Periodic Broadcast with Dynamic Server Selection

Ewa Kusmierek, Yingping Lu and David H.C. Du
Digital Technology Center and Department of Computer Science and Engineering
University of Minnesota
{kusmiere, lu, du}@cs.umn.edu

***Abstract*— Service replication is an effective way to address resource requirements and resource availability problem. Dynamic service selection enables clients to choose a server offering the best performance. Proper server selection is especially important for video streaming over the Internet due to its high bandwidth requirements. However, given the length of a typical video transmission, the server selected prior to sending a request may not be an optimal one for the duration of the whole transmission. We examine the possibility of switching to another server during an on-going transmission for periodic broadcast schemes. Due to the timing requirements typical for periodic broadcast scheme the server switch may cause playback disruption. We analyze the magnitude of the problem and propose an easy to implement solution. We define the criterion, additional to the bandwidth availability for example, according to which a new server should be selected. The client is also required to delay playback by the amount of time bounded by the server transmission offset. We propose also an alternative method to ensure uninterrupted playback that relies on proxy caching.**

I. Introduction

Multimedia applications are often characterized by relatively high requirements of the resources such as server I/O bandwidth and network bandwidth. A number of techniques can be used to address these requirements. Service replication is one of them. In a large scale multimedia system a set of videos is available from multiple geographically distributed servers. Multiple servers offer increased resource and service availability, reduction of resource requirements such as network bandwidth and increased fault tolerance. A client may use a number of criteria such as proximity to server, current server load and availability of network bandwidth to choose a server to request the video from. However, this initial choice may not be an optimal one throughout the whole transmission given that the length of the transmission may be substantial. Hence, the client should be able to switch to a different server in response to changes in the network bandwidth availability or a change in the server status. Server change during a on-going reception may also be beneficial in case of client mobility.

The server change can be fairly easily implemented if the video is transmitted on a per-client basis. Once the reception from one server starts, the client can request the remaining part of the video from a different server. However, unicast streaming is quite demanding in terms of server network and I/O bandwidth. Periodic broadcast (PB) improves the scalability of the resource usage for popular videos but introduces timing requirements for the video reception by a client. The client has to receive a number of video segments within specified time intervals relative to the beginning of the reception in order to ensure uninterrupted playback. We investigate whether, given these time constraints, a server switch is possible during the reception of a video in the PB mode.

In the ideal case two servers would be perfectly synchronized, i.e., would be transmitting the same data at the same time and server switch would not affect the reception. However, even then the delay between the client and each of the servers is most likely different and the same data from two different servers is reaching the client at different times. Given, that there is certain offset between transmissions from two different servers, we show that the client may face two problems upon server switch. Some frames may not be received at all and some frames may be received after their playback time. We analyze the magnitude of these problems, i.e., the number of frames affected and the delay with which frames arrive past their deadline, under the assumption that there is an upper bound on the offset between transmissions from two different servers.

We classify PB schemes into two groups and show how server switch affects video reception in each group. We next show how continuous playback can be ensured. A straightforward solution is to delay the playback of the video with respect to the beginning of the reception. However, the worst case delay required is a function of the transmission time of the largest PB segment. Such a delay may be unacceptable. We propose a way to ensure uninterrupted playback without a need for an excessive delay. Our solution relies on the client's ability to choose a server to switch to such that its transmission offset from the original server transmission satisfies certain condition. We propose also an alternative solution that relies on a proxy server to cache chunks of frames and deliver them to clients if needed.

The paper is organized as follows. The related work is presented in Section II. In Section III we characterize several typical PB scheme and classify them into two groups. We formulate the problem of server switch during an on-going PB reception in Section IV. The analysis of a server switch during reception of a single segment for both groups of PB schemes is presented in Section V and is followed by the analysis of the multiple segment reception in Section VII. We propose proxy caching to eliminate problems caused by server switch in Section VIII. We conclude the paper in Section IX.

II. Related Work

Multimedia applications based on video streaming have high resource requirements in terms of network bandwidth, server

This work is partially supported by NSF Grant EIA-0224424 and DTC Intelligent Storage Consortium

I/O bandwidth and storage space. In addition to the high requirements another factor makes delivering video over Wide Area Network (WAN) a challenge. Traffic load and consequently the available bandwidth change dynamically and in an unpredictable way. The resource requirements on a large-scale has been addressed in a number of ways.

Rate adaptation allows to adjust bandwidth requirements to bandwidth availability typically by controlling the quality of a video. The quality adjustments are preformed by excluding some frames from transmission [1], or reducing frame sizes. The latter is done by adjusting the compression ratio of an online encoder [2], transcoding a pre-encoded video [3] or dropping a layer of a hierarchically encoded video [4]. The rate adaption techniques require collecting information about the status of the network path [5].

Video caching close to a client by a proxy server [6–8] allows to reduce bandwidth requirements. A proxy can deliver (part of) a video stream directly to clients without consuming bandwidth along the path between a central server and clients. Caching by a proxy may also reduce server load.

