+2}{2(n+4)}p_1 \Longleftrightarrow \\
p_1 &= 1 - \frac{6}{n^2+7n+19}
\end{aligned}$$

We now compute the expected waiting time for both S^n and the optimal frequency-division schedule, S_{FDopt}, at these boundaries. We consider the case where $.1\,(1-p_1) \leq p_2 \leq .9\,(1-p_1)$. By symmetry, and concavity of $t_{freq} - t_{time}$, we need only consider the boundary $p_2 = .9\,(1-p_1)$. We let $\vec{p} = (p_1, p_2, p_3)$, where

$$\begin{aligned}
p_1 &= 1 - \frac{6}{n^2+7n+19} \\
p_2 &= \frac{9}{10}(1-p_1) \\
p_3 &= \frac{1}{10}(1-p_1)
\end{aligned}$$

$$\begin{aligned}
t_{time}^{n=n+1} = EWT(S^n,\vec{p}) &= \frac{n^2+6n+8}{2(n+3)} - \frac{n^2+6n+2}{2(n+3)} \cdot \left(1 - \frac{6}{n^2+7n+19}\right) \\
&= \frac{3(2n^2+13n-9)}{(n+3)(n^2+7n+19)} \\
t_{freq}^{n=n+1} = EWT(S_{FDopt},\vec{p}) &= 2\sqrt{\left(1-\frac{6}{n^2+7n+19}\right)\left(\frac{9}{10}\frac{6}{n^2+7n+19}\right)} \\
&\quad +2\sqrt{\left(1-\frac{6}{n^2+7n+19}\right)\left(\frac{1}{10}\frac{6}{n^2+7n+19}\right)} \\
&\quad +2\sqrt{\left(\frac{9}{10}\frac{6}{n^2+7n+19}\right)\left(\frac{1}{10}\frac{6}{n^2+7n+19}\right)} \\
&= \frac{12}{n^2+7n+19}\left(\frac{3}{10} + 2\sqrt{\frac{n^2+7n+13}{15}}\right)
\end{aligned}$$

We compare the expected waiting times for the time-division and frequency-division schedules and see that the time-division schedules are always better.

$$
\begin{aligned}
t_{freq}^{n=n+1} \geq t_{time}^{n=n+1} \quad & \Longleftrightarrow \quad 12\left(\frac{3}{10}+2\sqrt{\frac{n^2+7n+13}{15}}\right) \geq \frac{3\left(2n^2+13n-9\right)}{n+3} \\
& \Longleftrightarrow \quad 8\sqrt{3}\sqrt{n^2+7n+13} \geq \frac{1}{\sqrt{5}\,(n+3)}\left(30n^2+122n-189\right) \\
& \Longleftrightarrow \quad 192\left(n^2+7n+13\right) \geq \frac{\left(30n^2+122n-189\right)^2}{5\,(n+3)^2} \\
& \Longleftrightarrow \quad 60n^4+5160n^3+44816n^2 \\
& \qquad\quad +177696n+76599 \geq 0 \longleftarrow \ \text{true } \forall n \geq 1
\end{aligned}
$$

So, we see that we can always find a time-division schedule of the form 1^n213 that is better than the optimal frequency-division schedule. Hence, time-division is better than frequency-division for broadcasting static data when $.1\,(1-p_1) \leq p_2 \leq .9\,(1-p_1)$.

Actually, this only applies for $n \geq 1 \Longrightarrow p_1 \geq 1 - \frac{6}{1^2+7\cdot 1+19} = \frac{7}{9}$. However, the schedule 123 covers the remaining area. We first compute the expected waiting times for the schedules $S^1 = 1213$ and $S^0 = 123$.

$$
\begin{aligned}
EWT(S^1,\vec{p}) &= \frac{1^2+6\cdot 1+8}{2(1+3)} - \frac{1^2+6\cdot 1+2}{2(1+3)}p_1 \\
&= \frac{15}{8} - \frac{9}{8}p_1 \\
EWT(S^0,\vec{p}) &= \frac{4}{3}p_1 + \frac{4}{3}p_2 + \frac{4}{3}p_3 \\
&= \frac{4}{3}
\end{aligned}
$$

We compute the value of p_1 for which $EWT(S^1,\vec{p}) = EWT(S^0,\vec{p})$.

$$\begin{aligned}
\frac{15}{8} - \frac{9}{8}p_1 &= \frac{4}{3} \iff \\
p_1 &= \frac{8}{9}\left(\frac{15}{8} - \frac{4}{3}\right) \iff \\
p_1 &= \frac{13}{27}
\end{aligned}$$

We now compute the expected waiting time for both S^0 and the optimal frequency-division schedule, S_{FDopt}, at these boundaries. We consider the case where $.1\,(1 - p_1) \leq p_2 \leq .9\,(1 - p_1)$. By symmetry, and concavity of $t_{freq} - t_{time}$, we need only consider the boundary $p_2 = .9\,(1 - p_1)$. We let $\vec{p} = (p_1, p_2, p_3)$, where

$$\begin{aligned}
p_1 &= \frac{13}{27} \\
p_2 &= \frac{7}{15} \\
p_3 &= \frac{7}{135}
\end{aligned}$$

$$\begin{aligned}
t_{time} = EWT(S^0, \vec{p}) &= \frac{4}{3} \\
t_{freq} = EWT(S_{FDopt}, \vec{p}) &= 2\sqrt{\frac{13}{27} \cdot \frac{7}{135}} + 2\sqrt{\frac{13}{27} \cdot \frac{7}{135}} + 2\sqrt{\frac{7}{15} \cdot \frac{7}{135}} \\
&= 2\left((1+3)\sqrt{\frac{13}{27} \cdot \frac{7}{135}} + \frac{7}{45}\right) \\
&= \frac{42 + 8\sqrt{455}}{135}
\end{aligned}$$

We compare the expected waiting times for the time-division and frequency-division schedules and see that the time-division schedules are always better.

$$\begin{aligned}
t_{freq} \geq t_{time} &\iff \frac{42 + 8\sqrt{455}}{135} \geq \frac{4}{3} \\
&\iff 1.575\ldots \geq 1.333\ldots \longleftarrow \text{ true}
\end{aligned}$$

So, we have verified that time-division is better than frequency-division for broadcasting static data when $.1\,(1-p_1) \le p_2 \le .9\,(1-p_1)$.

Consider the schedule $S^n = 1^n 2 1^n 2 1^n 2 1^n 2 1 3$ where $l_1 = l_2 = l_3 = 1$. We compute the expected waiting times for each item and then the overall expected waiting time.

$$
\begin{aligned}
EWT_1(S^n) &= \frac{1}{4n+6}\left[\left((n-1)(0) + 1(1) + 1\left(\frac{1}{2}\right)\right)\cdot 4 + 1(1) + 1\left(\frac{1}{2}\right)\right] \\
&= \frac{15}{4(2n+3)} \\
EWT_2(S^n) &= \frac{1}{4n+6}\left[\left(n\left(\frac{n}{2}\right) + 1(n)\right)\cdot 3 + n\left(\frac{n}{2}\right) + 1(n+2) + 2(n+1)\right] \\
&= \frac{n^2+3n+2}{2n+3} \\
EWT_3(S^n) &= \frac{1}{4n+6}\left[(4n+5)\left(\frac{4n+5}{2}\right) + 1(4n+5)\right] \\
&= \frac{16n^2+48n+35}{4(2n+3)} \\
EWT(S^n,\vec{p}) &= EWT_1(S^n)p_1 + EWT_2(S^n)p_2 + EWT_3(S^n)(1-p_1-p_2) \\
&= \frac{16n^2+48n+35}{4(2n+3)} - \frac{4n^2+12n+5}{2n+3}p_1 - \frac{12n^2+36n+27}{4(2n+3)}p_2
\end{aligned}
$$

Figure 2.12 shows the regions where each time-division schedule is better than the optimal frequency-division schedule. We compute the values of p_1, p_2, and p_3 for which $EWT(S^n,\vec{p}) = EWT(S^{n+1},\vec{p})$.

$$
\begin{aligned}
EWT(S^n,\vec{p}) &= EWT(S^{n+1},\vec{p}) \Longleftrightarrow \\
4\left(4n^2+16n+19\right)p_1 + \left(12n^2+48n+45\right)p_2 &= 16n^2+64n+61
\end{aligned}
$$

We now compute the expected waiting time for both S^n and the optimal frequency-division schedule, S_{FDopt}, at these boundaries. We consider the case where $p_2 \le .1\,(1-p_1)$ OR $p_2 \ge .9\,(1-p_1)$. By symmetry, and concavity of $t_{freq} - t_{time}$, we need only consider the two boundary conditions $p_2 = 1 - p_1$ and $p_2 = .9\,(1-p_1)$. We first

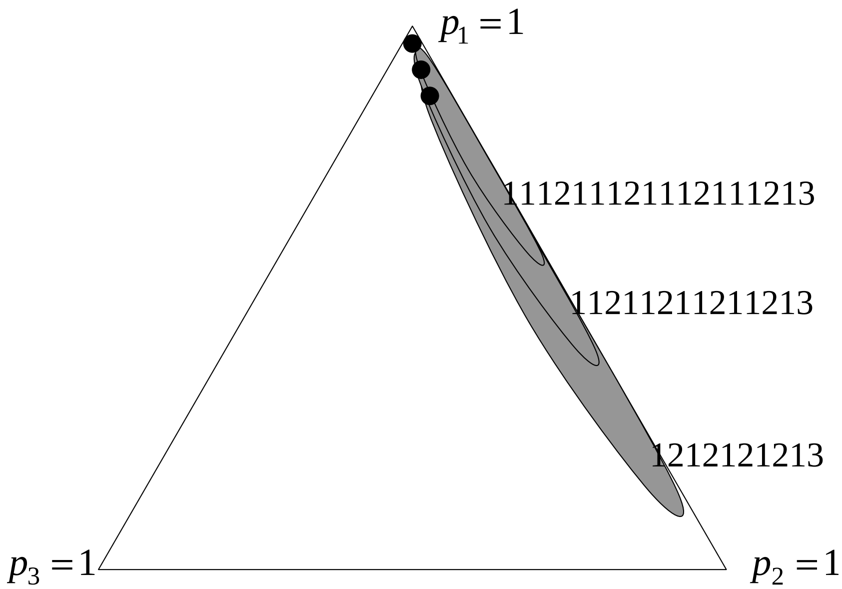

Figure 2.12: Regions where time-division schedules of the form $1^n21^n21^n21^n213$ have lower expected waiting time than the optimal frequency-division schedules.

examine the case where $p_2 = 1 - p_1$. We let

$$\begin{aligned} 4\left(4n^2+16n+19\right)p_1 + \left(12n^2+48n+45\right)p_2 &= 16n^2+64n+61 \\ p_2 &= 1-p_1 \\ p_3 &= 0 \end{aligned}$$

This gives us $\vec{p} = (p_1, p_2, p_3)$, where

$$\begin{aligned} p_1 &= \frac{4n^2+16n+16}{4n^2+16n+31} \\ p_2 &= \frac{15}{4n^2+16n+31} \\ p_3 &= 0 \end{aligned}$$

$$
\begin{aligned}
t_{time}^{n=n+1} = EWT(S^n, \vec{p}) &= \frac{15\,(n+2)}{4n^2+16n+31} \\
t_{freq}^{n=n+1} = EWT(S_{FDopt}, \vec{p}) &= 2\sqrt{\frac{15\,(4n^2+16n+16)}{(4n^2+16n+31)^2}} \\
&= \frac{4\sqrt{15}\,(n+2)}{4n^2+16n+31}
\end{aligned}
$$

We compare the expected waiting times for the time-division and frequency-division schedules and see that the time-division schedules are always better.

$$
\begin{aligned}
t_{freq}^{n=n+1} \geq t_{time}^{n=n+1} &\iff \frac{4\sqrt{15}\,(n+2)}{4n^2+16n+31} \geq \frac{15\,(n+2)}{4n^2+16n+31} \\
&\iff 4\sqrt{15} \geq 15 \\
&\iff 240 \geq 225 \longleftarrow \text{ true } \forall n \geq 1
\end{aligned}
$$

So, we have shown the first boundary condition.

Now we show the second, where $p_2 = .9\,(1-p_1)$. Let

$$
\begin{aligned}
4\left(4n^2+16n+19\right)p_1 + \left(12n^2+48n+45\right)p_2 &= 16n^2+64n+61 \\
p_2 &= \frac{9}{10}\,(1-p_1) \\
p_3 &= \frac{1}{10}\,(1-p_1)
\end{aligned}
$$

This gives us $\vec{p} = (p_1, p_2, p_3)$, where

$$
\begin{aligned}
p_1 &= 1 - \frac{150}{52n^2+208n+355} \\
p_2 &= \frac{135}{52n^2+208n+355} \\
p_3 &= \frac{15}{52n^2+208n+355}
\end{aligned}
$$

$$
\begin{aligned}
t_{time}^{n=n+1} = EWT(S^n, \vec{p}) &= \frac{195\,(n+2)}{52n^2+208n+355} \\
t_{freq}^{n=n+1} = EWT(S_{FDopt}, \vec{p}) &= 2\sqrt{\left(1-\frac{150}{52n^2+208n+355}\right)\left(\frac{135}{52n^2+208n+355}\right)} \\
&\quad +2\sqrt{\left(1-\frac{150}{52n^2+208n+355}\right)\left(\frac{15}{52n^2+208n+355}\right)} \\
&\quad +2\sqrt{\left(\frac{135}{52n^2+208n+355}\right)\left(\frac{15}{52n^2+208n+355}\right)} \\
&= \frac{90}{52n^2+208n+355} \\
&\quad +\frac{8}{52n^2+208n+355}\sqrt{15\,(52n^2+208n+205)} \\
&= \frac{90+8\sqrt{15\,(52n^2+208n+205)}}{52n^2+208n+355}
\end{aligned}
$$

We compare the expected waiting times for the time-division and frequency-division schedules and see that the time-division schedules are always better.

$$
\begin{aligned}
t_{freq}^{n=n+1} \geq t_{time}^{n=n+1} &\iff \frac{90+8\sqrt{15\,(52n^2+208n+205)}}{52n^2+208n+355} \geq \frac{195\,(n+2)}{52n^2+208n+355} \\
&\iff 11895n^2+82680n+106800 \geq 0 \longleftarrow \text{ true } \forall n \geq 1
\end{aligned}
$$

We have shown the second boundary condition. By symmetry and concavity, we see that there is a time-division schedule of the form $1^n21^n21^n21^n213$ that is better than the optimal frequency-division schedule when $p_2 \leq .1\,(1-p_1)$ OR $p_2 \geq .9\,(1-p_1)$.

We have shown that time-division is better than frequency-division for the small triangular regions indicated in Figure 2.13. By symmetry, it follows that time-division is better than frequency-division for all values of p_1, p_2, and p_3.

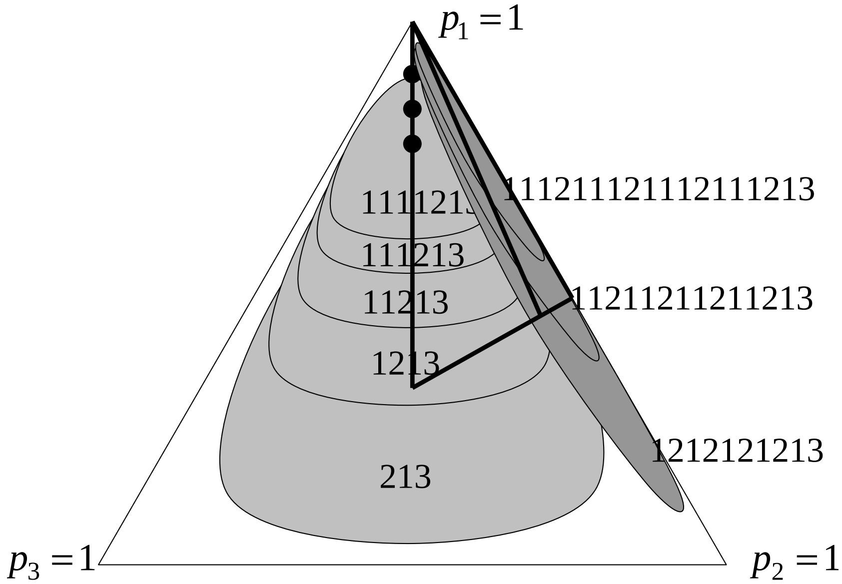

Figure 2.13: Regions we have examined, and the time-division schedules that have lower expected waiting time than optimal frequency-division schedules in those regions.

Chapter 3 Time-Division Schedules

We now consider time-division schedules exclusively. We showed that optimal frequency-division scheduling is often not very difficult, but time-division schedules are better than frequency-division schedules in many cases. Now, we address the problem of time-division scheduling. We look at the case where we have two items to schedule, and we find the optimal time-division schedules for this case, as a function of the lengths of the items and their demands.

In Section 3.1, we present Theorem 3.1, which tells us the optimal time-division schedules. In Section 3.2, we prove some lemmas. Then, in Section 3.3, we use the lemmas to prove Theorem 3.1.

3.1 Introduction

For these schedules, we can adopt a simpler notation for describing schedules. We will generally write a sequence of numbers, possibly with exponents, to describe a schedule. The numbers indicate the sequence of items to send, and the exponents indicate how long to spend sending that item. An exponent of 1 means to send the item exactly once. Integer exponents of 2, 3, 4, etc. indicate that the item is to be sent that number of times consecutively. The length of a schedule can be computed as the sum of the lengths of the pieces in its description.

For example, the schedule 121 means that we send item 1, then item 2, then item 1, and repeat this pattern indefinitely. The two schedules $1^2 2^3$ and 11222 are identical. In each we alternate between sending item 1 twice and item 2 three times. Since schedules are cyclic, any one of 12221, 22211, 22112, and 21122 could represent this schedule as well. For this schedule, if $l_1 = 5$ and $l_2 = 7$, then the length of the schedule is $2 \cdot 5 + 3 \cdot 7 = 31$.

We discuss the splitting of items into smaller pieces. When we split an item, we assume that it consists of a number of independent parts that can be sent independently. For example, if item 1 contains 10 stock quotes as 10 pairs of the form $(ticker, price)$, then we could split item 1 into 10 pieces, where each piece contained one such pair.

When we split an item into k equal-sized pieces, we write schedules in terms of the pieces, so the schedule $1^2 2^3$ in this context would mean to send two pieces of item 1 followed by three pieces of item 2, and repeat. Different periods of this schedule may send different pieces of the items, depending on which piece is next to be sent. For example, if we split item 1 into two pieces 1_A and 1_B, and split item 2 into two pieces 2_A and 2_B, then the schedule $1^2 2^3$ means we send the pieces in the following order: $1_A 1_B 2_A 2_B 2_A 1_A 1_B 2_B 2_A 2_B 1_A 1_B 2_A 2_B 2_A \ldots$.

In this chapter, we will prove a lemma about splitting items, but our theorem about optimal scheduling does not involve splitting of items.