Service replication offers three major benefits: 1) an increase in the amount of resources available such as I/O bandwidth and storage space, 2) increased service availability and fault tolerance due to replication, and 3) reduction of resource requirements such as WAN bandwidth. Given multiple servers the problem of server selection has been addressed in [9–12]. The dynamic server selection schemes have been shown to yield better performance than static selection of the server closest to the client. The performance is typically defined in terms of response time and depends on a number of factors such as network path characteristics (RTT and available bandwidth) and server load. These factor are highly variable and change quickly, hence, a need for dynamic server selection. The goal of a server selection scheme is to estimate the performance a client would experience from a given server.

In [9] the client is assumed to have a list of the addresses of available service providers. A set of latency and available bandwidth measurements are preformed prior to sending a request in order to choose the server providing the best performance at a given time. Technique presented in [10] relies on application level anycast service. An anycast resolver maps anycast domain name into one or more IP addresses. In addition to the mapping information each resolver maintains also an independent metric database with the server performance data. Hence, a name can be mapped into an address of the server that satisfies user-specified performance criteria the best. A method for collecting network path and server metric for multimedia server selection has been proposed in [12]. The method combines end-to-end with hop-by-hop measurements performed by a set of selected nodes.

Dynamic server selection is very important for multimedia applications due to a large amount of data transferred, high bandwidth requirements and the length of a typical video session. The last factor indicates that it may not be sufficient to select the best performance server prior to the beginning of the transmission due to dynamically changing network conditions. In our work we explore the possibility of switching from one server to another during an on-going transmission.

III. BACKGROUND

Periodic Broadcast schemes have been designed to reduce server bandwidth consumption for popular videos. PB schemes require the server to continuously broadcast a number of segments a video is partitioned into on a number of transmission channels. Each PB scheme defines a set of reception rules for the client that ensure that all frames are received on time. The client has to tune into each of the channels within specified time interval relative to the beginning of the reception. Hence, there are timing issues associated with the PB reception.

A number of PB schemes have been proposed [13–20]. They differ in video segmentation, selection of transmission rates and consequently reception rules for the client. They can be divided into two groups depending on the start-up delay, or more precisely reception delay. Schemes in the first group require that the client waits for the transmission of the beginning of the first segment to start reception. Hence, the reception delay is bounded by the transmission time of the first segment. The playback of the first segment starts immediately with the reception, while playback of the subsequent segments may be delayed with respect to their reception times. Some of the segments are received earlier and buffered until their playback. Pyramid Broadcast [13], Skyscraper Broadcast [17] and Greed Disk-conserving Broadcast [18] are examples of such schemes. Figure 1 illustrates Skyscraper Broadcast using temporal bandwidth map introduced in [20]. The lower part represents server broadcast area with the shaded rectangles marking the client reception. The top part corresponds to the playback area.

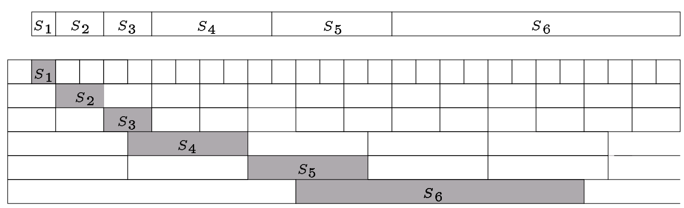


Fig. 1. Temporal bandwidth map for Skyscraper Broadcast

PB schemes in the second group allow the client to start the reception at any time (without delay) but the playback must be delayed until the first segment is completely received. Hence, the start-up delay is equal to the transmission time of the first segment. The same rule applies to the subsequent segments, each segment is received just-in time for its playback. Polyharmonic Broadcast [16], Server Optimal Broadcast [20] and Fibonacci Broadcast [20] belong to this group. Figure 2 shows temporal bandwidth map for Fibonacci Broadcast. We refer to these two groups as *delayed reception PB* and *delayed playback PB*, respectively.

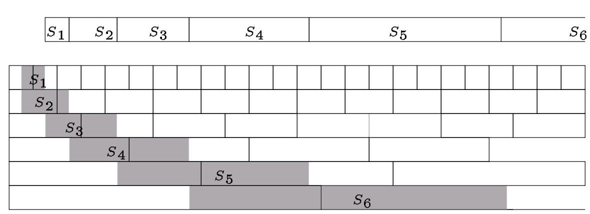


Fig. 2. Temporal bandwidth map for Fibonacci Broadcast

The transmission rate for each segment can be smaller as well as larger than the playback rate. The former is possible only if playback is delayed with respect to the reception time. However, delayed reception PB scheme uses transmission rate larger than or equal to the playback rate. Note that all schemes require client to receive more than one segment at a time. The number of segments received simultaneously varies from two (Pyramid and Skyscraper) to all of the segments (Polyharmonic and Server Optimal). Appendix I contains summary parameters for different PB schemes.