Theorem 3.1 *For two items of lengths $l_1 = 1$ and $l_2 = a$, the broadcast schedule that minimizes expected waiting time is*

$$1^1 2^1, \text{ if } p_1 \in \left[\frac{1}{2(a+1)}, 1 - \frac{a}{2(a+1)}\right]$$
$$1^1 2^n, \text{ if } p_1 \in \left[\frac{a+1}{n(n+1)a^2+2(n+1)a+2}, \frac{a+1}{n(n-1)a^2+2na+2}\right)$$
$$1^n 2^1, \text{ if } p_1 \in \left(1 - \frac{a(a+1)}{2a^2+2na+n(n-1)}, 1 - \frac{a(a+1)}{2a^2+2(n+1)a+n(n+1)}\right]$$

3.2 The Lemmas

Before we prove this theorem, we prove two lemmas that will be useful in the proof of this theorem. In the first lemma, the length of a schedule is the sum of the lengths of its pieces.

Lemma 3.1 *(The Splitting Lemma)*

Split item i into n_i pieces of length $\frac{l_i}{n_i}$, $\forall i, 1 \le i \le n$. Suppose time-division schedule S is written as $s_1 s_2 \ldots s_l$, where each $s_i \in \{1, 2, \ldots, n\}$, and represents the broadcast of one piece of item $1, 2, \ldots,$ or n, respectively. Suppose also that there is a schedule $C = c_1 c_2 \ldots c_k$, such that C has at least n_i i's $\forall i, 1 \le i \le n$, and $c_j =$

$s_{g_1+j \ mod \ l} = s_{g_2+j \ mod \ l}$ $\forall j$, $1 \le j \le k$, for some $g_1 \neq g_2$ mod l. Then $EWT(S,\, p_1)$ $= \frac{l_{S_1}}{l_S} \cdot EWT(S_1,\, p_1) + \frac{l_{S_2}}{l_S} \cdot EWT(S_2,\, p_1)$, where $S_1 = s_{(g_1+1)\, mod\, l}\, s_{(g_1+2)\, mod\, l} \ldots s_{g_2\, mod\, l}$, $S_2 = s_{(g_2+1)\, mod\, l}\, s_{(g_2+2)\, mod\, l} \ldots s_{g_1\, mod\, l}$, l_{S_1} = length of schedule S_1, l_{S_2} = length of schedule S_2, and $l_S = l_{S_1} + l_{S_2}$ = length of schedule S.

Proof of Lemma 3.1

Since S repeats cyclically, we can assume without loss of generality that it starts with one of the sub-schedules C. We can write schedule S as

$$(*)\ \overbrace{\overbrace{|C_{1(start)} \cdots\cdots}^{S_1}\overbrace{|C_{2(start)} \cdots\cdots}^{S_2}}^{S}\overbrace{|C_{1(start)} \cdots\cdots}^{S_1},$$

where $C_{1(start)}$ indicates the start of the first instance of sub-schedule C within S, $C_{2(start)}$ indicates the start of the second occurrence, and the dots represent everything else.

We can also write schedules S_1 and S_2 as follows:

$$(**)\ \overbrace{|C_{(start)} \cdots\cdots}^{S_1}\overbrace{|C_{(start)} \cdots\cdots}^{S_1}$$

$$(***)\ \overbrace{|C_{(start)} \cdots\cdots}^{S_2}\overbrace{|C_{(start)} \cdots\cdots}^{S_2}$$

If we start waiting at some time within the first S_1 group in either (*) or (**), we will wait through a certain number of pieces before finding enough pieces of the desired item. If we receive all needed data within the initial S_1 group, the times are identical for (*) and (**) since the sequence of pieces that we encounter is the same in each case. If we must proceed into the next group, we first enter the C sub-schedule in either case. Since C contains n_i pieces of item i $\forall i, 1 \le i \le n$, we will not have to advance past the C group, so the waiting times in these cases are identical, since we again encounter the same sequence of pieces. So, it follows that the waiting time for schedule S, given that we arrive somewhere within sub-schedule S_1, is the same as it is if we use S_1 as our schedule. The same applies for sub-schedule S_2. So, we have $EWT(S,p_1) = Pr(arrive\ in\ S_1)\cdot EWT(S_1,p_1) + Pr(arrive\ in\ S_2)\cdot EWT(S_2,p_1)$. The probabilities of arriving in sub-schedules S_1 and S_2 within schedule S are simply $\frac{l_{S_1}}{l_S}$ and $\frac{l_{S_2}}{l_S}$, respectively. So, we have $EWT(S,p_1) = \frac{l_{S_1}}{l_S}\cdot EWT(S_1,p_1) + \frac{l_{S_2}}{l_S}\cdot EWT(S_2,p_1)$.

For this section we will use Lemma 3.1 only with $n_1 = n_2 = 1$. In Chapter 4, we will consider $n_i > 1$. For Lemma 3.2, we do not consider splitting.

Lemma 3.2 *Let schedule* $S = 1^{m+2}2^{n+2}$ *and schedule* $S' = 1^{m+1}212^{n+1}$*, where* $l_1 = 1, l_2 = a > 0, m \geq 0, n \geq 0$. *Then* $EWT(S',p_1) < EWT(S,p_1)\, \forall a,m,n$.

Proof of Lemma 3.2

We compute EWT_1 and EWT_2 by first computing the expected waiting time for the items given that we arrive during different parts of the schedule. Then we weight these times by the probability that we arrive during that part of the schedule.

$$\begin{aligned}
EWT_1(S) &= \frac{1}{m+2+a(n+2)} \cdot \left(m\left(\frac{1}{2}\right) + 1\left(\frac{1}{2}\right) + 1\left(\frac{1}{2} + 2a + na\right) \right. \\
&\quad \left. + a\left(\frac{a}{2} + a + na\right) + a\left(\frac{a}{2} + na\right) + na\left(\frac{na}{2}\right)\right) \\
EWT_1(S') &= \frac{1}{m+2+a(n+2)} \cdot \left(m\left(\frac{1}{2}\right) + 1\left(\frac{1}{2} + a\right) + a\left(\frac{a}{2}\right) \right. \\
&\quad \left. +1\left(\frac{1}{2} + a + na\right) + a\left(\frac{a}{2} + na\right) + na\left(\frac{na}{2}\right)\right) \\
\Delta EWT_1 &= EWT_1(S) - EWT_1(S') \\
&= \frac{m\,(0) + 1\,(-a) + 1\,(a) + a\,(a+na) + a\,(0) + na\,(0)}{m+2+a(n+2)} \\
&= \frac{a^2 + na^2}{m+2+a(n+2)} > 0\, \forall a,m,n
\end{aligned}$$

$$\begin{aligned}
EWT_2(S) &= \frac{m\left(\frac{m}{2}+2\right)+1\left(\frac{1}{2}+1\right)+1\left(\frac{1}{2}\right)+na\left(\frac{a}{2}\right)+a\left(\frac{a}{2}\right)+a\left(\frac{a}{2}+m+2\right)}{m+2+a(n+2)} \\
EWT_2(S') &= \frac{m\left(\frac{m}{2}+1\right)+1\left(\frac{1}{2}\right)+a\left(\frac{a}{2}+1\right)+1\left(\frac{1}{2}\right)+na\left(\frac{a}{2}\right)+a\left(\frac{a}{2}+m+1\right)}{m+2+a(n+2)} \\
\Delta EWT_2 &= EWT_2(S)-EWT_2(S') \\
&= \frac{m\,(1)+1\,(1)+1\,(0)+a\,(-1)+na\,(0)+a\,(1)}{m+2+a(n+2)} \\
&= \frac{m+1}{m+2+a(n+2)} > 0\,\forall a,m,n \\
\Delta EWT(p_1) &= EWT(S,p_1)-EWT(S',p_1) \\
&= \Delta EWT_1 p_1 + \Delta EWT_2(1-p_1) > 0\,\forall a,m,n,p_1
\end{aligned}$$

So, it follows that $EWT(S',p_1) < EWT(S,p_1)\,\forall a,m,n,p_1$.

3.3 Proof of Theorem 3.1

First we note that it is never optimal to have a schedule with two occurrences of 12 or 21. This follows from Lemma 3.1. So the optimal schedule is always a sequence of 1's followed by a sequence of 2's. But if we have the sequence 1122, Lemma 3.2 tells us that it is better to replace it with 1212. So, the optimal schedule is one of 12^n or 1^n2. The expected waiting time for 12^n can be computed as:

$$EWT(12^n,p_1) = (n-1)\frac{na^2+2a}{2(na+1)}p_1 + \frac{na^2+2a+1}{2(na+1)}$$

We see that $EWT(12^n,p_1) = EWT(12^{n+1},p_1)$ at $p_1 = p_1^{12^{n,n+1}} = \frac{a+1}{n(n+1)a^2+2(n+1)a+2}$.

The expected waiting time for 1^n2 is

$$EWT(1^n2,p_1) = (n-1)\frac{n+2a}{2(n+a)}(1-p_1) + \frac{a^2+2a+n}{2(n+a)}$$

We see that $EWT(1^n2,p_1) = EWT(1^{n+1}2,p_1)$ at $p_1 = p_1^{1^{n,n+1}2} = 1 - \frac{a^2+a}{2a^2+2(n+1)a+n(n+1)}$.

To show that these schedules are optimal on the indicated ranges, we must show that no other schedule is better than 12^n or 1^n2 at $p_1^{12^{n,n+1}}$ or $p_1^{1^{n,n+1}2}$, respectively, for all $n \geq 1$. We first compare similar schedules with different values for n. We look at $p_1^{12^{n,n+1}}$ and compare the schedules 12^n and 12^{n+x}, where x is any integer value. We find the values of x such that 12^{n+x} has lower expected waiting time than 12^n.

$$\begin{aligned} \Delta EWT &= EWT\left(12^{n+x}, p_1^{12^{n,n+1}}\right) - EWT\left(12^n, p_1^{12^{n,n+1}}\right) \\ &= \frac{a^2(a+1)x(x-1)}{2((n+x)a+1)(n(n+1)a^2+2(n+1)a+2)} \\ &\leq 0 \Longleftrightarrow 0 \leq x \leq 1 \end{aligned}$$

We see that the only schedules that are not worse than 12^n at $p_1^{12^{n,n+1}}$ are 12^{n+0} and 12^{n+1}. So, all other schedules have higher expected waiting time at $p_1^{12^{n,n+1}}$ than these two. Now, we consider schedules of the form 1^m2.

$$\begin{aligned} \Delta EWT &= EWT\left(1^m2, p_1^{12^{n,n+1}}\right) - EWT\left(12^n, p_1^{12^{n,n+1}}\right) \\ &= \tfrac{1}{2(m+a)(na+1)(n(n+1)a^2+2(n+1)a+2)} \cdot \left(\left(n^2(m(n+1)-2)\right)a^4\right. \\ &\quad + (n(n+1)(m(mn+1)-2))a^3 \\ &\quad + \left(2mn^2(m-1) + mn(mn-1) + 2n\left(m^2-1\right)\right)a^2 \\ &\quad + (m(m-1)(3n+1))a + (m(m-1))) \\ &\geq 0 \;\forall m, n \geq 1 \end{aligned}$$

We look at the equation term by term to see that each coefficient of a is always greater than or equal to 0, with equality exactly when $m = n = 1$. So, no schedules of the form 1^m2 are better than 12^n at $p_1^{12^{n,n+1}}$, except that 1^12 has the same expected waiting time as 12^1. This is expected, since they are the same schedule.

Similarly, we can show that the best schedules at $p_1^{1^{n,n+1}2}$ are 1^n2 and $1^{n+1}2$. So, the endpoints for the range where 12^n is optimal are $p_1^{12^{n-1,n}}$ and $p_1^{12^{n,n+1}}$ for $n \geq 2$. The endpoints for the range where 1^n2 is optimal are $p_1^{1^{n-1,n}2}$ and $p_1^{1^{n,n+1}2}$ for $n \geq 2$. For 12, the optimal range is just the region between $p_1^{1^{1,2}2}$ and $p_1^{12^{1,2}}$. Substituting the

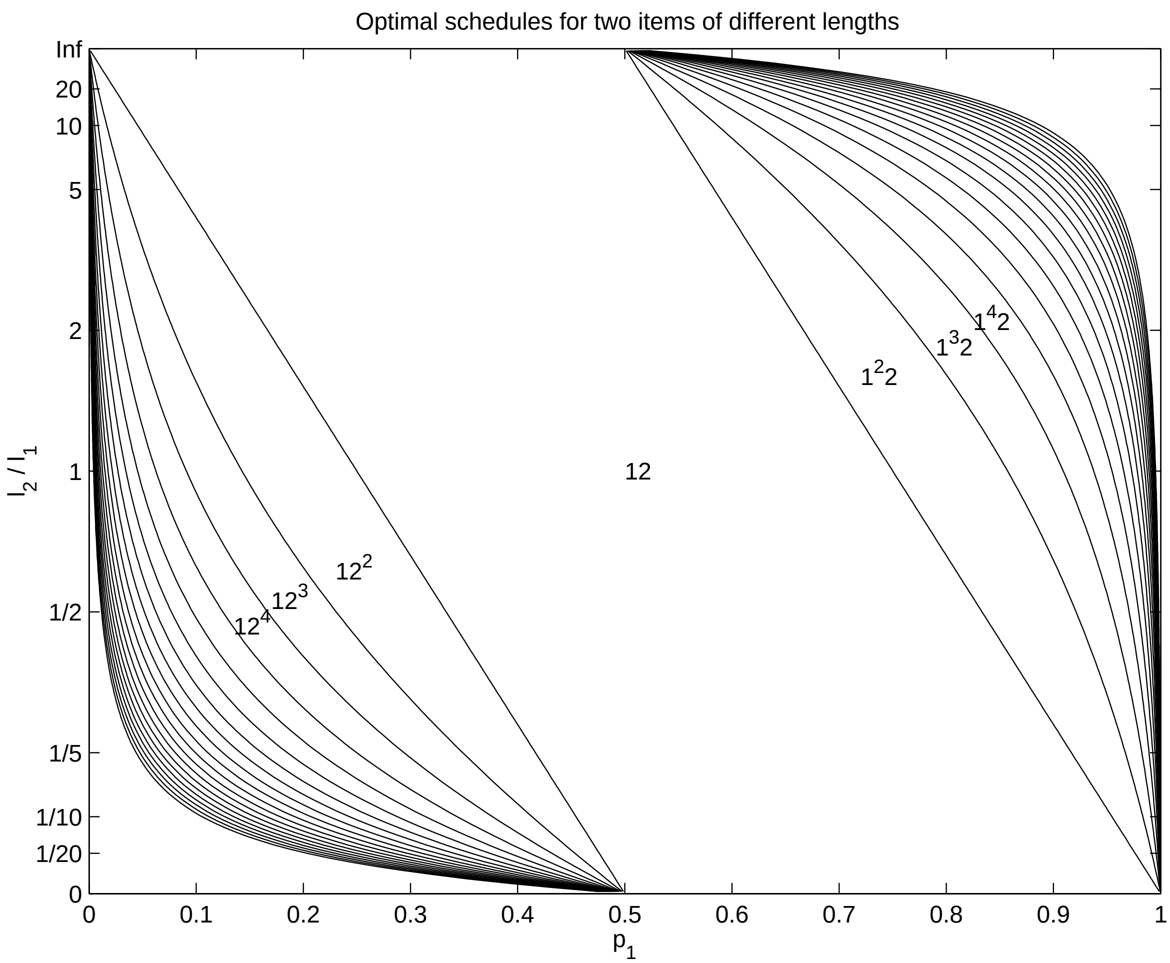

Figure 3.1: Regions where 12^n and 1^n2 are optimal, as a function of p_1 and a, for $n \leq 20$. In this plot, n increases in the regions to the lower left and upper right.

expressions for the $p_1^{12^{n,n+1}}$'s and $p_1^{1^{n,n+1}2}$'s, we see that the theorem is proved. Figure 3.1 shows us the values of p_1 and $\frac{l_2}{l_1} = a$ where each schedule is optimal, for $n \leq 20$. For the regions to the lower left and upper right, we use schedules with increasing values of n.

Chapter 4 Time-Division Scheduling with One Split

We have shown how to schedule optimally for two dynamic items using time-division scheduling. We now examine schedules for two items that we can split in half. We assume that we have the freedom to split our two items into two pieces of equal size and schedule these pieces. Section 4.1 describes this new idea and some new notation and presents the theorem describing the optimal schedules. In Section 4.2, we present two lemmas and describe how these two lemmas and Lemma 3.1 will be useful in manipulating schedules to find the optimal schedules. In Sections 4.3 and 4.4, we prove these lemmas. In Section 4.5, we use the lemmas to find a set of irreducible schedules. Then in Section 4.6, we compare these irreducible schedules and find the optimal schedules. In Section 4.7, we consider two items of different lengths and show that much of the work for items of equal lengths generalizes to arbitrary lengths.

4.1 Introduction

We examine the scheduling of two items for a broadcast disk. These items will be dynamic and split into two equal-size pieces. We can receive pieces out of order, but each piece must be received from start to finish. In this section, we think of each item as consisting of two halves, and a 1 or 2 will represent one of these halves, not the entire item. For example, a schedule in which the two items are broadcast one after the other in their entirety is 112211221122… and not 121212…, since we need two halves of each item to broadcast the entire item.

Broadcast schedules are cyclic, so we will represent them by one of their cycles. Since each item has two halves, we assume these halves are broadcast in an alternating way. The schedule 122, for example, should really be written as $1_A 2_A 2_B 1_B 2_A 2_B$, where

1_A and 1_B are the two halves of item 1, and 2_A and 2_B are the two halves of item 2. This would more accurately represent one period, but we shorten the representation to 122 with the understanding that the two halves of each item are broadcast in an alternating way.

Another representation of a schedule that we will use is based on the number of 2's between consecutive 1's in the schedule. We use a bracketed sequence of numbers that represent the number of 2's between each consecutive pair of 1's. For example, [0,2] represents the schedule 1122, and [0,0,1,3,2] represents 11121222122. We use the notation S^C to represent the complement of S, the schedule S with 1's and 2's swapped. For example, if $S = 12211112$, then $S^C = 21122221$. We use S^R to represent the reversal of S. With S as above, $S^R = 21111221$.

Sometimes we want to indicate that a certain instance of an item may or may not be present in a schedule. Parentheses will be used to indicate the possible presence of an instance of an item in a schedule. For example, a schedule in which we know only that item 2 is never broadcast twice consecutively is represented by $1(2)1(2)\ldots1(2)$.

We define a new notion of expected waiting time to deal with 2 split items.

Definition 4.1 *$EWT_S(S, p_1)$ is the expected waiting time using schedule S with demand probabilities p_1 and $p_2 = 1 - p_1$, for two items, each split into two halves.*

The mathematical definition is exactly the same as for no splits, except we allow the value of the integrals in the definitions of starting point and ending point to assume values that are half integer multiples of l_i, not just integer multiples of l_i. This reflects the idea that we can start and end at the start and end of the pieces (of size $\frac{1}{2} \cdot l_i$), not just the start and end of the items. The subscript S indicates that this is an expected waiting time with splitting.

We compute waiting time for an item by looking at how long a client must wait to get all of the desired item, starting in a particular section of the broadcast cycle. We average over all sections to get the expected waiting time for that item. We do this for each item and weight these times by the demand probabilities to get the overall expected waiting time.