IV. Problem Formulation

The settings under consideration are as follows. We assume that there are multiple servers broadcasting a set of popular videos. For a given video the same PB scheme is used by all servers, i.e., the same segmentation and transmission rates. However, there may be a time offset between transmissions of any two servers given by a difference between the times when a given frame is transmitted. Since the servers transmit each video segment continuously with a period T equal to the segment transmission time, the offset can be observed as one of two values whose sum is equal to T. We define the offset between two servers as the smaller of two values: $\Delta = \min(x, T-x)$ as illustrated in Figure 3.

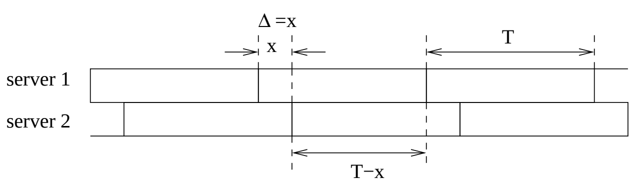


Fig. 3. Time offset between servers

A video is received by a client according to the corresponding PB reception schedule. The client starts receiving the video from the server, to which we refer as the *source server*. At some point in time the video stream from the source server is replaced with a video stream from the *destination server*. We assume at first that the change is instantaneous. In this way we can analyze the effect of the server offset on the playback alone. Next we relax this assumption by considering the delay between the time when the stream from the source server stops reaching the client and the time when the stream from the destination servers starts reaching the client. We analyze how server switch affects the playback. The goal is to ensure an uninterrupted playback, i.e., ensure that all frames are received on time.

We now introduce the notation that will be used throughout the paper. Let $t_0^i = 0$ denote the beginning of the reception of the ith segment by the client and $t_e^i = t_0^i + T_i$ denote the end of the reception, where T_i is the time needed to receive the whole ith segment. Let also t_1^i and t_2^i denote the first time the beginning of the ith segment is transmitted after t_0^i by the source and the destination server, respectively. The server offset observed by the client for the ith segment is equal to $|t_2^i - t_1^i| \in \{\Delta, T_i - \Delta\}$. The notation is summarized in Table I.

TABLE I Notation

S_i	length of ith segment
T_i	transmission length of the ith segment
r_t^i	transmission rate of the ith segment
t_0^i	beginning of the reception of the ith segment
t_e^i	end of the reception of the ith segment
t_p^i	beginning of the playback of the ith segment
r_p	playback rate
α_i	ratio of transmission to playback rate
t_c	time of server switch
Δ	time offset between transmissions from two servers
t_1^i	transmission time of the beginning of the ith segment by the source server
t_2^i	transmission time of the beginning of the ith segment by the destination server
S_m	size of missing segment part
d	delay required for all frames to arrive on time
d_m	delay required to patch the missing frames on time

V. Server Switch Influence on Single Segment Reception

We start the analysis of the effect the server offset has on the reception and playback of a video by considering a single segment first. We find out that the effect is different for each of the two groups of PB schemes identified in Section III. Thus, we present the results separately for each group.

A. Delayed Reception PB

In the first case the client has to wait for the transmission of the beginning of the segment by the source server to start the reception. The reception and playback start at the same time. Server switch during the reception can cause two problems: 1) some frames may be received past their playback deadline and 2) some frames may not be received during the regular reception time $(t_0^i, t_0^i + T_i)$.

We show that the frames arriving late miss their playback deadline by no more than Δ, the server offset. Thus, the playback disruption can be avoided by delaying the playback by at most Δ. This result tells us that we can control the delay required by controlling the offset between server transmissions. In practice it is not possible to maintain perfect synchronization of the servers but it is possible to keep the offset within certain limit. Another factor affecting the server offset observed by the client is the difference between delays from the client to each of the servers. Henceforth we assume that Δ denotes an upper bound on the server offset defined at the client: $|t_2^i - t_1^i| \leq \Delta$.

We show also that the amount of data not received from either server is bounded by Δr_t, where r_t is the transmission rate. Frames included in the part not received are transmitted by the destination server starting at time $t_0^i + T_i$. Hence, the frames can be received on time by delaying the playback and extending the reception time beyond $t_0^i + T_i$, both by no more than Δ.

Time Dependencies: Let t_c denote the time when the server switch occurs and let t_p be the beginning of playback. In order to simplify the notation we drop the superscript indicating the segment number. Since the client has to wait for the source server to transmit the beginning of the segment to start reception, then $t_0 = t_1$. The playback starts at the same time $t_p = t_0$. Both reception and playback should end at $t_e = t_0 + T$. Note that the destination server transmits the beginning of the segment no earlier than the source server from the client's point of view : $t_2 \geq t_1$. If $t_2 = t_1$ the playback is continuous and the disruption does not occur. Therefore, we concentrate on the case when $t_2 > t_1$. Note also that for a given server offset bound Δ the offset between the source and the destination server from the client point of view can be either equal to or smaller than Δ or equal to or larger than $T - \Delta$. We assume for now that $t_2 - t_1 \leq \Delta$.