We show the following:

Theorem 4.1 *For two items of the same length, each split into two halves, the broadcast schedule that minimizes expected waiting time is*

$$11222^n, n = Max\left(0, \left\lceil \frac{-9+\sqrt{-47+\frac{40}{p_1}}}{2} \right\rceil\right), \ \ \textit{if } p_1 \in \left[\frac{1}{5}, \frac{1}{2}\right]$$

$$1221222^n, n = Max\left(0, \left\lceil \frac{-13+\sqrt{-103+\frac{32}{p_1}}}{2} \right\rceil\right), \ \ \textit{if } p_1 \in \left(0, \frac{1}{5}\right)$$

In Section 4.2, we present lemmas useful in proving Theorem 4.1. We discuss the implications of each lemma. We describe how to reduce the search for optimal schedules from all schedules to a smaller set of irreducible schedules. We determine which schedules are in this set. Then, we compare these with each other and see that the schedules listed in Theorem 4.1 are optimal on their respective intervals. We also show the optimal schedules without splitting, for comparison.

In Section 4.7, we consider items of different lengths. We present numerical results that suggest that the optimal schedules for different length items are the same as those for items of the same length.

4.2 The Lemmas

These lemmas provide ways of manipulating schedules into similar but better schedules. In Lemmas 4.1 and 4.2, we assume two items, each split into two halves, with $l_1 = 1$ and $l_2 = a$.

Lemma 4.1 *(The Rearrangement Lemma)*

The following hold for all values of p_1:

(a)

$$\begin{aligned} & EWT_S(S', p_1) \leq EWT_S(S, p_1), \textit{ where} \\ S &= [a, b, c, \ldots, d, e, f, \ldots], \\ S' &= [a, b-1, c, \ldots, d, e+1, f, \ldots], \textit{ and} \\ & (a+b+c) - (d+e+f) \geq 1 \end{aligned}$$

(b)

$$\begin{aligned} & EWT_S(T', p_1) \leq EWT_S(T, p_1), \textit{ where} \\ T &= \ldots 12^n 1212^x 12^y 1 \ldots, \\ T' &= \ldots 12^n 1122^x 12^y 1 \ldots, \textit{ and} \\ & n \geq 2, x, y \in \{0, 1\} \end{aligned}$$

(c)

$$\begin{aligned} & EWT_S(U', p_1) \leq EWT_S(U, p_1), \textit{ where} \\ U &= \ldots 1^r 2^s 1^t 2^u 1 \ldots, \\ U' &= \ldots 1^r 2^s 1^{t-1} 2^{u-1} 1 \ldots, \textit{ and} \\ & r \geq 1, s \geq 1, t \geq 3, u \geq 3 \end{aligned}$$

(d)

$$\begin{aligned} & EWT_S(V', p_1) = EWT_S(V, p_1), \textit{ where} \\ V &= 1^{a_1} 2^{b_1} 1^{a_2} 2^{b_2} \ldots 1^{a_{k-1}} 2^{b_{k-1}} 1^{a_k} 2^{b_k}, \\ V' &= 2^{b_k} 1^{a_k} 2^{b_{k-1}} 1^{a_{k-1}} \ldots 2^{b_2} 1^{a_2} 2^{b_1} 1^{a_1} \end{aligned}$$

Lemma 4.2 *(Corollary to Lemma 4.1 part (a)) If, instead of $S = [a, b, c, \ldots, d, e, f, \ldots]$, S is one of $[a,\ b,\ c = d,\ e,\ f,\ \ldots]$, $[a,\ b = d,\ c = e,\ f,\ \ldots]$, $[a = f,\ b,\ c = d,\ e]$, or $[a = e,\ b = f,\ c = d]$, then Lemma 4.1 part (a) still holds, where S' is S with b decremented by 1 and e incremented by 1.*

Part (a) and its corollary tell us that it is generally not good to have strings of 2's of significantly different lengths. More specifically, for each string of two or more 2's, we add its length to the sum of the lengths of the adjacent strings of 2's. These sums should be as close to each other as possible (equal or within 1) for all strings of two or more 2's in the schedule.

For example, for the schedule 1212222211212212222222121, consider the two long strings of 2's of lengths 5 and 7. Their adjacent strings of 2's have lengths 1 and 0, and 2 and 1, respectively, giving sums of 5+1+0=6 and 7+2+1=10. Since $10 - 6 = 4 \geq 1$, we can move a 2 from the string of length 7 to the string of length 5, giving the new and better schedule 1212222221121221222222121, with sums 6+1+0=7 and 6+2+1=9. Since our new sums differ by 2, we can move another 2 to get an even better schedule 1212222222112122122222121 with sums 8 and 8. However, moving another 2 will not give us a better schedule, since $8 - 8 = 0 \not\geq 1$.

The corollary allows us to eliminate adjacent long strings of 2's. We do this by performing the operation in the lemma. We call our adjacent strings the b- and e-length strings. After multiple applications of the lemma (move a 2 from one of these strings to the other), we get one of the strings down to length 2.

As an example, consider the schedule 1212222212222112. We have adjacent strings of 2's of lengths 5 and 4. By considering these strings as we did above, we get sums of 5+1+4=10 and 4+5+0=9. So, if we move a 2 from the string of length 5 to the string of length 4, we improve our schedule. However, unlike before, our sums remain the same at 4+1+5=10 and 5+4+0=9, so we can repeat this procedure until we are left with only two 2's in the first string and a new schedule of 1212212222222112.

Part (b) applies when we have blocks of at least four 1's, possibly with some single 2's in them, bounded on both sides by at least two 2's. This lemma tells us that if the beginning or end of the block is 121, it is better to shift the 2 toward the inside of the block to get 112[*rest of block*] or [*rest of block*]211.

An example of this is the schedule 12212111212112222. This contains the string 1211121211, which is a block of seven 1's with only single 2's, bordered on both sides by 22. Part (b) of the lemma tells us that it is better to have 112 than 121 at the beginning of this string. It is better to rearrange the start of the string to get 1121121211, for a new and better schedule of 12211211212112222.

Part (c) tells us that large strings of 1's and 2's should not border each other. It is better to swap the innermost 1 and 2 if we have at least three of each. A simple example of this is the schedule 121111122212. We would be better off using 121111212212.

Part (d) tells us that reversing a schedule does not affect its expected waiting time. So, the schedules 121112222 and 222211121, for example, have the same expected waiting times.

We will also use Lemma 3.1, the Splitting Lemma. This lemma tells us that under certain conditions we can split a schedule S into two schedules, S_1 and S_2. Schedule S will have an expected waiting time that is the weighted mean of the expected waiting times of S_1 and S_2. At any value of p_1, one of S_1 and S_2 will have a shorter expected waiting time and the other will have a longer expected waiting time. As a result, we should not use S, but instead choose the better of S_1 and S_2.

As an example, consider the schedule 11212122221122. We rewrite it (by shifting, since we send data cyclically) as 12121222211221. This is just 12 concatenated with 121222211221. Each of these starts with 1212, which is a string with two 1's and two 2's. It's easy to see this for the second sub-schedule. For the first one, think of it not as 12, but as 121212..., which is what we broadcast when we use this schedule. We can split our schedule into these two smaller schedules. We can find the point p where the two schedules have the same expected waiting time, and see that 12 is better for $p_1 > p$ and 121222211221 is better for $p_1 < p$. So, instead of using the original schedule 11212122221122, we should use either 12 or 121222211221, depending on the value of p_1 compared to p.

4.3 Proof of Lemma 4.1

4.3.1 Lemma 4.1 Part (a)

$S = \cdots 12^a12^b12^c1\cdots 12^d12^e12^f1\cdots$, $S' = \cdots 12^a12^{b-1}12^c1\cdots 12^d12^{e+1}12^f1\cdots$, $b \geq 3, e \geq 2$.

S and S' have the same length, l_S. To show $EWT_S(S', p_1) \leq EWT_S(S, p_1)\,\forall p_1$, we show $\Delta t = EWT_S(S, p_1) - EWT_S(S', p_1) \geq 0\,\forall p_1$. We compute Δt as $p_1\Delta t_1 + p_2\Delta t_2 = p_1\,(t_1 - t_1') + p_2\,(t_2 - t_2')$, where t_1 and t_2 are the waiting times for items 1 and 2 using schedule S, and t_1' and t_2' are the waiting times for items 1 and 2 using schedule S'. The algebra to show this follows. In each expression, we have a sum of terms of the form $length(time_1 + \ldots + time_n)$. The *length* value corresponds to arriving in a section of the schedule with a length of *length*. The sum of the times is the sum of the lengths of the undesired sections that one waits through before getting all of the desired item, when arriving in the section with a length of *length*. Adding all terms of this form and dividing by the total length of the schedule gives us the waiting time for an item.

$$\begin{aligned}
t_1 &= \frac{1}{l_S}\left[\cdots + 1\left(\frac{1}{2} + al_2 + bl_2\right) + al_2\left(\frac{al_2}{2} + bl_2\right) + 1\left(\frac{1}{2} + bl_2 + cl_2\right) + \cdots\right. \\
&\quad + bl_2\left(\frac{bl_2}{2} + cl_2\right) + 1\left(\frac{1}{2} + dl_2 + el_2\right) + dl_2\left(\frac{dl_2}{2} + el_2\right) \\
&\quad \left. + 1\left(\frac{1}{2} + el_2 + fl_2\right) + el_2\left(\frac{el_2}{2} + fl_2\right) + \cdots\right] \\
t_1' &= \frac{1}{l_S}\left[\cdots + 1\left(\frac{1}{2} + al_2 + bl_2 - l_2\right) + al_2\left(\frac{al_2}{2} + bl_2 - l_2\right)\right. \\
&\quad + 1\left(\frac{1}{2} + bl_2 + cl_2 - l_2\right) + bl_2\left(\frac{bl_2}{2} - \frac{l_2}{2} + cl_2\right) - l_2\left(\frac{bl_2}{2} - \frac{l_2}{2} + cl_2\right) + \cdots \\
&\quad + 1\left(\frac{1}{2} + dl_2 + el_2 + l_2\right) + dl_2\left(\frac{dl_2}{2} + el_2 + l_2\right) + 1\left(\frac{1}{2} + el_2 + fl_2 + l_2\right) \\
&\quad \left. + el_2\left(\frac{el_2}{2} + \frac{l_2}{2} + fl_2\right) + l_2\left(\frac{el_2}{2} + \frac{l_2}{2} + fl_2\right) + \cdots\right] \\
\Delta t_1 &= t_1 - t_1' \\
&= \frac{1}{l_S}\left[1(l_2) + al_2(l_2) + 1(l_2) + bl_2\left(\frac{l_2}{2}\right) + l_2\left(\frac{bl_2}{2} - \frac{l_2}{2} + cl_2\right)\right. \\
&\quad \left. + 1(-l_2) + dl_2(-l_2) + 1(-l_2) + el_2\left(-\frac{l_2}{2}\right) - l_2\left(\frac{el_2}{2} + \frac{l_2}{2} + fl_2\right)\right] \\
&= \frac{1}{l_S}\left[l_2^2(a + b + c - d - e - f - 1)\right]
\end{aligned}$$

It is easy to see that $t_2 = t_2'$, so $\Delta t_2 = 0$. It follows that $\Delta t \geq 0 \iff (a+b+c)-(d+e+f) \geq 1$. Thus, $EWT_S(S', p_1) \leq EWT_S(S, p_1) \iff (a+b+c)-(d+e+f) \geq 1$, and part (a) of the lemma is proved.

4.3.2 Lemma 4.1 Part (b)

$T = \cdots 12^n 1212^x 12^y 1 \cdots, T' = \cdots 12^n 1122^x 12^y 1 \cdots, n \geq 2, x, y \in \{0, 1\}$.

T and T' have the same length, l_T. To show $EWT_S(T', p_1) \leq EWT_S(T, p_1)\,\forall p_1$, we show $\Delta t = EWT_S(T, p_1) - EWT_S(T', p_1) \geq 0\,\forall p_1$. We compute Δt as $p_1\Delta t_1 + p_2\Delta t_2 = p_1(t_1 - t_1') + p_2(t_2 - t_2')$, where t_1 and t_2 are the waiting times for items 1 and 2 using schedule T, and t_1' and t_2' are the waiting times for items 1 and 2 using schedule T'. The algebra to show this follows. In each expression, we have a sum of terms of the form $length(time_1 + ... + time_n)$. The *length* value corresponds to arriving in a section of the schedule with a length of *length*. The sum of the times is the sum of the lengths of the undesired sections that one waits through before getting all of the desired item, when arriving in the section with a length of *length*. Adding all terms of this form and dividing by the total length of the schedule gives us the waiting time for an item.

$$
\begin{aligned}
t_1 &= \frac{1}{l_T}\left[\cdots + 1\left(\frac{1}{2} + nl_2 + l_2\right) + nl_2\left(\frac{nl_2}{2} + l_2\right) + 1\left(\frac{1}{2} + l_2 + xl_2\right)\right. \\
&\quad \left. + l_2\left(\frac{l_2}{2} + xl_2\right) + 1\left(\frac{1}{2} + xl_2 + yl_2\right) + \cdots\right] \\
t_1' &= \frac{1}{l_T}\left[\cdots + 1\left(\frac{1}{2} + nl_2\right) + nl_2\left(\frac{nl_2}{2}\right) + 1\left(\frac{1}{2} + l_2 + xl_2\right)\right. \\
&\quad \left. + 1\left(\frac{1}{2} + l_2 + xl_2 + yl_2\right) + l_2\left(\frac{l_2}{2} + xl_2 + yl_2\right) + \cdots\right] \\
\Delta t_1 &= t_1 - t_1' \\
&= \frac{1}{l_T}\left[1\left(l_2\right) + nl_2\left(l_2\right) + 1\left(0\right) + l_2\left(-yl_2\right) + 1\left(-l_2\right)\right] \\
&= \frac{1}{l_T}\left[\left(n - y\right) l_2^2\right] \\
t_2 &= \frac{1}{l_T}\left[\cdots + 1\left(\frac{1}{2}\right) + \left(n-2\right) l_2\left(\frac{l_2}{2}\right) + l_2\left(\frac{l_2}{2} + 1\right) + l_2\left(\frac{l_2}{2} + 1 + 1 + T_2\right)\right. \\
&\quad \left. + 1\left(\frac{1}{2} + 1 + T_2\right) + l_2\left(\frac{l_2}{2} + 1 + T_{22}\right) + 1\left(\frac{1}{2} + T_{22}\right) + \cdots\right] \\
t_2' &= \frac{1}{l_T}\left[\cdots + 1\left(\frac{1}{2}\right) + \left(n-2\right) l_2\left(\frac{l_2}{2}\right) + l_2\left(\frac{l_2}{2} + 1 + 1\right) + l_2\left(\frac{l_2}{2} + 1 + 1 + T_2\right)\right. \\
&\quad \left. + 1\left(\frac{1}{2} + 1 + T_2\right) + 1\left(\frac{1}{2} + T_2\right) + l_2\left(\frac{l_2}{2} + T_{22}\right) + \cdots\right] \\
\Delta t_2 &= t_2 - t_2' \\
&= \frac{1}{l_T}\left[1\left(0\right) + \left(n-2\right) l_2\left(0\right) + l_2\left(-1\right) + l_2\left(0\right) + 1\left(0\right) + l_2\left(1\right) + 1\left(T_{22} - T_2\right)\right] \\
&= \frac{1}{l_T}\left[T_{22} - T_2\right]
\end{aligned}
$$

Here T_2 is the time to get one piece of item 2 when we start listening at the beginning of "2^x" and T_{22} is the time to get two pieces of item 2 when we start listening at the beginning of "2^x". It is easy to see that $T_{22} \geq T_2$. Also, $n \geq y$, by the restrictions we placed on them. So, $\Delta t_1 \geq 0$ and $\Delta t_2 \geq 0$. Since $\Delta t = p_1 \Delta t_1 + p_2 \Delta t_2$, we see that $\Delta t \geq 0$, and part (b) is proved.

4.3.3 Lemma 4.1 Part (c)

$U = \cdots 21^r 2^s 1^t 2^u 1 \cdots, U' = 21^r 2^s 1^{t-1} 212^{u-1} 1 \cdots, r, s \geq 1, t, u \geq 3.$

U and U' have the same length, l_U. To show $EWT_S(U', p_1) \leq EWT_S(U, p_1) \, \forall p_1$, we show $\Delta t = EWT_S(U, p_1) - EWT_S(U', p_1) \geq 0 \, \forall p_1$. We compute Δt as $p_1 \Delta t_1 + p_2 \Delta t_2 = p_1(t_1 - t_1') + p_2(t_2 - t_2')$, where t_1 and t_2 are the waiting times for items 1 and 2 using schedule U, and t_1' and t_2' are the waiting times for items 1 and 2 using schedule U'. The algebra to show this follows. In each expression, we have a sum of terms of the form $length(time_1 + ... + time_n)$. The *length* value corresponds to arriving in a section of the schedule with a length of *length*. The sum of the times is the sum of the lengths of the undesired sections that one waits through before getting all of the desired item, when arriving in the section with a length of *length*. Adding all terms of this form and dividing by the total length of the schedule gives us the waiting time for an item.