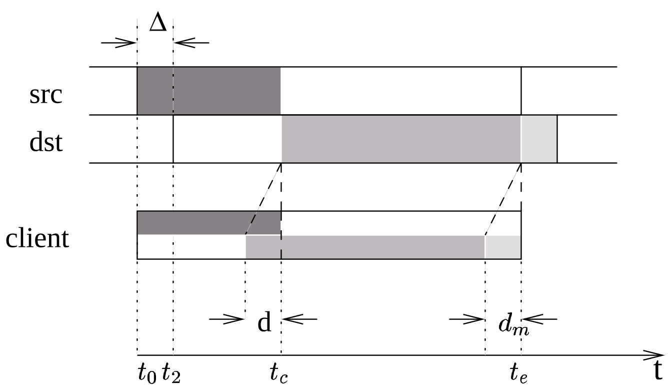


Fig. 4. Delay for delayed reception and $t_c > t_2$

1) Missed Playback Deadline: We now examine the problem of frames arriving past their playback deadline. We distinguish two cases depending on when server switch occurs with respect to the transmission time of the beginning of the segment by the destination server. We assume first that the server switch occurs after the destination server transmits the beginning of the segment ($t_c > t_2$). Such a situation is illustrated in Figure 4. The top part shows the transmission by both servers, the bottom the reception by a client. The dark gray area represents data received from the source server, the medium gray area data received from the destination server. The portion of the segment transmitted by the destination server during time interval $(t_c, t_c + \min(t_2 - t_1, t_e - t_c))$ has already been received by the client from the source server. Recall that the destination server is "behind" the source server with its transmission. Thus, the client does not receive any *new* frames for the time interval of length $\min(t_2 - t_1, t_e - t_c) \leq \Delta$. If the amount of data buffered at the client at this time is not sufficient to play during this time, *all remaining frames*, starting with the frame arriving at $t_c + (t_2 - t_1)$, will arrive past their playback time.

Similar situation occurs when the server switch takes place before the destination server transmits the beginning of the segment (Figure 5). During time interval (t_c, t_2) the destination server transmits the tail of the segment that the client buffers for the later playback, and during time interval $(t_2, t_2 + (t_c - t_0))$ the destination server transmits the beginning of the segment that the client has already received from the source server. Thus, the client must have enough frames buffered to support continuous playback during the time interval of length $(t_2 - t_c) + (t_c - t_0) \leq \Delta$.

In a general case the transmission rate can be equal to or

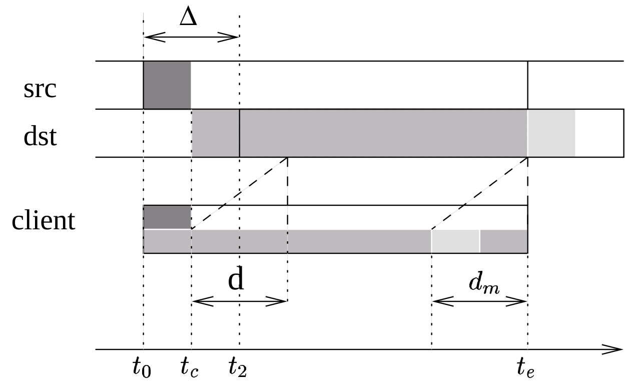


Fig. 5. Delay for delayed reception and $t_c < t_2$

larger than the playback rate. Let $r_t = \alpha r_p$ and $\alpha \geq 1$, where r_p is the playback rate. At the time of server switch t_c, the amount of data buffered by the client is: $r_p(\alpha - 1)(t_c - t_0)$. The buffering is due to the potential difference between the transmission rate and the playback rate. Such an amount is played during the time interval of length $(\alpha - 1)(t_c - t_0)$. Therefore the playback delay needed to avoid disruption is:

$$d \leq \max(0, \Delta - (\alpha - 1)(t_c - t_0)) \leq \Delta \tag{1}$$

We find that a delay no longer than the maximum server offset Δ is sufficient to avoid disruption caused by late frame arrival.

2) Missing Frames: Due to the time offset between servers part of the segment is not received by the client during regular reception time (t_0, t_e). The missed part is represented by the light gray area in Figures 4, 5 and 6. The size of the missed portion of the segment is equal to:

$$S_m = \min(t_2 - t_1, t_c - t_0, t_e - t_c) r_t \leq \Delta r_t \tag{2}$$

Note that $t_c - t_0$ expresses the length of the reception from the source server and $t_e - t_c$ expresses the length of the reception from the destination server. The size of the missing part is bounded by the amount of data that is transmitted during the time interval of length equal to the server offset. The size can be smaller than that if the reception interval from either server is shorter than Δ. Such a situation ($t_e - t_c < t_2 - t_1$) is illustrated in Figure 6. Given the upper bound on the server offset, the client cannot miss more than is transmitted during time interval of length Δ.

In order to avoid playback disruption, the client has to patch the missing frames on time. We now examine the earliest time the client has a chance to receive the missing frames and compare it with their playback deadlines. Note that patching can take place by reception from the destination server after the end of the regular reception time t_e. The transmission offset of the missed part from the beginning of the segment, i.e., the difference between transmission time of the beginning of the segment and the transmission time of the first bit of the missed part, is equal to $\max(t_e - t_2, t_c - t_1)$.

If $t_e - t_2 \geq t_c - t_1$ (Figure 4 and 5), then the missed part is transmitted by the destination server at $t_2 + t_e - t_2 = t_e$ and the playback of the missed part starts at $t_0 + t_e - t_2$. The delay required to avoid interruption is determined by the difference between the two times $d_m = t_2 - t_0$.

If $t_c - t_1 > t_e - t_2$ (Figure 6), then the missed part is transmitted by the destination server at $t_2 + t_c - t_1$ and the playback of the missed part starts at $t_0 + t_c - t_1$. The delay required in this case is equal to $d_m = t_2 - t_0$. Thus, the delay needed to avoid playback disruption is

$$d_m = t_2 - t_0 \leq \Delta \tag{3}$$

The delay is no longer than the worst case server offset Δ. Note that if the server offset from the client point of view is $t_2 - t_1 \geq T - \Delta$, then the delay required to receive the missed part on time would be also greater than or equal to $T - \Delta$. Hence, a small playback delay is sufficient only *if the destination server transmission is delayed by at most Δ w.r.t. the source server transmission*.

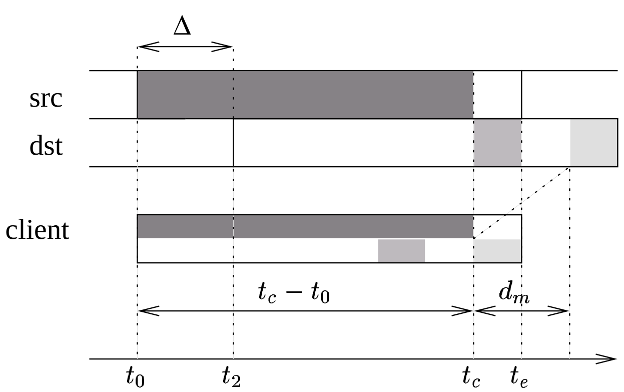


Fig. 6. Delay for delayed reception and $t_c - t_0 > t_e - t_2$

B. Delayed Playback PB

The PB schemes in the second group allow the client to start receiving the segment at any time ($t_0 \leq t_1$) but the playback has to be delayed until the whole segment is received: $t_p = t_0 + T$. Note that no part of the segment received during regular reception time (t_0, t_e) can miss its playback deadline. However, a part of the segment is not received from either server during the regular reception time. The client has another chance to receive the missing frames after t_e but the frames may be received past their playback deadlines. Hence, we next analyze playback delay that would allow to avoid disruption.

The delay is determined by the difference between the next transmission time (after t_e) of the missed part and its playback time. We assume that the transmission rate can be equal to, larger or smaller than the playback rate. Therefore, the delay necessary to avoid disruption is computed as a difference between the transmission time and the playback time of the *last bit* of the missed portion. Let f denote the transmission offset of the missing part. Playback time of the last bit is then equal to $t_p + \alpha f + \frac{S_m}{r_p}$ and its transmission time is equal to $t_2 + f + \frac{S_m}{\alpha r_p}$ (or $t_2 - f + \frac{S_m}{\alpha r_p} + T$). The last element in both sums accounts for the length of the playback and length of the transmission of the missed part, respectively. The delay is given by:

$$d_m = \max(0, t_2 - t_p + (1-\alpha)f + (1-\alpha)\frac{S_m}{\alpha r_p}) \tag{4}$$

The value of the offset f depends on the relation between three variables: t_1, t_2 and t_c. In addition, given the time offset between transmissions from the source server and the destination server bounded by Δ, the offset as observed by the client can be $|t_2 - t_1| \leq \Delta$ or $|t_2 - t_1| \geq T - \Delta$ (Figure 7). Note that in both cases the size of the missing part of the segment is bounded by $r_t\Delta$. Hence, we compute delay in all six cases of different ordering of t_1, t_2 and t_c and for both cases of server offset that can be observed by the client. We present here only the value of the missed part transmission offset f and the resulting delay bound in each case. The details can be found in the Appendix II. Table II summarizes the results.