$$\begin{aligned}
t_1 &= \frac{1}{l_U}\left[\cdots + l_2\left(\frac{l_2}{2} + (r=1)\, sl_2\right) + (r>2)\,(r-2)\left(\frac{1}{2}\right) + (r \geq 2)\left(\frac{1}{2} + sl_2\right)\right. \\
&\quad + \left(\frac{1}{2} + sl_2\right) + (s>2)\,(s-2)\, l_2\left(\frac{(s-2)\, l_2}{2} + 2l_2\right) + (s \geq 2)\, l_2\left(\frac{l_2}{2} + l_2\right) \\
&\quad + l_2\left(\frac{l_2}{2}\right) + (t-2)\left(\frac{1}{2}\right) + 1\left(\frac{1}{2} + ul_2\right) + 1\left(\frac{1}{2} + ul_2 + T_1\right) \\
&\quad \left. + ul_2\left(\frac{ul_2}{2} + T_1\right) + \cdots\right] \\
t_1' &= \frac{1}{l_U}\left[\cdots + l_2\left(\frac{l_2}{2} + (r=1)\, sl_2\right) + (r>2)\,(r-2)\left(\frac{1}{2}\right) + (r \geq 2)\left(\frac{1}{2} + sl_2\right)\right. \\
&\quad + \left(\frac{1}{2} + sl_2\right) + (s>2)\,(s-2)\, l_2\left(\frac{(s-2)\, l_2}{2} + 2l_2\right) + (s \geq 2)\, l_2\left(\frac{l_2}{2} + l_2\right) \\
&\quad + l_2\left(\frac{l_2}{2}\right) + (t-3)\left(\frac{1}{2}\right) + 1\left(\frac{1}{2} + l_2\right) + 1\left(\frac{1}{2} + ul_2\right) + l_2\left(\frac{l_2}{2} + ul_2 - l_2\right) \\
&\quad \left. + 1\left(\frac{1}{2} + ul_2 - l_2 + T_1\right) + (u-1)\, l_2\left(\frac{(u-1)\, l_2}{2} + T_1\right) + \cdots\right] \\
\Delta t_1 &= t_1 - t_1' \\
&= \frac{1}{l_U}\left[l_2\,(0) + (r>2)\,(r-2)\,(0) + (r \geq 2)\,(0) + (0) + (0) + (0) + l_2\,(0) + \frac{1}{2}\right. \\
&\quad + (u-1)\, l_2 + T_1 \\
&\quad + ul_2\left(\frac{l_2}{2}\right) + l_2\left(\frac{(u-1)\, l_2}{2} + T_1\right) - l_2\left(\frac{l_2}{2} + ul_2 - l_2\right) \\
&\quad \left. - 1\left(\frac{1}{2} + ul_2 - l_2 + T_1\right)\right] \\
&= \frac{1}{l_U}\left[l_2 T_1\right]
\end{aligned}$$

$$
\begin{aligned}
t_2 &= \frac{1}{l_U}\left[\cdots + l_2\left(\frac{l_2}{2} + r + (s=1)\,t\right) + r\left(\frac{r}{2} + (s=1)\,t\right)\right.\\
&\quad + (s>2)\,(s-2)\,l_2\left(\frac{l_2}{2}\right) + (s\geq 2)\,l_2\left(\frac{l_2}{2}+t\right) + l_2\left(\frac{l_2}{2}+t\right) + t\left(\frac{t}{2}\right)\\
&\quad \left. + l_2\left(\frac{l_2}{2}\right) + \cdots\right]\\
t_2' &= \frac{1}{l_U}\left[\cdots + l_2\left(\frac{l_2}{2} + r + (s=1)\,(t-1)\right) + r\left(\frac{r}{2} + (s=1)\,(t-1)\right)\right.\\
&\quad + (s>2)\,(s-2)\,l_2\left(\frac{l_2}{2}\right) + (s\geq 2)\,l_2\left(\frac{l_2}{2}+t-1\right) + l_2\left(\frac{l_2}{2}+t\right)\\
&\quad \left. + (t-1)\left(\frac{t-1}{2}+1\right) + l_2\left(\frac{l_2}{2}+1\right) + 1\left(\frac{1}{2}\right) + \cdots\right]\\
\Delta t_2 &= t_2 - t_2'\\
&= \frac{1}{l_U}\left[l_2\left((s=1)\,1\right) + r\left((s=1)\,(1)\right) + (s>2)\,(0) + (s\geq 2)\,l_2 + t - 2 - l_2\right]\\
&= \frac{1}{l_U}\left[(s\geq 2)\,l_2 + (s=1)\,(l_2+r) + t - 2 - l_2\right]\\
&= \frac{1}{l_U}\left[(s=1)\,r + t - 2\right]
\end{aligned}
$$

Here T_1 is the time to get one piece of item 1 when we start listening at the end of the piece of item 1 just after "2^u" in schedule U. This is the same as the time when we start listening at the end of the piece of item 1 just after "2^{u-1}" in schedule U'. Expressions such as $(r=1)$ and $(s>2)$ are evaluated as 1 if the expression in the parentheses is true and 0 if it is false. This is just a shorthand way of considering multiple cases with one equation. Since $T_1 \geq 0$, $r \geq 1$, and $t \geq 3$, we see that $\Delta t_1 \geq 0$ and $\Delta t_2 \geq 0$. Since $\Delta t = p_1\Delta t_1 + p_2\Delta t_2$, we see that $\Delta t \geq 0$, and part (c) is proved.

4.3.4 Lemma 4.1 Part (d)

$V = 1^{a_1}2^{b_1}1^{a_2}2^{b_2}\cdots 1^{a_{k-1}}2^{b_{k-1}}1^{a_k}2^{b_k}, V' = 2^{b_k}1^{a_k}2^{b_{k-1}}1^{a_{k-1}}\cdots 2^{b_2}1^{a_2}2^{b_1}1^{a_1}$. Without loss of generality, we assume all a_i's are 1. If they are greater than 1, we can simply replace 1^{a_i} with a_i copies of $1^1 2^0$.

V and V' have the same length, l_V. To show $EWT_S(V', p_1) = EWT_S(V, p_1)\,\forall p_1$, we show $\Delta t = EWT_S(V, p_1) - EWT_S(V', p_1) = 0\,\forall p_1$. We compute Δt as $p_1\Delta t_1 + p_2\Delta t_2 = p_1(t_1 - t_1') + p_2(t_2 - t_2')$, where t_1 and t_2 are the waiting times for items 1 and 2 using schedule V, and t'_1 and t'_2 are the waiting times for items 1 and 2 using schedule V'. The algebra to show this follows. In each expression, we have a sum of terms of the form $length(time_1 + ... + time_n)$. The $length$ value corresponds to arriving in a section of the schedule with a length of $length$. The sum of the times is the sum of the lengths of the undesired sections that one waits through before getting all of the desired item, when arriving in the section with a length of $length$. Adding all terms of this form and dividing by the total length of the schedule gives us the waiting time for an item.

$$\begin{aligned}
t_1 &= \frac{1}{l_V}\sum_{i=1}^{k}\left[1\left(\frac{1}{2} + b_i l_2 + b_{(i+1)mod\,k} l_2\right) + b_i l_2\left(\frac{b_i l_2}{2} + b_{(i+1)mod\,k} l_2\right)\right] \\
t'_1 &= \frac{1}{l_V}\sum_{i=1}^{k}\left[b_i l_2\left(\frac{b_i l_2}{2} + b_{(i-1)mod\,k} l_2\right) + 1\left(\frac{1}{2} + b_{(i-1)mod\,k} l_2 + b_{(i-2)mod\,k} l_2\right)\right] \\
\Delta t_1 &= t_1 - t'_1 \\
&= \frac{1}{l_V}\sum_{i=1}^{k}\left[l_2\left(b_{(i+1)mod\,k} + b_i - b_{(i-1)mod\,k} - b_{(i-2)mod\,k}\right)\right. \\
&\quad \left. + b_i l_2\left(b_{(i+1)mod\,k} l_2 - b_{(i-1)mod\,k} l_2\right)\right] \\
&= \frac{1}{l_V}\left[l_2\left(\sum_{i=1}^{k} b_{(i+1)mod\,k} + \sum_{i=1}^{k} b_i - \sum_{i=1}^{k} b_{(i-1)mod\,k} - \sum_{i=1}^{k} b_{(i-2)mod\,k}\right)\right. \\
&\quad \left. + b_i l_2^2\left(\sum_{i=1}^{k} b_{(i+1)mod\,k} - \sum_{i=1}^{k} b_{(i-1)mod\,k}\right)\right] \\
&= \frac{1}{l_V}\left[l_2\left(\sum_{i=1}^{k} b_i + \sum_{i=1}^{k} b_i - \sum_{i=1}^{k} b_i - \sum_{i=1}^{k} b_i\right) + b_i l_2^2\left(\sum_{i=1}^{k} b_i - \sum_{i=1}^{k} b_i\right)\right] \\
&= 0
\end{aligned}$$

Similarly, $\Delta t_2 = 0$. So, we have $\Delta t = 0$ and part (d) is proved.

4.4 Proof of Lemma 4.2

We use the result of Lemma 4.1 (a), plus an additional little trick. We write S as $SSSSSS$, repeating the schedule six times. This is still the same schedule, since we broadcast schedules repeatedly. Now, we choose the b-length section from the second S and the e-length section from the fifth S, and choose a and c adjacent to b, and d and f adjacent to e in the overall schedule. There is no overlap of these regions. We now apply Lemma 4.1 (a) to $SSSSSS$ to get $SS^{-}SSS^{+}S$. Here S^{-} is S with b decreased by one and S^{+} is S with e increased by one. We will write $S^{\pm}$ to represent S with b decreased by one and e increased by one. We then do a cyclic shift and repeat on $S^{-}SSS^{+}SS$ to get $S^{-}S^{-}SS^{+}S^{+}S$. We shift and repeat four more times, giving $S^{\pm}S^{\pm}S^{\pm}S^{\pm}S^{\pm}S^{\pm} = S^{\pm} = S'$ in Lemma 4.1 (a). At each step, we reduced EWT_S, so $EWT_S(S', p_1) \leq EWT_S(S, p_1)\,\forall p_1$, and Lemma 4.2 is proved.

4.5 The Irreducible Schedules

We begin the proof of Theorem 4.1 by classifying schedules into one of two sets. The first set is the "reducible" schedules, the set of all schedules for which the lemmas can be applied to give a strictly better schedule. The second set is the "irreducible" schedules, the set of all schedules for which no lemmas can be used to give a strictly better schedule. We see that any schedule will be in exactly one of these two sets. We then look at the set of irreducible schedules, since any reducible schedule is worse than some irreducible schedule and hence not optimal. We compare these irreducible schedules and find that a small subset of them (the set of schedules listed in Theorem 4.1) forms the set of optimal schedules.

Each of the lemmas provides a way to change a schedule to get another equally good or better schedule. We can think of the lemmas as describing actions we can perform on schedules to change them. There are two types of actions. For each type, we assume some fixed value of p_1.

The first type reduces the expected waiting time of a schedule. This type of action gives a schedule that is strictly better than the original. These actions establish a partial ordering, ρ_2, among schedules. When one schedule can be modified by one of these actions to get a second schedule, the second schedule is less than the first according to ρ_2.

The second type of action changes the structure of the schedule, but keeps the expected waiting time the same. This type of action does not give a measurably better schedule, but instead identifies schedules that are equal with respect to ρ_2. When one schedule can be modified by one of these actions to get a second schedule, the two schedules are equal.

There are two orderings of schedules. The first, ρ_1, is by EWT_S. Any two schedules can be compared using EWT_S. This is the ordering we use to define the optimal schedule at any value of p_1. The optimal schedule is simply the one that is "less than or equal to" all other schedules according to ρ_1.

The second ordering, ρ_2, is by the actions described above. Not all schedules can be compared by ρ_2, just the ones that are the initial and resulting schedules from some action. Any two that can be compared with ρ_2 will have the same ordering as by ρ_1, so ρ_2 is really a sub-ordering of ρ_1. That is, the set of relationships described by ρ_2 is a subset of the set of relationships described by ρ_1.

So, any minimal schedule under ρ_1 will be a minimal schedule under ρ_2. Our strategy for determining the optimal schedule (the minimal element under ρ_1) will be to determine the minimal elements under ρ_2 and then compare them under ρ_1 to find the optimal schedule.

For ease of referencing the lemmas and their associated transformations, we list the following actions that we can perform on schedules:

A1: $S = \mathbf{AB} \rightarrow S_1' = \mathbf{A}, S_2' = \mathbf{B}$, where both A and B start with the same sub-schedule C, which contains at least two 1's and two 2's.

A2: $S = \ldots 221\mathbf{21}\overbrace{(2)1(2)1\ldots(2)1}^{n\ 1's}22\ldots \rightarrow S' = \ldots 221\mathbf{12}\overbrace{(2)1(2)1\ldots(2)1}^{n\ 1's}22\ldots$, $n \geq 2$

A3: $S \rightarrow S' = S^R$

A4: $S = [\ldots, a, \mathbf{n}, \mathbf{m}, b, \ldots] \rightarrow S' = [\ldots, a, \mathbf{2}, \mathbf{m{+}n{-}2}, b, \ldots]$, $n \geq 2$, $m \geq 2$, $b \geq a$

A5: $S = [\ldots, a, \mathbf{b}, c, \ldots, d, \mathbf{e}, f, \ldots] \rightarrow S' = [\ldots, a, \mathbf{b{-}1}, c, \ldots, d, \mathbf{e{+}1}, f, \ldots]$, $b \geq 3$, $e \geq 2$

A6: $S = \ldots 12^n22\mathbf{21}111^m2\ldots \rightarrow S' = \ldots 12^n22\mathbf{12}111^m2\ldots, n \geq 0, m \geq 0$

We use the notation "N" to mean "three or more." Different occurrences of N within a schedule can correspond to different numbers greater than or equal to three. For example, [0,N,1,2,N] can represent the schedule [0,3,1,2,3], [0,3,1,2,4], or [0,10,1,2,12], but not [0,3,1,2,1], [0,0,1,2,2], or [0,10,1,2,2].

We first show a weaker version of Theorem 4.1:

Proposition 4.1 *All optimal schedules are equal to a schedule or the complement of a schedule in the following list:*

[0,1], [0,1,1], [0,1,2], [0,1,2,N], [0,1,N], [0,1,N,2], [0,1,N,2,N], [0,2], [0,2,1,N], [0,2,N], [0,2,N,1,N], [0,N], [0,N,1,2,N], [0,N,1,N], [0,N,1,N,2,N] [0,N,2,N], [1], [1,2], [1,2,N], [1,N], [1,N,2,N], [2], [2,N].

Note that the optimal schedules ([0,2], [0,N], [2], and [2,N]) are all included in the list. We show later that these are the best schedules in the list. We now derive this list containing all the irreducible schedules.

First we consider all schedules that don't contain both three consecutive 1's and three consecutive 2's. For now, we assume there are no 111's. We will eliminate certain schedules based on the fact that we can find a better schedule according to ρ_2. Since we can find a better schedule, the original schedule is reducible, and we can exclude it from the list.

We know that no irreducible schedule can contain more than one each of 1122, 2211, 1212, 2121, 1221, or 2112, since we could use action A1 on such a schedule to get a better schedule. Here, and in general, we ignore schedules that are optimal at exactly one point, such as the schedule before splitting, at the value of p_1 where the two sub-schedules have the same waiting time as the original. This is because the other two schedules are not only optimal at that point, but also at neighboring points.

We can not have a schedule with two 0's in it, since we would have [...,0,...,0,...], which gives us two 2112's, or [...,0,0,...], which gives us a 111. We can not have a schedule with two 2's in it, since we would have [...,2,...,2,...], which gives us two

1221's. If we have a schedule with two 1's in it, we have two instances of 121. If both are preceded by or followed by something other than 0, we get two instances of 1212 or 2121. The only way to prevent this is for one to be preceded by 0 and the other to be followed by 0. Since we can have at most one 0, we need to have [...,1,0,1,...], with only 2's or larger in the rest of the schedule. But with 2's in the schedule, we can use action A2 to get a better schedule. So, the only irreducible schedule with two 1's is [0,1,1].

We can not have a schedule with two adjacent N's, where N represents some number greater than 2, since we can use action A4 on such a schedule to get a better schedule. So, the irreducible schedules are [0,1,1] and all schedules that have at most one 0, 1, and 2, and no adjacent N's. It is straightforward to list these:

[0,1,1], [N], [0], [0,N], [1], [1,N], [2], [2,N], [0,1], [0,1,N], [0,N,1], [0,N,1,N], [0,2], [0,2,N], [0,N,2], [0,N,2,N], [1,2], [1,2,N], [1,N,2], [1,N,2,N], [0,1,2], [0,1,2,N], [0,1,N,2], [0,N,1,2], [0,1,N,2,N], [0,N,1,2,N], [0,N,1,N,2], [0,2,1], [0,2,1,N], [0,2,N,1], [0,N,2,1], [0,2,N,1,N], [0,N,2,1,N], [0,N,2,N,1], [1,0,2], [1,0,2,N], [1,0,N,2], [1,N,0,2], [1,0,N,2,N], [1,N,0,2,N], [1,N,0,N,2], [1,2,0], [1,2,0,N], [1,2,N,0], [1,N,2,0], [1,2,N,0,N], [1,N,2,0,N], [1,N,2,N,0], [2,0,1], [2,0,1,N], [2,0,N,1], [2,N,0,1], [2,0,N,1,N], [2,N,0,1,N], [2,N,0,N,1], [2,1,0], [2,1,0,N], [2,1,N,0], [2,N,1,0], [2,1,N,0,N], [2,N,1,0,N], [2,N,1,N,0], [0,N,1,N,2,N], [0,N,2,N,1,N].

We can eliminate repetitions of the same schedule using periodicity of the schedules. For example, [0,2,N,1] = [1,0,2,N]. We can use action A3 to eliminate reversals. We also eliminate [0], since it does not contain item 2. When we do this, we get the following list of schedules:

[0,1], [0,1,1], [0,1,2], [0,1,2,N], [0,1,N], [0,1,N,2], [0,1,N,2,N], [0,2], [0,2,1,N], [0,2,N], [0,2,N,1,N], [0,N], [0,N,1,2,N], [0,N,1,N], [0,N,1,N,2,N] [0,N,2,N], [1], [1,2], [1,2,N], [1,N], [1,N,2,N], [2], [2,N].

If we allow 111 and not 222, we get the complements of the schedules in this list. Thus, the set of these schedules and their complements contains all irreducible schedules that do not have both a sequence of 111 and a sequence of 222.

We now consider schedules with both 111 and 222 sequences. We can decompose any such schedule into a sequence of sub-schedules that begin and end with either 111 or 222, and have no sequences of three or more 1's or 2's other than at their beginning or end. We will now derive the set of irreducible sub-schedules that can be combined to form irreducible schedules.

To do this, we start with the sequence 1112 and extend this schedule one piece at a time until we reach either a 111...111 sub-schedule, a 111...222 sub-schedule, or a reducible sub-schedule (a sub-schedule such that any schedule containing it is reducible). For each position, we can choose either 1 or 2, so we search all possibilities using a binary tree. This is diagrammed in Figure 4.1.

From this tree, we see that there are seven irreducible 111...222 sub-schedules. We label them A through G. The irreducible 222...111 sub-schedules are simply the complements of the irreducible 111...222 sub-schedules. We will call these $\overline{\mathrm{A}}$ through $\overline{\mathrm{G}}$. We now generate 111...222... schedules by combining these 111...222 and 222...111 sub-schedules. We use action A1 to eliminate reducible schedules and we find that the only irreducible combinations are $\mathrm{C}\overline{\mathrm{C}}$, $\mathrm{C}\overline{\mathrm{E}}$, $\mathrm{E}\overline{\mathrm{C}}$, $\mathrm{F}\overline{\mathrm{G}}$, $\mathrm{G}\overline{\mathrm{F}}$, and $\mathrm{G}\overline{\mathrm{G}}$.

Of these, only $\mathrm{C}\overline{\mathrm{C}}$ and $\mathrm{G}\overline{\mathrm{G}}$ do not contain all six possible patterns of length four that have two each of item 1 and item 2. In these we can replace 111 with 111...111, if the 111...111 sub-schedule does not have any of the same patterns in it as the starting schedule. We find that the only such 111...111 sub-schedule is 111212111. When this sub-schedule is added to $\mathrm{C}\overline{\mathrm{C}}$ and $\mathrm{G}\overline{\mathrm{G}}$, each resulting schedule contains all six patterns. So, the only possible schedules with both 111 and 222 are: $\mathrm{C}\overline{\mathrm{C}}$, $\mathrm{C}\overline{\mathrm{E}}$, $\mathrm{E}\overline{\mathrm{C}}$, $\mathrm{F}\overline{\mathrm{G}}$, $\mathrm{G}\overline{\mathrm{F}}$, $\mathrm{G}\overline{\mathrm{G}}$, $\mathrm{C'}\overline{\mathrm{C}}$, $\mathrm{G'}\overline{\mathrm{G}}$, $\mathrm{C}\overline{\mathrm{C'}}$, and $\mathrm{G}\overline{\mathrm{G'}}$, where C' is C with 111 replaced by 111212111 and G' is G with 111 replaced by 111212111.