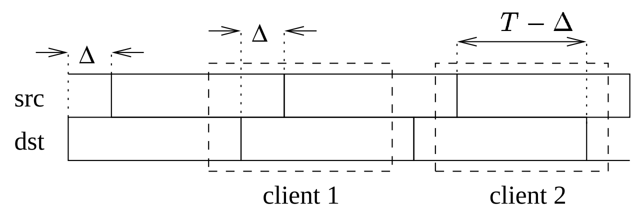


Fig. 7. Time offset between server for different clients

We draw the following conclusions from the analysis of delay. When the server offset from the client point of view is no larger than Δ, the delay in the first three cases in Table II) is bounded by a function of the server offset only, while in the remaining three cases, the delay is bounded by a function of the segment transmission time in addition to Δ. The first three cases are characterized by the fact that the source server transmits the beginning of the segment earlier than the destination server (from the client point of view). Hence, in order to avoid delay proportional to the segment transmission time, the client should choose a destination server for which $t_2 - t_1 \leq \Delta$. If this condition is satisfied delaying playback by at most $(1-\alpha)\Delta$ allows to avoid playback disruption. The client also has to receive the missing frames past the regular reception time. Note that for transmission rate no smaller than the playback rate ($\alpha \geq 1$) no delay is necessary.

When the server offset from the client point of view is no smaller than $T - \Delta$, no delay is needed in all cases but two, provided that the transmission rate is no smaller than playback rate. In case 5) and 6) the delay is bounded by a function of the server offset Δ. For $\alpha < 1$ certain delay is required in all cases, and in two of these cases the delay is proportional to the segment transmission time. The conditions under which such a delay is necessary cannot be defined as clearly as for $|t_2 - t_1| \leq \Delta$. The time of server switch, which cannot be controlled, plays an important role.

C. Disruption Avoidance Guidelines

The analysis of the server switch influence on the reception of a single segment results in the following guidelines for ensuring uninterrupted playback.

TABLE II DELAY FOR DELAYED PLAYBACK PB

case		offset f	delay upper bound	
			$\|t_2 - t_1\| \le \Delta$	$\|t_2 - t_1\| \ge T - \Delta$
1)	$t_1 < t_c < t_2$	$\max(t_c - t_1, t_e - t_2)$	$\Delta(1-\alpha)$	$T(1-\alpha)$
2)	$t_c < t_1 < t_2$	$t_e - t_2$	$\Delta(1-\alpha)$	$\Delta(1-\alpha)$
3)	$t_1 < t_2 < t_c$	$\max(t_e - t_2, t_c - t_1)$	$\Delta(1-\alpha)$	$T(1-\alpha)$
4)	$t_c < t_2 < t_1$	$\max(t_e - t_2, t_e - t_1 + t_c - t_0)$	$T(1-\alpha)$	$\Delta - \alpha T$
5)	$t_2 < t_1 < t_c$	$t_c - t_1$	T	Δ
6)	$t_2 < t_c < t_1$	0	T	Δ

In the group of PB schemes with delayed reception the client needs to select as the destination server whose transmission is delayed by Δ w.r.t. the source server transmission from the client point of view. Then it is sufficient to delay the playback by Δ and extend the reception time by Δ.

In the group of PB schemes with delayed playback the recommendation depends on the relation between the transmission rate and the playback rate. If the transmission rate is no lower than playback rate there are two options. The client may choose a destination server such that its transmission is delayed by no more than Δ with respect to the source server transmission. The second option is to choose, as a destination, the server whose offset with the source server is no less than $T - \Delta$, which is less strict than the previous condition but requires also delaying playback by Δ. The condition is less strict since it is defined only for the offset between servers but not their order.

If transmission rate is lower than playback rate the only way to avoid playback disruption without introducing long delay is to choose, as a destination, the server whose transmission is delayed by no more than Δ with respect to the source server transmission and delay the playback by $\Delta(1-\alpha)$.

VI. INFLUENCE OF STREAM INTERRUPTION ON SEGMENT RECEPTION

In reality a switch between the servers is not instantaneous. The reception may be interrupted, i.e., the client may not receive any data from either server for certain amount of time. Such a situation takes place once video stream from the source server does not reach client any more and the stream from the destination server does not reach the client yet. We now consider the influence of such an interruption on the playback. We find that if the length of time interval during which the client does not receive any data is longer than Δ, then the delay required to avoid playback disruption may be much longer than Δ. Such a delay is needed to patch a part of the segment, which is not received due to the interruption.

Let us consider the delayed reception case. If the server switch takes place before transmission of the segment beginning by the destination server $t_c < t_2$, the interruption causes an increase in the size of the missed part but does not increase the delay if interruption is shorter than Δ. All missing frames are in one contiguous chunk. If the interruption is longer than Δ another part of the segment located just behind frames received from the source server is missing. The additional missing part is represented by the black rectangle in Figure 9. The delay d_{m2} required to receive this part on time is proportional to the segment transmission time.

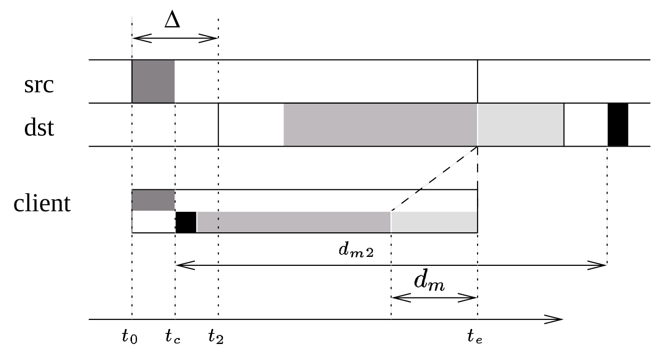


Fig. 8. Delay with reception interruption and $t_c < t_2$

Similar result holds in the case when the server switch takes place after destination server transmits beginning of the segment: $t_c \ge t_2$. The size of the missed part is increased if the interruption is longer than the server offset. Figure 8 illustrates such a situation.

Therefore, it is beneficial to set also a lower bound on the server offset in addition to the upper bound Δ. Let δ denote the lower bound, then δ should be no smaller than the length of reception interruption. Henceforth we assume that the reception interruption is no longer than the server transmission offset.

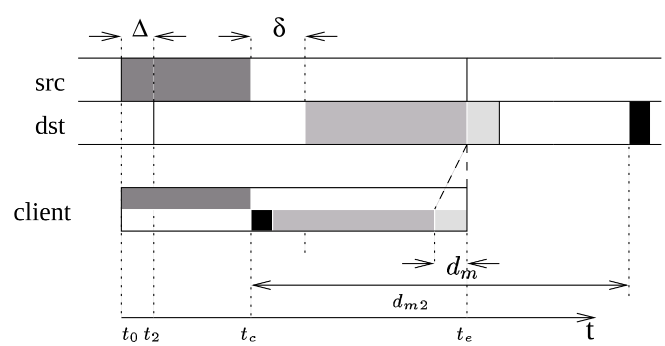


Fig. 9. Delay with reception interruption and $t_c \ge t_2$

VII. SERVER SWITCH INFLUENCE ON MULTIPLE SEGMENT RECEPTION

In any periodic broadcast scheme the client typically receives multiple segments simultaneously. Therefore, a server switch may affect the reception of multiple segments. Through the analysis of the server switch influence on the reception of a single segment we find two ways to avoid playback disruption. We now ask the question whether either of these two methods can prevent playback disruption when multiple segments are affected.

Delaying the playback of a single segment naturally requires delaying the playback of all segments, thus prevents playback disruption for all of them. We find that it may not be possible to choose a destination server such that one of the offset conditions is satisfied for a number of consecutive segments affected by the server switch. However, we find also that it is possible to choose a destination that satisfies one of two offset conditions such that playback disruption may be avoided with a small delay.

A. Delayed Reception PB

Given that multiple segments are affected by the server switch, the analysis provided in Section V-A can be applied separately to each of these segments. Choosing a destination server for which $t_2 - t_1 \leq Delta$ and delaying playback of the entire video by Δ allows all frames to be received on time. Note that if the condition $t_2^i - t_1^i \leq \Delta$ holds for one segment, it holds for all segments.

Since the client has to receive the missing frames past the regular reception time of each affected segment, the bandwidth requirements may be temporarily increased. The client may have to tune to a larger number of channels than required by a PB scheme.

B. Delayed Playback PB

From the analysis of a single segment reception we have learned that it is beneficial to choose a destination server whose transmission is delayed by at most Δ relative to the transmission of the source server or whose offset is larger than $T_i - \Delta$. Such a choice allows to avoid playback disruption by delaying the playback by only a small amount of time. The question arises whether it is always possible to satisfy these conditions for *more than one* segment.

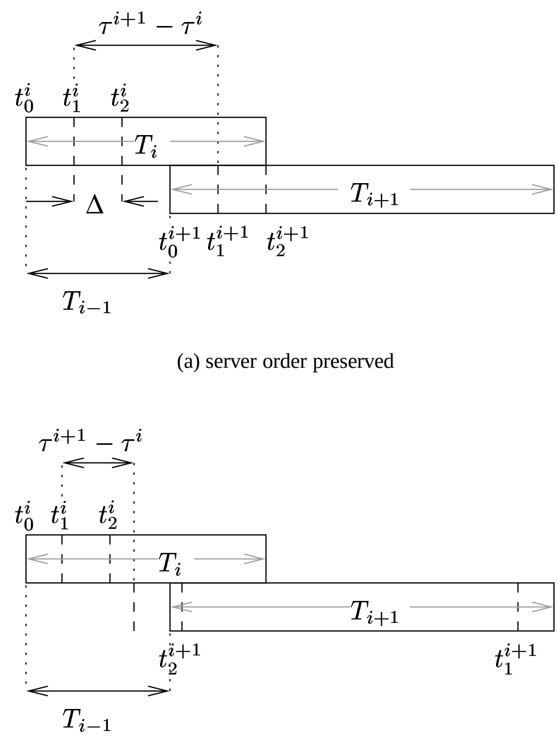
We first assume that the server offset condition: $t_2^i - t_1^i = \Delta$ holds for the ith segment and show that it is possible that the condition is not satisfied for the $(i+1)st$ segment. However, the offset for the $(i+1)st$ segment is such that a small delay is still sufficient to ensure uninterrupted playback.