We need only consider $\mathrm{C}\overline{\mathrm{C}}$, $\mathrm{C}\overline{\mathrm{E}}$, and $\mathrm{C'}\overline{\mathrm{C}}$, since $\mathrm{E}\overline{\mathrm{C}} = (\mathrm{C}\overline{\mathrm{E}})^C$, $\mathrm{F}\overline{\mathrm{G}} = (\mathrm{C}\overline{\mathrm{E}})^R$, $\mathrm{G}\overline{\mathrm{F}} = (\mathrm{C}\overline{\mathrm{E}})^{CR}$, $\mathrm{G}\overline{\mathrm{G}} = (\mathrm{C}\overline{\mathrm{C}})^{CR}$, $\mathrm{G'}\overline{\mathrm{G}} = (\mathrm{C'}\overline{\mathrm{C}})^R$, $\mathrm{C}\overline{\mathrm{C'}} = (\mathrm{C'}\overline{\mathrm{C}})^C$, and $\mathrm{G}\overline{\mathrm{G'}} = (\mathrm{C'}\overline{\mathrm{C}})^{CR}$. We use action A4 on $\mathrm{C}\overline{\mathrm{C}} = 111211222122$ to get 111211221222. We then use action A6 to get the schedule 211211221221, to which we can apply action A2. We can do the same thing to $\mathrm{C}\overline{\mathrm{E}} = 11121122212122$ and $\mathrm{C'}\overline{\mathrm{C}} = 111212111211222122$, applying action A4 to give 11211122212122 and 111212112111222122, respectively, and then A6 to get 11211212212122 and 111212112112122122, which we can reduce using action A2.

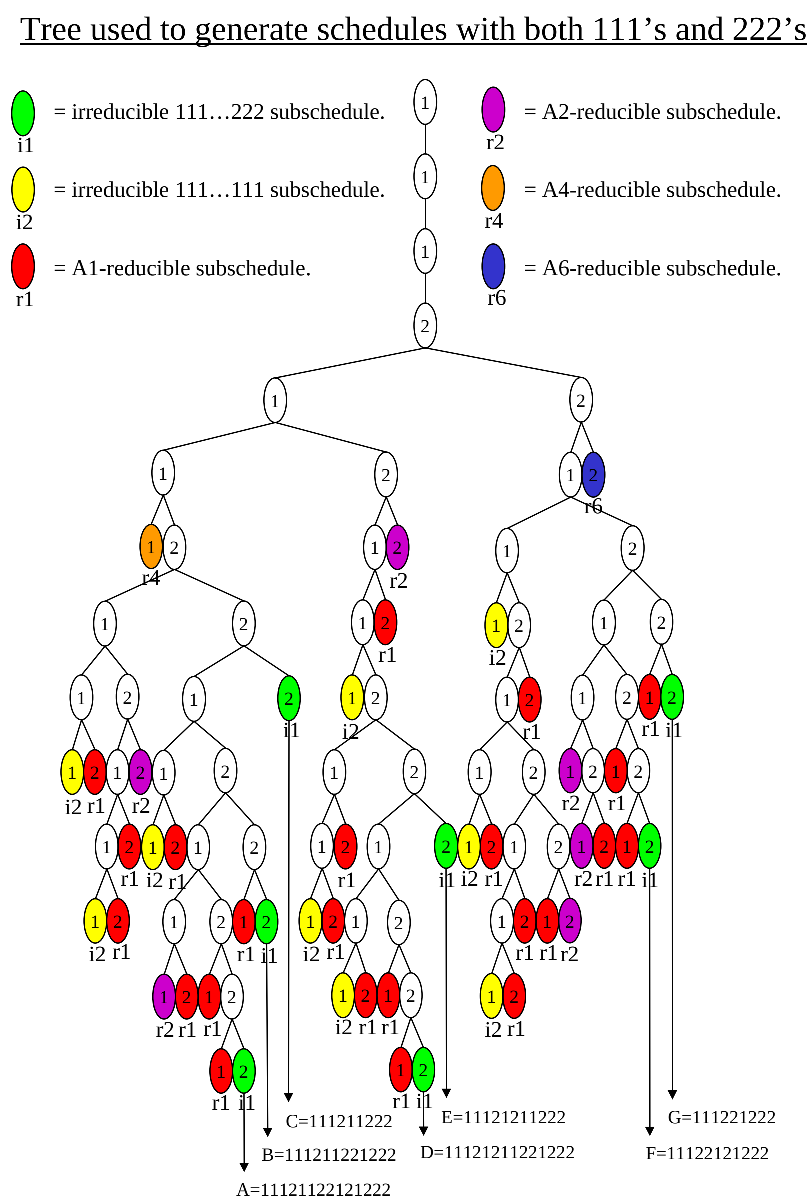

Figure 4.1: Tree of schedules.

Since all of the schedules listed above can be reduced, there are no irreducible schedules with both a 111 and a 222. So, the schedules listed previously and their complements are the only possible schedules that can not be reduced to better schedules, and Proposition 4.1 follows.

4.6 The Optimal Schedules

We have thus far shown that the schedules 1122, 11222, 112222, and 1221222^n, $n \geq 0$, are irreducible. Now we finish the proof by showing they are better than the other irreducible schedules:

Proposition 4.2 *At any value of* p_1*,* $EWT_S(S, p_1)$ *is minimized over all irreducible* S *by one of* 1122*,* 11222*,* 112222*, or* 1221222^n*,* $n \geq 0$*.*

To show this, we first compute their expected waiting times as a function of p_1. We then plot expected waiting time versus p_1 for each schedule. We find the appropriate intersection points and see that 1122 is best on $p_1 \in \left(\frac{5}{16}, \frac{1}{2}\right)$, 11222 is best on $\left(\frac{5}{21}, \frac{5}{16}\right)$, 112222 is best on $\left(\frac{1}{5}, \frac{5}{21}\right)$, 122122 is best on $\left(\frac{2}{17}, \frac{1}{5}\right)$, and $1221222^n, n \geq 1$, is best on $\left(\frac{8}{(n+1)^2+13(n+1)+68}, \frac{8}{n^2+13n+68}\right)$. We compute these intersection points by first computing EWT_S for the schedules as a function of p_1, and n when appropriate. We then set "neighboring" schedules' EWT_S's equal to each other and solve for p_1. We use this value of p_1 to compute the EWT_S and get our (p_1, EWT_S) pairs.

This is illustrated in Figure 4.2. The curve in the figure is the best we can do without splitting, as given by Theorem 3.1.

The splitting schedules 1122 and 112222 are similar to the no splitting schedules 12 and 122, since we send the same information in the same order in each. However, the splitting schedules allow a lower EWT_S because we can start getting an item halfway through, unlike the no splitting schedules where we must wait until it is sent again. As p_1 decreases, we also take advantage of the fact that we can separate the two halves of item 1 within the schedule, to get even greater gains in performance.

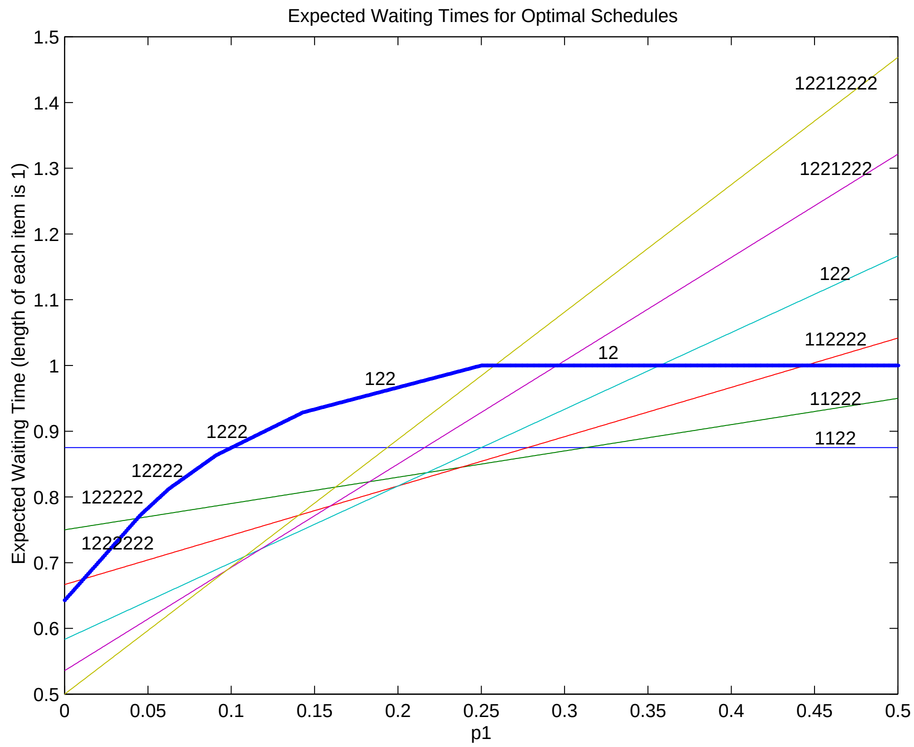

Figure 4.2: Expected waiting time versus p_1 for some of the optimal schedules. The lines are for splitting, the solid piecewise linear curve is for no splitting.

It remains to show that any other irreducible schedule is worse than one of the optimal schedules listed above for any value of p_1. Suppose there is some schedule, with graph defined by $t = a \cdot p_1 + b$, that has a shorter expected waiting time at some p_1 than any of our optimal schedules. Then it will dip below the (piecewise linear) minimum function of the graphs. Since this function is linear, it must be less at some corner of the minimum function. So, if we can show that no schedule has a graph that dips below a corner, we have shown that our schedules are optimal.

We do this by checking the set $C_1 = \left\{\left(\frac{5}{16}, \frac{7}{4}\right), \left(\frac{5}{21}, \frac{71}{42}\right), \left(\frac{1}{5}, \frac{49}{30}\right)\right\}$ of intersection points of 1122 and 11222, 11222 and 112222, and 112222 and 122, and the set $C_2 = \left\{ \left(\frac{8}{n^2+13n+68}, \frac{1}{2} + \frac{8\left(n^2+14n+48\right)}{n^3+19n^2+146n+408} \right) \middle| \, n \geq 0 \right\}$, where 1221222^n and 1221222^{n+1} intersect, $\forall n \geq 0$. These are our corner points. We evaluate $t = a \cdot p_1 + b$ at the corner point (p_c, t_c) and compute the difference $t - t_c = a \cdot p_c + b - t_c$. We then show this is positive at all corner points.

As an example, consider the schedule [0,N]. If $m = N-2$, this has expected waiting time $\frac{m^2+7m+10}{4(m+4)}$ for item 1 and $\frac{5}{4(m+4)}$ for item 2. So, $t - t_c$ at the "n^{th}" corner point in C_2 is

$$\frac{m^2+7m}{4m+16} \cdot \frac{8}{n^2+13n+68} + \frac{5}{2m+8} - \frac{4(n^2+14n+48)}{n^3+19n^2+146n+408}$$

This is non-negative $\Leftrightarrow$

$$4(m^2+7m)(n+6) + 5(n^3+19n^2+146n+408) - 4(2m+8)(n^2+14n+48) \geq 0$$

We rewrite as a quadratic in m to get

$$(4n+24)m^2 - (8n^2+84n+216)m + (5n^3+63n^2+282n+504) \geq 0$$

We know this is always positive if it has no real roots, so it is always positive if its discriminant is negative. So, we want

$$(8n^2+84n+216)^2 - 4(4n+24)(5n^3+63n^2+282n+504) < 0$$

Simplifying, we get

$$16n^4 + 144n^3 + 48n^2 - 1152n + 1728 > 0$$

This is true for $n = 0, 1, 2$. For $n \geq 3$, note that this expression is greater than $16(n-3)^4$, which is non-negative for all $n \geq 3$. So, this expression is always positive, and hence there are no values of $m \geq 0$ and $n \geq 0$ where the schedule 11222^m is better than 1221222^n. We also check against 1122, 11222, 112222, and 122122, and see that we do not dip below their corner points. We do this the same way, except now we only have the single parameter n instead of both m and n.

We use the same basic idea for all other irreducible schedules. Some of them have more than one group of N 2's. For these, we use action A5 to determine

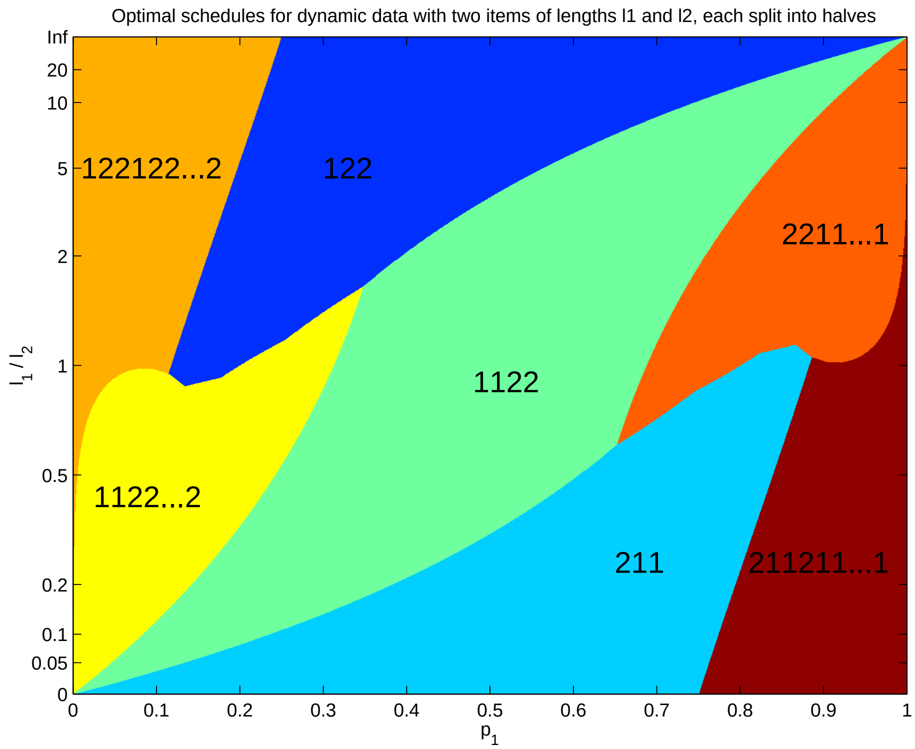

Figure 4.3: Optimal scheduling with different lengths.

how many 2's the blocks can have relative to each other, and we consider all possible combinations, doing the above calculation for each. So, Proposition 4.2 is proved, and combining with Proposition 4.1 we see that our small set of schedules $\{1122, 11222, 112222, 1221222^n, n \geq 0\}$ performs better than any other schedule. The intervals on which each is optimal are described by C_1 and C_2. These agree with the intervals in the statement of Theorem 4.1, thus completing the proof.

4.7 Different Length Items

We have shown the optimal scheduling for two items of the same length, when we split them each into halves. Now we consider the same situation, but with items of different lengths. We fix the length of item 1 at $l_1 = 1$ and let item 2 have length l_2 that can be any positive value. We then split item 1 into two pieces, each of size $\frac{1}{2}$, and split item 2 into two pieces, each of size $\frac{l_2}{2}$. We want to find the optimal schedule for the items, as a function of the ratio of the lengths, l_2, and the demand probabilities p_1 and $p_2 = 1 - p_1$.

We attempt to use the same reasoning as with $l_2 = l_1$. Since the lemmas we used for equal lengths also hold true for different lengths, we can use the same reductions and arrive at the same set of irreducible schedules. However, comparing the schedules to each other is more difficult than when l_2 is fixed at 1.

To get an idea about which schedules are optimal for which values of l_2 and p_1, we numerically checked a range of values of l_2 and p_1, and found the optimal schedule at each pair of values. The results are shown in Figure 4.3. We see that the only optimal schedules are 11222^n, 1221222^n, $n \geq 0$, and their complements. It is interesting to note that these are the same optimal schedules as for equal length items.

It would be interesting to know if this is provably true for all p_1 and l_2. We have been able to eliminate some of the schedules as being non-optimal. For example, [1,n+2] is always worse than [0,n+3], for all l_2, p_1, and $n \geq 0$, so [1,N] is worse than [0,N] and hence [1,N] is never an optimal schedule. However, we have not been able to eliminate all other schedules, as we could for $l_2 = l_1$.

Chapter 5 Time-Division Scheduling with Multiple Splits

In this chapter, we extend the idea of splitting. Instead of splitting items in half, we look at splitting them in an arbitrary way. We essentially assume that a data item can be split into n pieces of equal size, and then assume $n \to \infty$.

Section 5.1 presents these new ideas and some new notation and a proposition that we would like to prove. Then, in Section 5.2, we present some lemmas that prove parts of the proposition. Sections 5.3 through 5.6 present proofs of these lemmas.

5.1 Introduction

In this section, we think of each item as consisting of very many very small pieces, so a 1 or 2 will represent some number of these pieces, and not the entire item. We will use exponents to indicate how many pieces. The exponent is the fraction of the complete item that is broadcast. For example, $1^{\frac{1}{2}}2^1$ means broadcast half of item 1 and then all of item 2.

Broadcast schedules are cyclic, so we will represent them by one of their cycles. Since we have pieces of each item, we assume these pieces are broadcast in sequence, where the next piece to broadcast depends only on the previous piece of that item that was broadcast.

We define a new notion of expected waiting time to deal with this arbitrary splitting of items.

Definition 5.1 $EWT_{AS}(S, p_1)$ *is the expected waiting time using schedule* S *with demand probabilities* p_1 *and* $p_2 = 1 - p_1$*, assuming two items, each of length one unit, split into arbitrarily small pieces.*

The AS indicates "Arbitrary Splitting" of items. The lower the value of $EWT_{AS}(S, p_1)$, the better schedule S is for demand probabilities p_1 and $p_2 = 1 - p_1$. This is simply a generalization of the previous section. We assume we have dynamic data that is split as before, so we can receive pieces start to finish and out of order. However, as the number of pieces becomes large, this limiting case is simply static data. So, to avoid confusion, we will simply assume the data is static.

In general, we would like to show the following:

Proposition 5.1 *For two static items of the same length, each split into arbitrarily small pieces, the broadcast schedule that minimizes expected waiting time is:*

$1^1 2^1$, *if* $p_1 \in \left(\frac{3}{8}, \frac{1}{2}\right]$

$1^1 2^{l-1}, l = \sqrt{\frac{3}{p_1} - 4}$, *if* $p_1 \in \left(\frac{15-4\sqrt{5}}{145}, \frac{3}{8}\right]$

$1^{\frac{1}{n}} 2^1 ... 1^{\frac{1}{n}} 2^{l-n}, l = \frac{1}{n}\sqrt{\frac{n(2n+1)}{p_1} - n(n+1)(n^2+1)}$, *if*

$p_1 \in \left(\frac{1-n+3n^2+8n^3+4n^4-2n(n+1)\sqrt{-1-6n+8n^3+4n^4}}{1-2n+11n^2+42n^3+53n^4+32n^5+8n^6}, \frac{1+n+3n^2-8n^3+4n^4-2n(n-1)\sqrt{1+2n-8n^3+4n^4}}{1+2n+3n^2-18n^3+17n^4-16n^5+8n^6}\right]$

Following are results related to this proposition. We present and prove some lemmas related to Proposition 5.1.

5.2 The Lemmas

These lemmas tell us how to optimize certain classes of schedules described in the proposition. For $p_1 > \frac{1}{2}$, we swap which items we call "1" and "2."

Lemma 5.1 *(Optimization for 1 piece)*

If a schedule S_1 of the form $1^1 2^{l-1}$ minimizes $EWT_{AS}(S, p_1)$, where $l \geq 2$ and $0 \leq p_1 \leq \frac{3}{8}$, then $EWT_{AS}(S, p_1)$ is minimized by $S = S_1$ with $l = \sqrt{\frac{3}{p_1} - 4}$.

Lemma 5.2 *(Optimization for 2 pieces)*

If a schedule S_2 of the form $1^\alpha 2^\beta 1^{1-\alpha} 2^{l-1-\beta}$ minimizes $EWT_{AS}(S, p_1)$, where $\alpha > 0$, $\beta > 0$, $l \geq 2 + \beta$, and $0 \leq p_1 \leq \frac{5}{33}$, then $EWT_{AS}(S, p_1)$ is minimized by $S = S_2$ with $\alpha = \frac{1}{2}, \beta = 1, l = \frac{1}{2}\sqrt{\frac{10}{p_1} - 30}$.

Lemma 5.3 *(Optimization for 3 pieces)*

If a schedule S_3 of the form $1^{\alpha}2^{\beta}1^{\gamma}2^{\delta}1^{1-\alpha-\gamma}2^{l-1-\beta-\delta}$ minimizes $EWT_{AS}(S,p_1)$, where $\alpha > 0, \beta > 0, \gamma > 0, \delta > 0, l \geq 2+\beta+\delta$, and $0 \leq p_1 \leq \frac{7}{88}$, then $EWT_{AS}(S,p_1)$ is minimized by $S = S_3$ with $\alpha = \gamma = \frac{1}{3}, \beta = \delta = 1, l = \frac{1}{3}\sqrt{\frac{21}{p_1} - 120}$.

Lemma 5.4 *(Restricted optimization for N pieces)*

If a schedule S_N of the form $1^{\frac{1}{N}}2^1 1^{\frac{1}{N}}2^1 \cdots 1^{\frac{1}{N}}2^{l-N}$ minimizes $EWT_{AS}(S,p_1)$, where $l \geq N+1$, and $0 \leq p_1 \leq \frac{2N+1}{(N+1)(2N^2+N+1)}$, then $EWT_{AS}(S,p_1)$ is minimized by $S = S_N$ with $l = \frac{1}{N}\sqrt{\frac{N(2N+1)}{p_1} - N(N+1)(N^2+1)}$.

5.3 Proof of Lemma 5.1

$S = S_1$ has the form $1^1 2^{l-1}$, $l \geq 2$, and $0 \leq p_1 \leq \frac{3}{8}$. We want to choose l to minimize $EWT_{AS}(S,p_1)$.

$$
\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}(l-1) + \frac{l-1}{l}\left(\frac{l-1}{2}\right) \\
&= \frac{1}{2l}\left(2l - 2 + l^2 - 2l + 1\right) \\
&= \frac{l^2-1}{2l} \\
EWT_{AS,2}(S) &= \frac{1}{l}\left(\frac{1}{2}\right) + \frac{l-2}{l}(0) + \frac{1}{l}(1) \\
&= \frac{1}{2l}(1+2) \\
&= \frac{3}{2l} \\
EWT_{AS}(S,p_1) &= t_1 p_1 + t_2(1-p_1) \\
&= \frac{l^2-4}{2l}p_1 + \frac{3}{2l} \\
\frac{dEWT_{AS}(S,p_1)}{dl} &= \frac{1}{2}p_1 + 2l^{-2}p_1 - \frac{3}{2}l^{-2} \\
&= \frac{(l^2+4)\,p_1 - 3}{2l^2}
\end{aligned}
$$

$$\begin{aligned}
\frac{dEWT_{AS}(S,p_1)}{dl} &= 0 \Longrightarrow \\
\left(l^2+4\right)p_1 &= 3 \Longrightarrow \\
l &= \sqrt{\frac{3}{p_1}-4}
\end{aligned}$$

This is valid for $l \geq 2 \Longleftrightarrow p_1 \leq \frac{3}{8}$.

$$\begin{aligned}
\frac{d^2EWT_{AS}(S,p_1)}{dl^2} &= -4l^{-3}p_1 + 3l^{-3} \\
&= -4p_1\left(\frac{3}{p_1}-4\right)^{-2}\sqrt{\frac{3}{p_1}-4} + 3\left(\frac{3}{p_1}-4\right)^{-2}\sqrt{\frac{3}{p_1}-4} \\
&= (3-4p_1)\left(\frac{3}{p_1}-4\right)^{-2}\sqrt{\frac{3}{p_1}-4} > 0 \ \forall p_1 \leq \frac{3}{8}
\end{aligned}$$

So, we have a minimum value of EWT_{AS} at $l = \sqrt{\frac{3}{p_1}-4}$, and it follows that $EWT_{AS}(S,p_1)$ is minimized for $S = 1^1 2^{l-1}$ when $l = \sqrt{\frac{3}{p_1}-4}$.

When we choose l this way, the expected waiting time is

$$\begin{aligned}
EWT_{AS}(S,p_1) &= \frac{l^2-4}{2l}p_1 + \frac{3}{2l} \\
&= \frac{\left(\frac{3}{p_1}-4\right)-4}{2\sqrt{\frac{3}{p_1}-4}}p_1 + \frac{3}{2\sqrt{\frac{3}{p_1}-4}} \\
&= \sqrt{3p_1-4p_1^2}, \text{ for } 0 \leq p_1 \leq \frac{3}{8}
\end{aligned}$$

5.4 Proof of Lemma 5.2

$S = S_2$ has the form $1^\alpha 2^\beta 1^{1-\alpha} 2^{l-1-\beta}, \alpha > 0, \beta > 0, l \geq 2+\beta, 0 \leq p_1 \leq \frac{5}{33}$. We want to find the values of α, β, and l that minimize $EWT_{AS}(S,p_1)$.

5.4.1 Case 1: $\beta \geq 1$

$$\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}\left[\alpha\,(l-1) + \beta\left(l-1-\frac{\beta}{2}\right) + (1-\alpha)\,(l-1)\right. \\
&\quad \left. + \,(l-1-\beta)\left(\frac{l}{2}-\frac{1}{2}+\frac{\beta}{2}\right)\right] \\
&= \frac{1}{l}\left[\frac{l^2}{2}+\beta l-\beta^2-\beta-\frac{1}{2}\right] \\
\frac{dEWT_{AS,1}(S)}{d\beta} &= \frac{1}{l}\,[l-\beta-1] = 0 \Longrightarrow \\
l &= 2\beta+1 \Longrightarrow \\
\beta &= \frac{l-1}{2}
\end{aligned}$$

This is a relative maximum, so we choose β as large or small as possible to minimize $EWT_{AS,1}(S)$. The limits on β are 1 and $l-2$. These two values give the same schedule, so choose $\beta = 1$.

$$\begin{aligned}
EWT_{AS,2}(S) &= \frac{1}{l}\left[\alpha\left(\frac{\alpha}{2}\right) + (\beta-1)\,(0) + 1\,(1-\alpha) + (1-\alpha)\left(\frac{1-\alpha}{2}\right)\right. \\
&\quad \left. + \,(l-2-\beta)\,(0) + 1\,(\alpha)\right] \\
&= \frac{1}{l}\left[\alpha^2-\alpha+\frac{3}{2}\right] \\
\frac{dEWT_{AS,2}(S)}{d\alpha} &= \frac{1}{l}\,[2\alpha-1] = 0 \Longrightarrow \\
\alpha &= \frac{1}{2}
\end{aligned}$$

This is a relative minimum, so we choose $\alpha = \frac{1}{2}$. The resulting schedule is the following:

$$1^{\frac{1}{2}}2^{1}1^{\frac{1}{2}}2^{l-2}$$

5.4.2 Case 2: $\beta < 1$

$$\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}\left[\alpha(l-1)+\beta\left(l-1-\frac{\beta}{2}\right)+(1-\alpha)(l-1)\right.\\
&\quad \left. +(l-1-\beta)\left(\frac{l}{2}-\frac{1}{2}+\frac{\beta}{2}\right)\right]\\
&= \frac{1}{l}\left[\frac{l^2}{2}+\beta l-\beta^2-\beta-\frac{1}{2}\right]\\
\frac{dEWT_{AS,1}(S)}{d\beta} &= \frac{1}{l}\left[l-2\beta-1\right]=0 \Longrightarrow\\
l &= 2\beta+1 \Longrightarrow\\
\beta &= \frac{l-1}{2}\\
t_2 &= \frac{1}{l}\left[\alpha\left(1-\frac{\alpha}{2}\right)+\beta(1-\alpha)+(1-\alpha)\left(\frac{1-\alpha}{2}\right)+(l-2-\beta)(0)\right.\\
&\quad \left. +\beta(\alpha)+(1-\beta)(1)\right]\\
&= \frac{1}{l}\left[\frac{3}{2}\right]
\end{aligned}$$

We see that for $\beta \geq 1$ and $\beta < 1$ we get the same expression for $EWT_{AS,1}(S)$. For $EWT_{AS,2}(S)$, we get $\frac{1}{l}\left[\alpha^2-\alpha+\frac{3}{2}\right]$ when $\beta \geq 1$ and $\frac{1}{l}\left[\frac{3}{2}\right]$ when $\beta < 1$. We know $\alpha^2-\alpha < 0\ \forall\alpha, 0<\alpha<1$, so it follows that $EWT_{AS,2}(S)$ is always less when $\beta \geq 1$ than when $\beta < 1$. So the case of $\beta < 1$ is never optimal.

So the best schedule is when $\alpha = \frac{1}{2}$ and $\beta = 1$.

Now we choose the length, l, of the schedule.

$$\begin{aligned}
EWT_{AS}(S,p_1) &= \frac{1}{l}\left[\frac{l^2}{2}+l-\frac{5}{2}\right]p_1+\frac{1}{l}\left[\frac{5}{4}\right](1-p_1)\\
&= \frac{1}{l}\left[\left(\frac{l^2}{2}+l-\frac{15}{4}\right)p_1+\frac{5}{4}\right]\\
\frac{dEWT_{AS}(S,p_1)}{dl} &= \frac{1}{2}p_1+\frac{15}{4l^2}p_1-\frac{5}{4l^2}=0 \Longrightarrow\\
\frac{1}{l^2}\left(\frac{5}{4}-\frac{15}{4}p_1\right) &= \frac{1}{2}p_1 \Longrightarrow\\
l &= \frac{1}{2}\sqrt{\frac{10}{p_1}-30}
\end{aligned}$$

This is valid for $l \geq 3 \iff p_1 \leq \frac{5}{33}$.

$$\begin{aligned}
\frac{d^2 EWT_{AS}(S,p_1)}{dl^2} &= \frac{-15}{2} l^{-3} p_1 + \frac{5}{2} l^{-3} \\
&= \frac{-15}{2} p_1 \left[\frac{1}{4} \left(\frac{10}{p_1} - 30 \right)^{-2} \frac{1}{2} \sqrt{\frac{10}{p_1} - 30} \right] \\
&\quad + \frac{5}{2} \left[\frac{1}{4} \left(\frac{10}{p_1} - 30 \right)^{-2} \frac{1}{2} \sqrt{\frac{10}{p_1} - 30} \right] \\
&= \frac{1}{4} (10 - 30 p_1) \frac{1}{4} \left(\frac{10}{p_1} - 30 \right)^{-2} \frac{1}{2} \sqrt{\frac{10}{p_1} - 30} \geq 0 \ \forall p_1 \leq \frac{5}{33}
\end{aligned}$$

So, we have a minimum value of EWT_{AS} at $l = \frac{1}{2}\sqrt{\frac{10}{p_1} - 30}$, and it follows that $EWT_{AS}(S,p_1)$ is minimized for $S = 1^{\frac{1}{2}} 2^1 1^{\frac{1}{2}} 2^{l-2}$ when $l = \frac{1}{2}\sqrt{\frac{10}{p_1} - 30}$.

When we choose l this way, the expected waiting time is

$$\begin{aligned}
EWT_{AS}(S,p_1) &= \frac{2l^2 + 4l - 15}{4l} p_1 + \frac{5}{4l} \\
&= \frac{2\left(\frac{1}{4}\left(\frac{10}{p_1} - 30\right)\right) + 4\left(\frac{1}{2}\sqrt{\frac{10}{p_1} - 30}\right) - 15}{4\left(\frac{1}{2}\sqrt{\frac{10}{p_1} - 30}\right)} p_1 + \frac{5}{4\left(\frac{1}{2}\sqrt{\frac{10}{p_1} - 30}\right)} \\
&= p_1 + \frac{1}{2}\sqrt{10 p_1 - 30 p_1^2}, \text{ for } 0 \leq p_1 \leq \frac{5}{33}
\end{aligned}$$

5.5 Proof of Lemma 5.3

$S = S_3$ has the form $1^\alpha 2^\beta 1^\gamma 2^\delta 1^{1-\alpha-\gamma} 2^{l-1-\beta-\delta}, \alpha > 0, \beta > 0, \gamma > 0, \delta > 0, l \geq 2 + \beta + \delta,$ and $0 \leq p_1 \leq \frac{7}{88}$. We want to choose $\alpha, \beta, \gamma, \delta,$ and l to minimize $EWT_{AS}(S,p_1)$.

5.5.1 Case 1: $\beta \geq 1, \delta \geq 1$

$$
\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}\left[\alpha(l-1)+\beta\left(l-1-\frac{\beta}{2}\right)+\gamma(l-1)+\delta\left(l-1-\frac{\delta}{2}\right)\right. \\
&\quad +(1-\alpha-\gamma)(l-1) \\
&\quad \left. +(l-1-\beta-\delta)\left(\frac{l}{2}-\frac{1}{2}+\frac{\beta}{2}+\frac{\delta}{2}\right)\right] \\
&= \frac{1}{l}\left[\frac{l^2}{2}+(\beta+\delta)l-\left(\beta^2+\beta\delta+\delta^2+\beta+\delta+\frac{1}{2}\right)\right] \\
\frac{dEWT_{AS,1}(S)}{d\beta} &= \frac{1}{l}\left[l-(2\beta+\delta+1)\right]=0 \Longrightarrow \\
l &= 2\beta+\delta+1 \Longrightarrow \\
\beta &= \frac{l-1-\delta}{2} \geq \frac{1+\beta}{2} \Longrightarrow \\
\beta &\geq 1
\end{aligned}
$$

This value of β gives us a maximum, so choose $\beta = 1$ to minimize $EWT_{AS,1}(S)$. Using similar reasoning for δ, we see that we should choose $\delta = 1$ to minimize $EWT_{AS,1}(S)$.

$$
\begin{aligned}
EWT_{AS,2}(S) &= \frac{1}{l}\left[\alpha\left(\frac{\alpha}{2}\right)+1(\gamma)+\gamma\left(\frac{\gamma}{2}\right)+1(1-\alpha-\gamma)\right. \\
&\quad \left. +(1-\alpha-\gamma)\left(\frac{1}{2}-\frac{\alpha}{2}-\frac{\gamma}{2}\right)+1(\alpha)\right] \\
&= \frac{1}{l}\left[\alpha^2+\gamma^2+\frac{3}{2}-\alpha-\gamma+\alpha\gamma\right] \\
\frac{dEWT_{AS,2}(S)}{d\alpha} &= \frac{1}{l}\left[2\alpha-1+\gamma\right]=0 \Longrightarrow \\
\alpha &= \frac{1-\gamma}{2} \\
\frac{dEWT_{AS,2}(S)}{d\gamma} &= \frac{1}{l}\left[2\gamma-1+\alpha\right]=0 \Longrightarrow \\
\gamma &= \frac{1-\alpha}{2}
\end{aligned}
$$

From $\alpha = \frac{1-\gamma}{2}$ and $\gamma = \frac{1-\alpha}{2}$, we get

$$\begin{aligned} \alpha + \gamma &= 1 - \frac{(\alpha+\gamma)}{2} \Longrightarrow \\ \alpha + \gamma &= \frac{2}{3} \\ EWT_{AS,2}(S) &= \frac{1}{l}\left[(\alpha+\gamma)^2 + \frac{3}{2} - (\alpha+\gamma) - \alpha\gamma\right] \\ &= \frac{23}{18} - \alpha\gamma \end{aligned}$$

This is minimized when $\alpha\gamma$ is maximized, which is at $\alpha = \gamma = \frac{1}{3}$. These values of α and γ give us the following schedule:

$$S = 1^{\frac{1}{3}}2^1 1^{\frac{1}{3}}2^1 1^{\frac{1}{3}}2^{l-3}$$

5.5.2 Case 2a: $\beta \geq 1, \delta < 1$

$$\begin{aligned} EWT_{AS,1}(S) &= \frac{1}{l}\left[\alpha\,(l-1) + \beta\left(l - 1 - \frac{\beta}{2}\right) + \gamma\,(l-1) + \delta\left(l - 1 - \frac{\delta}{2}\right)\right. \\ &\quad \left. + (1-\alpha-\gamma)\,(l-1) + (l-1-\beta-\gamma)\left(\frac{l}{2} - \frac{1}{2} + \frac{\beta}{2} + \frac{\gamma}{2}\right)\right] \\ &= \frac{1}{l}\left[\frac{l^2}{2} + (\beta+\delta)\,l - \left(\beta^2 + \beta\delta + \delta^2 + \beta + \delta + \frac{1}{2}\right)\right] \\ \frac{dEWT_{AS,1}(S)}{d\beta} &= \frac{1}{l}\left[l - 2\beta - \delta - 1\right] = 0 \Longrightarrow \\ \beta &= \frac{l-1-\delta}{2} \geq \frac{1+\beta}{2} \Longrightarrow \\ \beta &\geq 1 \\ \frac{dEWT_{AS,1}(S)}{d\delta} &= \frac{1}{l}\left[l - \beta - 2\delta - 1\right] = 0 \Longrightarrow \\ \delta &= \frac{l-1-\beta}{2} \geq \frac{1+\delta}{2} \Longrightarrow \\ \delta &\geq 1 \end{aligned}$$

These are the values where β and δ maximize t_1, so we choose $\beta = 1$ and $\delta = \epsilon, \epsilon \to 0$ to minimize t_1.

$$\begin{aligned}
EWT_{AS,2}(S) &= \frac{1}{l}\left[\alpha\left(\frac{\alpha}{2}\right) + (\beta-1)(0) + \delta(\gamma) + (1-\delta)(1-\alpha) + \gamma\left(1-\alpha-\frac{\gamma}{2}\right)\right. \\
&\quad +\delta(1-\alpha-\gamma) + (1-\alpha-\gamma)\left(\frac{1}{2}-\frac{\alpha}{2}-\frac{\gamma}{2}\right) + (l-2-\beta-\delta)(0) \\
&\quad \left. +1(\alpha)\right] \\
&= \frac{1}{l}\left[\alpha^2 - \alpha + \frac{3}{2}\right] \\
\frac{dEWT_{AS,2}(S)}{d\alpha} &= \frac{1}{l}[2\alpha-1] = 0 \Longrightarrow \\
\alpha &= \frac{1}{2}
\end{aligned}$$

The schedule we get is the following:

$$1^{\frac{1}{2}}2^{1}1^{\delta}2^{\epsilon}1^{\frac{1}{2}-\delta}2^{l-2-\epsilon}$$

But as $\epsilon \rightarrow 0$, this simply becomes the following schedule:

$$1^{\frac{1}{2}}2^{1}1^{\frac{1}{2}}2^{l-2}$$

This is the schedule from Lemma 5.2. So we see that the case of $\beta \geq 1$ and $\delta < 1$ is never optimal.

5.5.3 Case 2b: $\beta < 1, \delta \geq 1$

This case is symmetric to Case 2a and hence is never optimal.

5.5.4 Case 3: $0 < \beta < 1, 0 < \delta < 1, \beta + \delta \geq 1$

$$
\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}\left[\alpha(l-1) + \beta\left(l-1-\frac{\beta}{2}\right) + \gamma(l-1) + \delta\left(l-1-\frac{\delta}{2}\right)\right. \\
&\quad \left. + (1-\alpha-\gamma)(l-1) + (l-1-\beta-\gamma)\left(\frac{l}{2}-\frac{1}{2}+\frac{\beta}{2}+\frac{\delta}{2}\right)\right] \\
&= \frac{1}{l}\left[\frac{l^2}{2} + \beta l + \delta l - \beta^2 - \delta^2 - \beta\delta - \beta - \delta - \frac{1}{2}\right] \\
\frac{dEWT_{AS,1}(S)}{d\beta} &= \frac{1}{l}\left[l - 2\beta - \delta - 1\right] = 0 \Longrightarrow \\
\beta &= \frac{l-1-\delta}{2} \\
\frac{dEWT_{AS,1}(S)}{d\delta} &= \frac{1}{l}\left[l - 2\delta - \beta - 1\right] = 0 \Longrightarrow \\
\delta &= \frac{l-1-\beta}{2}
\end{aligned}
$$

Combining $\beta = \frac{l-1-\delta}{2}$ and $\delta = \frac{l-1-\beta}{2}$ we get $\beta = \delta = \frac{1}{3}(l-1)$. These values maximize $EWT_{AS,1}(S)$, so choose β and δ as small as possible.

$$
\begin{aligned}
EWT_{AS,2}(S) &= \frac{1}{l}\left[\alpha\left(\frac{\alpha}{2}+\gamma\right) + (\beta+\gamma-1)(\gamma) + (1-\delta)(1-\alpha) + \gamma(1-\alpha-\gamma)\right. \\
&\quad + (1-\alpha-\gamma)\left(\frac{1}{2}-\frac{\alpha}{2}-\frac{\gamma}{2}\right) + (l-2-\beta-\delta)(0) + \beta(\alpha) \\
&\quad \left. + (1-\beta)(\alpha+\gamma)\right] \\
&= \frac{1}{l}\left[\alpha^2 + \alpha\gamma - \alpha + \frac{3}{2}\right] \\
\frac{dEWT_{AS,2}(S)}{d\alpha} &= \frac{1}{l}\left[2\alpha + \gamma - 1\right] = 0 \Longrightarrow \\
\alpha &= \frac{1-\gamma}{2}
\end{aligned}
$$

We minimize $EWT_{AS,2}(S)$ by choosing $\gamma = 0 \Longrightarrow \alpha = \frac{1}{2}$. So we get the optimal schedule $1^{\frac{1}{2}}2^{\beta+\gamma}1^{\frac{1}{2}}2^{l-1-\beta-\gamma}$. But we choose β and γ as small as possible, so their sum will be 1 and we get the schedule $1^{\frac{1}{2}}2^{1}1^{\frac{1}{2}}2^{l-2}$, which is just the schedule in Lemma 5.2. So this case is never optimal.

5.5.5 Case 4: $0 < \beta < 1, 0 < \delta < 1, \beta + \delta < 1$

$$\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}\left[\alpha(l-1) + \beta\left(l-1-\frac{\beta}{2}\right) + \gamma(l-1) + \delta\left(l-1-\frac{\delta}{2}\right)\right. \\
&\quad \left. + (1-\alpha-\gamma)(l-1) + (l-1-\beta-\delta)\left(\frac{l}{2}-\frac{1}{2}+\frac{\beta}{2}+\frac{\delta}{2}\right)\right] \\
&= \frac{1}{l}\left[\frac{l^2}{2} + \beta l + \delta l - \beta^2 - \beta\delta - \delta^2 - \beta - \delta - \frac{1}{2}\right] \\
\frac{dEWT_{AS,1}(S)}{d\beta} &= \frac{1}{l}\left[l - 2\beta - \delta - 1\right] = 0 \Longrightarrow \\
\beta &= \frac{l-1-\delta}{2} \\
\frac{dEWT_{AS,1}(S)}{d\delta} &= \frac{1}{l}\left[l - \beta - 2\delta - 1\right] = 0 \Longrightarrow \\
\delta &= \frac{l-1-\beta}{2}
\end{aligned}$$

Combining $\beta = \frac{l-1-\delta}{2}$ and $\delta = \frac{l-1-\beta}{2}$ we get $\beta = \delta = \frac{1}{3}(l-1)$. These values maximize $EWT_{AS,1}(S)$, so choose β and δ as small as possible.

$$\begin{aligned}
EWT_{AS,2}(S) &= \frac{1}{l}\left[\alpha\left(1-\frac{\alpha}{2}\right) + \beta(1-\alpha) + \gamma\left(1-\alpha-\frac{3}{2}\right) + \delta(1-\alpha-\gamma)\right. \\
&\quad + (1-\alpha-\gamma)\left(\frac{1}{2}-\frac{\alpha}{2}-\frac{\gamma}{2}\right) + (l-2-\beta-\delta)(0) + \beta(\alpha) \\
&\quad \left. +\delta(\alpha+\gamma) + (1-\beta-\delta)(1)\right] \\
&= \frac{1}{l}\left[\alpha\beta - \alpha\delta + \frac{3}{2}\right]
\end{aligned}$$

If $\beta > \delta$, we want to choose $\alpha \to 0$ to minimize t_2. If $\beta < \delta$, we want to choose $\alpha \to 1$ to minimize t_2. We want to choose $\beta \to 0$ and $\delta \to 0$, so our resulting schedule approaches $1^1 2^{l-1}$ in the limit as $\beta \to 0, \delta \to 0, \alpha \to 0$ or 1. But this is just the schedule in Lemma 5.1, so it follows that this case is never optimal.

So the best schedule is when $\alpha = \gamma = \frac{1}{3}$ and $\beta = \delta = 1$.

Now we choose the length, l, of the schedule.

$$\begin{aligned}
EWT_{AS}(S,p_1) &= \frac{1}{l}\left[\left(\frac{l^2}{2}+2l-\frac{11}{2}\right)p_1+\left(\frac{7}{6}\right)(1-p_1)\right]\\
&= \frac{1}{l}\left[\left(\frac{l^2}{2}+2l-\frac{20}{3}\right)p_1+\frac{7}{6}\right]\\
\frac{dEWT_{AS}(S,p_1)}{dl} &= \frac{1}{2}p_1+\frac{20}{3l^2}p_1-\frac{7}{6l^2}=0 \Longrightarrow\\
\frac{1}{l^2}\left(\frac{7}{6}-\frac{20}{3}p_1\right) &= \frac{1}{2}p_1 \Longrightarrow\\
l &= \frac{1}{3}\sqrt{\frac{21}{p_1}-120}
\end{aligned}$$

This is valid for $l \geq 3 \Longleftrightarrow p_1 \leq \frac{5}{33}$.

$$\begin{aligned}
\frac{d^2t}{dl^2} &= \frac{-40}{3}p_1 l^{-3}+\frac{7}{3}l^{-3}\\
&= \frac{-40}{3}p_1\left[\frac{1}{9}\left(\frac{21}{p_1}-120\right)^{-2}\frac{1}{3}\sqrt{\frac{21}{p_1}-120}\right]\\
&\quad +\frac{7}{3}\left[\frac{1}{9}\left(\frac{21}{p_1}-120\right)^{-2}\frac{1}{3}\sqrt{\frac{21}{p_1}-120}\right]\\
&= \frac{1}{9}(21-120p_1)^{-1}\frac{1}{9}\frac{1}{3}\sqrt{\frac{21}{p_1}-120}\geq 0\ \forall p_1\leq\frac{7}{88}
\end{aligned}$$

So, we have a minimum value of EWT_{AS} at $l=\frac{1}{3}\sqrt{\frac{21}{p_1}-120}$, and it follows that $EWT_{AS}(S,p_1)$ is minimized for $S=1^{\frac{1}{3}}2^1 1^{\frac{1}{3}}2^1 1^{\frac{1}{3}}2^{l-3}$ when $l=\frac{1}{3}\sqrt{\frac{21}{p_1}-120}$.

When we choose l this way the expected waiting time is

$$\begin{aligned}
EWT_{AS}(S,p_1) &= \frac{3l^2+12l-40}{6l}p_1+\frac{7}{6l}\\
&= \frac{3\left(\frac{1}{9}\left(\frac{21}{p_1}-120\right)\right)+12\left(\frac{1}{3}\sqrt{\frac{21}{p_1}-120}\right)-40}{6\left(\frac{1}{3}\sqrt{\frac{21}{p_1}-120}\right)}+\frac{7}{6\left(\frac{1}{3}\sqrt{\frac{21}{p_1}-120}\right)}\\
&= 2p_1+\frac{1}{3}\sqrt{21p_1-120p_1^2},\text{ for } 0\leq p_1\leq\frac{7}{88}
\end{aligned}$$

5.6 Proof of Lemma 5.4

$S = S_N$ has the form $1^{\frac{1}{n}}2^1 1^{\frac{1}{n}}2^1 \cdots 1^{\frac{1}{n}}2^{l-n}, l \geq n+1, 0 \leq p_1 \leq \frac{2n+1}{(n+1)(2n^2+n+1)}$. We want to find the value of l that minimizes $EWT_{AS}(S, p_1)$.

$$\begin{aligned}
EWT_{AS,1}(S) &= \frac{1}{l}\left[1\,(l-1) + n\left(l - \frac{3}{2}\right) + (l-n-1)\left(\frac{l}{2} + \frac{n}{2} - \frac{3}{2}\right)\right] \\
&= \frac{1}{l}\left[\frac{1}{2}l^2 + (n-1)\,l + \left(-\frac{1}{2}n^2 - \frac{1}{2}n + \frac{1}{2}\right)\right] \\
EWT_{AS,2}(S) &= \frac{1}{l}\left[1\left(\frac{1}{2n}\right) + n\left(\frac{1}{n}\right) + (l-n-1)\,(0)\right] \\
&= \frac{1}{l}\left[1 + \frac{1}{2n}\right] \\
EWT_{AS}(S,p_1) &= \frac{1}{l}\left[\left(\frac{1}{2}l^2 + (n-1)\,l + \left(-\frac{1}{2}n^2 - \frac{1}{2}n - \frac{1}{2} - \frac{1}{2n}\right)\right)p_1 \right. \\
&\quad \left. + \left(1 + \frac{1}{2n}\right)\right] \\
\frac{dEWT_{AS}(S,p_1)}{dl} &= \left(\frac{1}{2} + \left(\frac{1}{2}n^2 + \frac{1}{2}n + \frac{1}{2} + \frac{1}{2n}\right)l^{-2}\right)p_1 + \left(-1 - \frac{1}{2n}\right)l^{-2}
\end{aligned}$$

$$\begin{aligned}
\frac{dEWT_{AS}(S,p_1)}{dl} &= 0 \Longrightarrow \\
0 &= \frac{1}{2}p_1 + l^{-2}\left(\left(\frac{1}{2}n^2 + \frac{1}{2}n + \frac{1}{2} + \frac{1}{2n}\right)p_1 + \left(-1 - \frac{1}{2n}\right)\right) \Longrightarrow \\
l &= \frac{1}{n}\sqrt{\frac{n\,(2n+1)}{p_1} - n\,(n+1)\,(n^2+1)}
\end{aligned}$$

This is valid for $l \geq n+1 \Longleftrightarrow p_1 \leq \frac{2n+1}{(n+1)(2n^2+n+1)}$.

So, we have a minimum value of EWT_{AS} at $l = \frac{1}{n}\sqrt{\frac{n(2n+1)}{p_1} - n\,(n+1)\,(n^2+1)}$, and it follows that $EWT_{AS}(S, p_1)$ is minimized for $S = 1^{\frac{1}{n}}2^1 1^{\frac{1}{n}}2^1 \cdots 1^{\frac{1}{n}}2^{l-n}$ when $l = \frac{1}{n}\sqrt{\frac{n(2n+1)}{p_1} - n\,(n+1)\,(n^2+1)}$.

When we choose l this way, the expected waiting time is

$$\begin{aligned}
t &= \frac{1}{l}\left[\left(\frac{1}{2}l^2+(n-1)\,l+\left(-\frac{1}{2}n^2-\frac{1}{2}n-\frac{1}{2}-\frac{1}{2n}\right)\right)p_1+\left(1+\frac{1}{2n}\right)\right] \\
&= \frac{1}{\frac{1}{n}\sqrt{\frac{n(2n+1)}{p_1}-n\,(n+1)\,(n^2+1)}}\left[\left(\frac{1}{2}\left(\frac{1}{n^2}\left(\frac{n\,(2n+1)}{p_1}-n\,(n+1)\left(n^2+1\right)\right)\right)\right.\right. \\
&\quad +(n-1)\,\frac{1}{n}\sqrt{\frac{n\,(2n+1)}{p_1}-n\,(n+1)\,(n^2+1)}+\left(-\frac{1}{2}n^2-\frac{1}{2}n-\frac{1}{2}-\frac{1}{2n}\right)\Bigg)\,p_1 \\
&\quad \left.+\left(1+\frac{1}{2n}\right)\right] \\
&= (n-1)\,p_1+\frac{1}{n}\sqrt{n\,(2n+1)\,p_1-n\,(n+1)\,(n^2+1)\,p_1^2}
\end{aligned}$$

Chapter 6 Mixing Time-Division and Frequency-Division

We have examined frequency-division scheduling and time-division scheduling of both dynamic and static data. Now we consider a different type of data and a different way of scheduling. We consider data that has a specific bandwidth requirement, such as a video transmission with a required quality of service. We also wish to schedule many dynamic items in addition to this video data. We attempt to combine these two types of data on a single broadcast channel, and we search for the best way to schedule this data.

In Section 6.1, we describe this new type of scheduling and the methods of scheduling we will be examining. In Section 6.2, we compute expected waiting times for the time-division schedules and the mixed schedules. Then, in Section 6.3 we compute bounds on where the two scheduling methods are better.

6.1 Introduction

We require that the video has a fixed portion of the total bandwidth, with a small amount of buffering at the receiver allowed, and we try to minimize the expected waiting time for the dynamic items. An application where this type of scheduling would be used is for sending both video and data over a digital cable line. Users require that the video arrives at some minimum bandwidth, but they also want dynamic information with as little waiting as possible. We assume we have a channel of bandwidth $B = 1$, of which a fraction α is required for video broadcast. In addition, we have k dynamic data items of length l that we wish to send on this channel. We consider three ways of doing this.

The first is time-division. This is shown in Figure 6.1. We break up the video

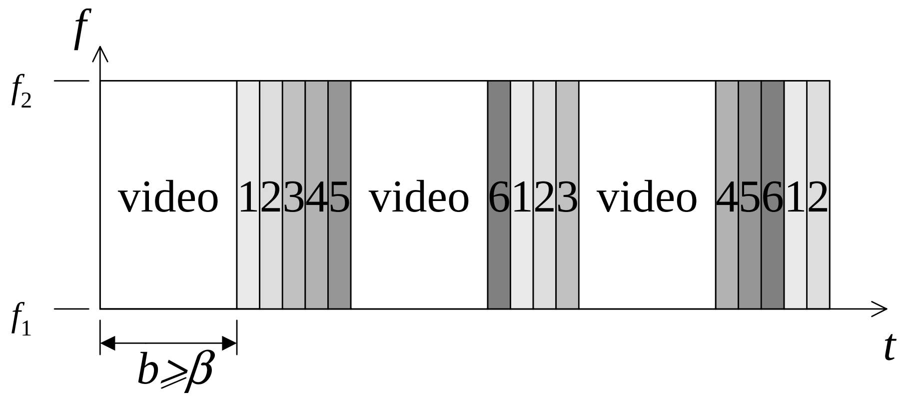

Figure 6.1: Time-division scheduling for fixed-bandwidth video data and 6 dynamic data items.

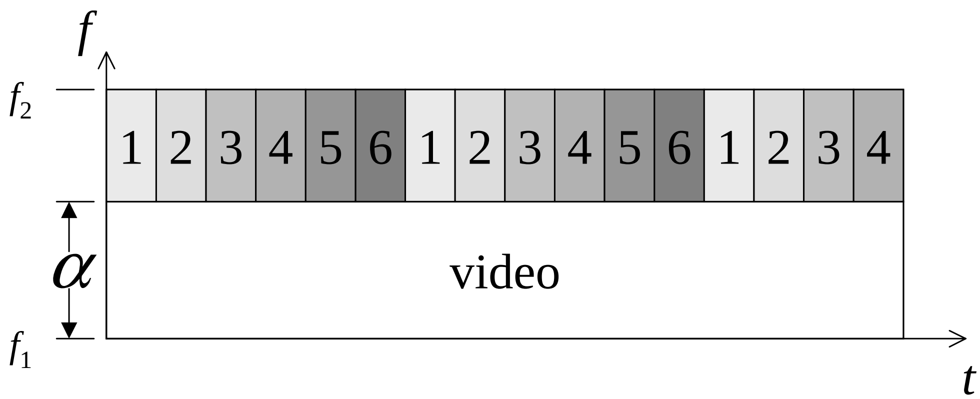

Figure 6.2: Mixed time-division and frequency-division scheduling for fixed-bandwidth video data and 6 dynamic data items.

data into packets and use time-division scheduling to send these packets and the other items. We require that α of our bandwidth is spent sending the video packets, and the other $1 - \alpha$ is used for the other items. We also impose a limit on how small the video packets can be. We assume that sending packets of size less than β is not practical, so our actual packet size will be some value $b \geq \beta$. We will not split up the dynamic data items, so these packets will have a spacing that is some integral multiple of l.

The second broadcast method will be a mix of time-division and frequency-division. This is shown in Figure 6.2. We first allocate bandwidth $\alpha \cdot B$ for the video, and then send the data items using time-division on the remaining bandwidth $(1 - \alpha) \cdot B$. For each of these broadcast methods, we assume that all items are the same length and have the same demand probability, so our ordering of the data items is simply $1, 2, 3, \ldots, k$. This allows us to better isolate the effects of the scheduling method rather than the actual scheduling of the data with this method.

The third method we consider is frequency-division, where we give the streaming data one channel, and each item gets its own channel in the bandwidth that remains. However, we can use the results of Chapter 2 to show that the mix is always better, since we are essentially comparing time-division and frequency-division now. So, we will ignore frequency-division from now on and only compare time-division and the mix of time-division and frequency-division.

6.2 Expected Waiting Times

6.2.1 Time-Division

For time-division scheduling, we compute the expected waiting time for an item by considering the spacing of its broadcasts. We know that the video takes bandwidth α, so the average number of video packets between each broadcast of an item is $\frac{\frac{kl}{1-\alpha}}{\frac{b}{\alpha}} = \frac{kl\alpha}{b(1-\alpha)}$. So, the average spacing of an item is $\overline{s} = kl + b\left(\frac{kl\alpha}{b(1-\alpha)}\right) = \frac{kl}{1-\alpha}$. We assume that our video packets are spaced as evenly as possible, so this is one of $kl + b\left\lfloor\frac{kl\alpha}{b(1-\alpha)}\right\rfloor$ or $kl + b\left\lceil\frac{kl\alpha}{b(1-\alpha)}\right\rceil$.

If the items were all spaced evenly, then the expected waiting time would simply be half the spacing, or $\frac{kl}{2(1-\alpha)}$. However, this is typically not the case, so this value is a lower bound for the expected waiting time. We know that the spacing will be one of two values, depending on whether we use the floor or the ceiling function above. We will call the first possible spacing a, so the second will be $a + b$. We will have spacing a some fraction, $1 - q$, of the time and spacing $a + b$ the remaining fraction, q, of the time. So, our average spacing is $a + qb$.

We can compute the values a and q. The value a is simply the smaller spacing, so $a = kl + b\left\lfloor\frac{kl\alpha}{b(1-\alpha)}\right\rfloor$. We compute q as the value such that $a + qb = \overline{s}$, giving $q = \frac{kl\alpha}{b(1-\alpha)} - \left\lfloor\frac{kl\alpha}{b(1-\alpha)}\right\rfloor$. It is now easy to compute the expected waiting time for time-division. We simply look at the probability of starting to listen during a spacing of that length, and the expected waiting time when we arrive within a spacing of each length. This gives us

$$\begin{aligned} EWT_{TD} &= \frac{(1-q)a}{a+qb}\left(\frac{1}{2}a\right) + \frac{q(a+b)}{a+qb}\left(\frac{1}{2}(a+b)\right) \\ &= \frac{1}{2(a+qb)}\left(a^2 - qa^2 + qa^2 + 2qab + qb^2\right) \\ &= \frac{a^2 + 2qab + qb^2}{2(a+qb)} \\ &= \frac{a+qb}{2} + \frac{q(1-q)b^2}{2(a+qb)} \end{aligned}$$

We see here that the expected waiting time is simply the ideal expected waiting time, or half the average spacing, plus a penalty term for not achieving the ideal spacing.

6.2.2 Mixed Frequency-Division and Time-Division

With the mixed scheduling, we simply compute the expected waiting time as half of the spacing, plus the time to receive the item minus the length of the item sent at full bandwidth. This is

$$\begin{aligned} EWT_{mix} &= \frac{kl}{2(1-\alpha)} + \frac{l}{1-\alpha} - l \\ &= \frac{kl}{2(1-\alpha)} + \frac{2l\alpha}{2(1-\alpha)} \\ &= \frac{(k+2\alpha)l}{2(1-\alpha)} \end{aligned}$$

We can write this, using the notation we used for time-division, as

$$EWT_{mix} = \frac{a+qb}{2} + \frac{(a+qb)\alpha}{k}$$

We see here that again we have a sum of half the average spacing plus a penalty term for using less than full bandwidth to send the items.

6.3 Bounds on Regions of Optimality

We have computed the expected waiting times for each method of scheduling. Now, we would like to find where each method performs better. Finding exactly when each method is better is difficult. We will show bounds on where each is better. First, we show when time-division is better.

6.3.1 Time-Division

From our expected waiting time computations, we can compute that time-division is better when

$$\begin{aligned} \frac{q(1-q)b^2}{2(a+qb)} &< \frac{(a+qb)\alpha}{k} \iff \\ q(1-q)b^2k &< 2\left(a+qb\right)^2\alpha \iff \\ q(1-q)b^2k &< 2\left(\frac{kl}{1-\alpha}\right)^2\alpha \iff \\ b^2 &< 2\alpha\left(\frac{kl^2}{(1-\alpha)^2}\right)\frac{1}{q(1-q)} \iff \\ b^2 &< \frac{2}{q(1-q)}\cdot\frac{\alpha}{(1-\alpha)^2}\cdot kl^2 \end{aligned}$$

The value of $q(1-q)$ is at most $\frac{1}{4}$, so $\frac{2}{q(1-q)} \geq 8$. So, we can say that time-division is better when $b^2 < \frac{2}{q(1-q)} \cdot \frac{\alpha}{(1-\alpha)^2} \cdot kl^2$, or

$$b < \frac{2l\sqrt{2\alpha k}}{1-\alpha}$$

We can not use this equation to say anything about when the mixed schedule is better, however, because $\frac{2}{q(1-q)}$ is not bounded above.

6.3.2 Mixed Frequency-Division and Time-Division

To find where mixing time-division and frequency-division is better than time-division alone, we calculate the difference in their expected waiting time. However, we make an assumption about the spacings between items, first. We see that if $b \leq \frac{\alpha}{1-\alpha}kl$, then there will always be at least one packet of video between each instance of an item. However, if $b > \frac{\alpha}{1-\alpha}kl$, then there will either be zero or one packet of video between each instance of an item. We saw that the expected waiting time for time-division was simply half the average spacing, plus a scheduling penalty, for not having equal spacing. So, if we can space items equally, we will have an optimal schedule. When $b \leq \frac{\alpha}{1-\alpha}kl$, we can simply increase b so that we always achieve the lower of the two possible numbers of video packets between items. So, when $\beta \leq \frac{\alpha}{1-\alpha}kl$, we can schedule with no penalty using time-division. Thus, time-division is always, better, since mixing always has a nonzero penalty.

When $\beta > \frac{\alpha}{1-\alpha}kl$, penalty-free scheduling is not possible, since the lower number of video packets is zero, and having zero packets of video between items means video is never sent. So, it is only this case where we will find the mix performing better than time-division. For time-division, we use our expected waiting time formula with the a's and q's, but replace the a's and q's with their equivalents in α, b, k, and l. This gives us

$$\begin{aligned}
EWT_{TD} &= \frac{1}{\left(1-\frac{\alpha}{1-\alpha}\frac{kl}{b}\right)(kl)+\left(\frac{\alpha}{1-\alpha}\frac{kl}{b}\right)(kl+b)} \cdot \left(\left(1-\frac{\alpha}{1-\alpha}\frac{kl}{b}\right)(kl)\left(\frac{kl}{2}\right)\right. \\
&\quad \left. + \left(\frac{\alpha}{1-\alpha}\frac{kl}{b}\right)(kl+b)\left(\frac{kl+b}{2}\right)\right) \\
&= \frac{1}{kl-\frac{\alpha}{1-\alpha}\frac{(kl)^2}{b}+\frac{\alpha}{1-\alpha}\frac{(kl)^2}{b}+\frac{\alpha}{1-\alpha}(kl)} \cdot \left(\frac{1}{2}(kl)^2-\frac{\alpha}{1-\alpha}\frac{(kl)^3}{2b}\right. \\
&\quad \left. +\frac{\alpha}{1-\alpha}\frac{kl}{2b}\left((kl)^2+2klb+b^2\right)\right) \\
&= \frac{1}{kl\left(1+\frac{\alpha}{1-\alpha}\right)} \cdot \left(\frac{1}{2}(kl)^2-\frac{\alpha}{1-\alpha}\frac{(kl)^3}{2b}+\frac{\alpha}{1-\alpha}(kl)^2+\frac{\alpha}{1-\alpha}\frac{klb}{2}\right) \\
&= \frac{1-\alpha}{kl} \cdot \left(\left(\frac{\alpha}{1-\alpha}+\frac{1}{2}\right)(kl)^2+\frac{\alpha}{1-\alpha}\left(\frac{1}{2}klb\right)\right) \\
&= \left(\alpha+\frac{1-\alpha}{2}\right)(kl)+\alpha\left(\frac{1}{2}b\right)
\end{aligned}$$

We know that the expected waiting time for mixing is $EWT_{mix} = \frac{kl}{2(1-\alpha)} + \frac{l\alpha}{1-\alpha}$.

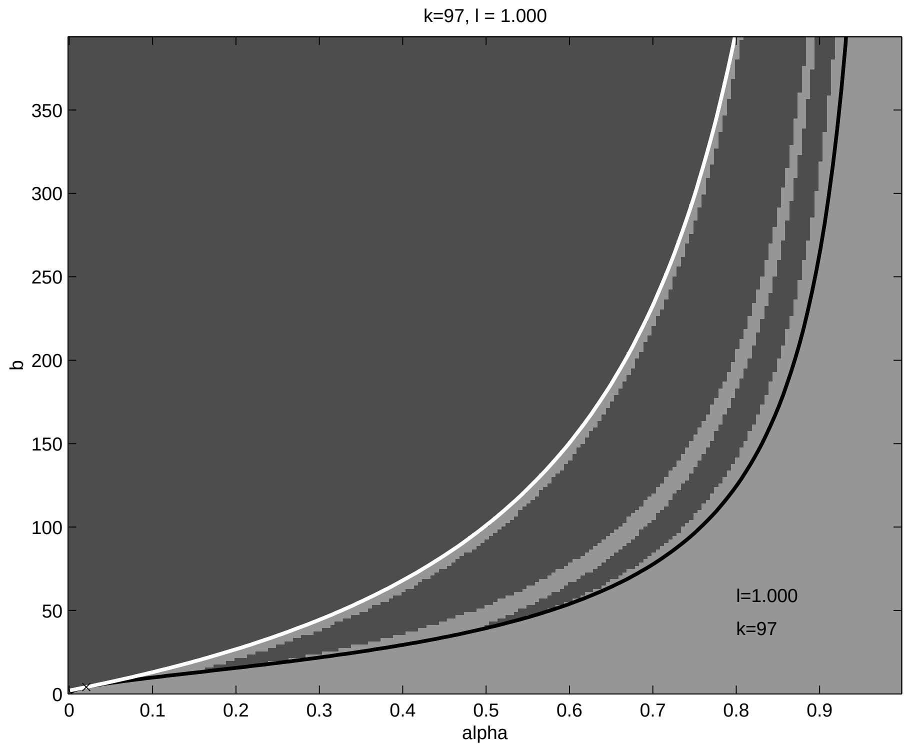

Figure 6.3: The regions where time-division (light gray) and mixed (dark gray) scheduling are better, for $k = 97$ items of length $l = 1$. The white line is the bound from Section 6.3.1 and the black line is the bound from Section 6.3.2.

We can now determine when mixing is better by computing when time-division has a higher expected waiting time.

$$\begin{aligned}
\frac{kl}{2(1-\alpha)} + \frac{l\alpha}{1-\alpha} &< \left(\alpha + \frac{1-\alpha}{2}\right) kl + \alpha\left(\frac{1}{2}b\right) \iff \\
kl + 2l\alpha &< 2\alpha(1-\alpha)\,kl + (1-\alpha)^2\,kl + \alpha(1-\alpha)\,b \iff \\
b &> \frac{1}{\alpha(1-\alpha)}\left(kl + 2l\alpha - 2\alpha(1-\alpha)kl - (1-\alpha)^2 kl\right) \iff \\
b &> \frac{1}{\alpha(1-\alpha)}\left(\alpha^2 kl + 2\alpha l\right) \iff \\
b &> \frac{l}{1-\alpha}(\alpha k + 2)
\end{aligned}$$

So, we have bounds on the region where mixing is better and the region where time-division is better. Between these bounds, it is not as easy to compute which method is better to use. Figure 6.3 shows the nature of these bounds and the region between them for $k = 97$ and $l = 1$.

Chapter 7 Conclusions and Future Directions

We have examined the scheduling of data over a broadcast channel. We looked at different types of data, and different methods of scheduling. In Chapter 2, we looked at frequency-division scheduling. We showed that optimal frequency-division scheduling is possible and showed how to do it. Then we compared these schedules to time-division schedules. We showed that for any frequency-division schedule for items of equal length, we can generate a time-division schedule with lower expected waiting time. It is somewhat surprising that even non-optimal time-division schedules can always perform better than the optimal frequency-division schedules.

Motivated by our result in Chapter 2, we looked in Chapter 3 for optimal time-division schedules. We showed that for two dynamic items of arbitrary lengths there is a simple schedule that is optimal. We compared these simple schedules and found the values of demands and lengths for which each is optimal.

In Chapter 4, we looked at how splitting items into smaller pieces could help us reduce expected waiting time. We looked at two items as in Chapter 3, but allowed them to be split in half. We narrowed our search for optimal schedules to a small set of "irreducible" schedules. For equal lengths, we compared these and found optimal schedules. We provided evidence that suggests that the set of optimal schedules may be the same for arbitrary lengths.

In Chapter 5, we looked at two dynamic data items of equal length and considered splitting them into arbitrarily many pieces. We then proved some lemmas that support the idea that the optimal schedules are somewhat simple. All of the good schedules that we found involve splitting the item with lower demand into n pieces of size $\frac{1}{n}$ and scheduling them in a simple pattern.

In Chapter 6, we looked at a new type of data, one which requires a constant, fixed bandwidth. We looked at a new method of scheduling that combines frequency-division and time-division and showed that it is sometimes better than either frequency-division or time-division alone.

There are many areas remaining for future research. Continuing the research in this thesis, we could try to show something about frequency-division and time-division for different length items. For the simple case of two dynamic items with $p_1 = \frac{19}{20}$, $p_2 = \frac{1}{20}$, $l_1 = 1$, and $l_2 = 19$ the optimal frequency-division schedule is actually better than the optimal time-division schedule. However, if we allow splitting of item 2, then time-division can perform better. The trade-offs between number of splits and ratio of lengths could be explored.

We could also try to prove or disprove that the optimal schedules for two different length dynamic data items are the same as for equal-length data items when we can split them in half. This would be very interesting if the added generality of arbitrary lengths did not change which schedules are optimal.

Another more ambitious goal would be to prove the proposition in Chapter 5 about optimal schedules with arbitrary splits. Extending the work with two items to arbitrary numbers of items would also be nice. Work examining different ways to mix data types or scheduling methods would also be interesting. For example, a new type of schedule might work well for many items, where some of them are dynamic and the others are static.

In a broader sense, looking at issues such as error detection and correction, multiple broadcast servers, and client constraints such as power and mobility would be interesting. The idea of clients caching and prefetching are also issues that people are working on. On the practical side, it would be good to study what we can do in a real system by getting numbers for the parameters we consider, such as bandwidth, file sizes, and expected waiting times.

Bibliography

[1] D. Aksoy, M. Altinel, R. Bose, U. Centemel, M. Franklin, J. Wang, S. Zdonik, "Research in Data Broadcast and Dissemination," *Proceedings of 1st International Conference on Advanced Multimedia Content Processing*, Osaka, Japan, Nov. 1998.

[2] D. Aksoy, M. J. Franklin, "Scheduling for Large-Scale On-Demand Data Broadcasting," IEEE INFOCOM, San Francisco, CA, March 1998.

[3] M. H. Ammar, J. W. Wong, "On the Optimality of Cyclic Transmission in Teletext Systems," *IEEE Transactions on Communications*," vol. 35, pp. 68-73, January 1987.

[4] A. Bar-Noy, R. Bhatia, J. Naor, B. Schieber, "Minimizing Service and Operations Cost of Periodic Scheduling," 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1998.

[5] A. Bar-Noy, A. Nisgav, B. Patt-Shamir, "Nearly Optimal Perfectly-Periodic Schedules," PODC 2001, Newport, RI, August 2001.

[6] A. Bestavros, "AIDA-based Real-Time Fault-Tolerant Broadcast Disks," Technical Report 96-001, Computer Science, Boston University, January 5, 1996.

[7] K. Foltz, J. Bruck, "Robustness of Time-Division Schedules for Internet Broadcast," ISIT 2002, Lausanne, Switzerland.

[8] K. Foltz, J. Bruck, "Splitting the Scheduling Headache," ISIT 2000, Sorrento, Italy.

[9] K. Foltz, J. Bruck, "Splitting Schedules for Periodic Internet Broadcast," *IEEE Trans. Info. Theory*, vol. 48, pp. 345-358, Feb. 2002.

[10] K. Foltz, J. Bruck, "Time-Division is Better Than Frequency-Division for Periodic Internet Broadcast," ISIT 2001, Washington, DC. [Full version available at `http://www.paradise.caltech.edu/ETR.html`] [submitted to *IEEE Trans. Info. Theory*].

[11] M. J. Franklin, S. Zdonik, "Dissemination-Based Information Systems," IEEE Data Engineering Bulletin, vol. 19, no. 3, September 1996.

[12] M. Franklin, S. Zdonik, "Data in Your Face: Push Technology in Perspective," Proc. ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 98), Seattle, WA, June, 1998, pp. 516-519.

[13] S. Hameed, *Scheduling Information Broadcast in Asymmetric Environment*, Masters of Science thesis, Texas A&M University, May 1997.

[14] S. Hameed, N. H. Vaidya, "Log-time Algorithms for Scheduling Single and Multiple Channel Data Broadcast," MOBICOM'97, Budapest, September 1997.

[15] R. Hassin, N. Megiddo, "Exact Computation of Optimal Inventory Policies over and Unbounded Horizon," *Mathematics of Operations Research*, 80:534-546, 1991.

[16] S. Jiang, N. H. Vaidya, "Scheduling Algorithms for a Data Broadcast System: Minimizing Variance of the Response Time," Technical Report 98-005, Computer Science, Texas A&M University, February 4, 1998.

[17] S. Khanna, S. Zhou, "On Indexed Data Broadcast," Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC-98), pp. 463-472, New York, 1998.

[18] C. Kenyon, N. Schabanel, "The Data Broadcast Problem with Non-Uniform Transmission Times," Proc. of the 10th ACM-SIAM Symp. on Discrete Algorithms (SODA '99), pp. 547-556, January 1999.

[19] H. V. Leong, A. Si, "Database Caching Over the Air-Storage," *The Computer Journal* 40(7): 401-415, 1997.

[20] V. Liberatore, "Caching and Scheduling for Broadcast Disk Systems," ALENEX00: Second Workshop on Algorithm Engineering and Experiments, San Francisco, January 7-8, 2000.

[21] C. J. Su, L. Tassiulas, "Broadcast Scheduling for Information Distribution," Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFO-COM'97), vol. 1, pp. 109-117, 1997.

[22] L. Tassiulas, C. J. Su, "Optimal Memory Management Strategies for a Mobile User in a Broadcast Data Delivery System," *IEEE JSAC Special Issue on Networking and Performance Issues of Personal Mobile Communications*, 15(7):1226-1238, September 1997.

[23] N. H. Vaidya, S. Hameed, "Data Broadcast in Asymmetric Wireless Environments," First International Workshop on Satellite-based Information Services (WOSBIS), Rye, NY, November 1996.

[24] L. Xu, "Efficient and Scalable On-Demand Data Streaming Using UEP Codes," ACM Multimedia 2001, Ottowa, Ontario, Canada, October 2001.

[25] S. Zdonik, M. J. Franklin, R. Alonso, S. Acharya, "Are 'Disks in the Air' Just 'Pie in the Sky'?," IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994.