1) Fibonacci Periodic Broadcast: We first analyze Fibonacci scheme (FPB). We assume that the server offset condition is satisfied for the ith segment: $t_2^i - t_1^i \leq \Delta$. There are two possible cases for the relation between t_1^{i+1} and t_2^{i+1} as presented in Figure 10. The Figure outlines the reception of the ith and $(i+1)st$ segment. The dashed lines mark transmission of the beginnings of both segments by the source server and the destination server.

In case (a) the server order is preserved. In case (b) the order of the server transmissions is reversed for the $(i+1)st$ segment, i.e., $t_1^{i+1} - t_2^{i+1} \geq T_{i+1} - \Delta$, the transmission from the destination server is ahead of the transmission from the source server. The reversal occurs when the beginning of the $(i+1)st$ segment is transmitted by the source server less than Δ before the beginning of this segment reception. This implies that the source server transmits the beginning of the ith and $(i+1)st$ segment within time interval no larger than $t_0^{i+1} - t_0^i$.

We now show that such a situation can take place. Let τ^i denote a set of transmission times of the beginning of the ith segment: $\tau^i = kT_i$, where $k = 0, 1, \ldots$. Note that the sizes of any two consecutive segments given by two consecutive Fibonacci numbers are relative primes. The difference between τ^{i+1} and the latest τ^i such that $\tau^i \leq \tau^{i+1}$ takes a value from the following set $\{0, 1, 2, \ldots, T_i\}$, where the time unit is equal to the transmission time of the first segment T_1. Then, the server order reversal takes place when $\tau^{i+1} - \tau^i \leq t_0^{i+1} - t_0^i = T_{i-1}$. The order reversal is possible since $\tau^{i+1} - \tau^i$ can be smaller than T_{i-1} and there is no lower bound on $t_1^i - t_0^i$.

We conclude that it is possible that server order condition: $t_2 - t_1 \leq \Delta$ is satisfied for the ith segment but for the $(i+1)st$ segment the following is true: $t_1 - t_2 \geq T - \Delta$.



(a) server order preserved

(b) server order reversed

Fig. 10. Server order in Fibonacci Broadcast

2) Server Optimal Broadcast: A similar argument can be made for Server Optimal Broadcast scheme. For the server order to be reversed for the $(i+1)st$ segment, the source server must transmit the beginning of the $(i+1)st$ segment less than Δ before the end of this segment reception: $t_1^{i+1} \geq t_e^{i+1} - \Delta$. At the same time the beginning of the ith segment must be transmitted no later than $t_1^i \leq t_e^i - \Delta$ as illustrated in Figure 11. Both conditions imply that $t_1^{i+1} - t_1^i \geq t_e^{i+1} - t_e^i = \alpha T_i$. In the Server Optimal scheme $T_i = T_1(1+\alpha)^{i-1}$. Then $\frac{T_{i+1}}{T_i} = 1 + \alpha$. If α is an integer (transmission rate is a multiple of playback rate), then $\tau^{i+1} - \tau^i \in \{0, T_i\}$. Otherwise $\tau^{i+1} - \tau^i \in \{0, (1+r_t), 2(1+r_t), \ldots, T_i\}$. Therefore, server order can be reversed if $\alpha T_i \leq T_i$, which means that $\alpha \leq 1$.

3) Polyharmonic Broadcast: Polyharmonic Broadcast has the same condition for the server reversal order as Server Op-

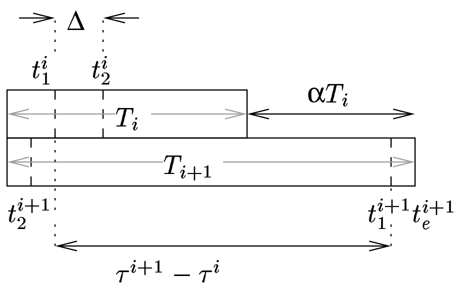


Fig. 11. Server order in Server Optimal Scheme

timal scheme: $t_1^{i+1} - t_1^i \geq t_e^{i+1} - t_e^i = \alpha_i T_i$. Since different channels have different transmission rates, α has also different values. Specifically $\alpha_i = \frac{1}{m-i+1}$. In this case $\frac{T_{i+1}}{T_i} = \frac{m+1}{m+i-1}$, where m is an integer greater or equal to 1. Thus, $\tau^{i+1} - \tau^i \in \{0, 1, ..., T_i\}$. The server order is reversed if $\alpha_i \leq 1$, which implies $i \geq 2 - m$. For $m = 1$ the reversal cannot happen only for the first two segments, while $m > 1$, the reversal can happen for any pair of two consecutive segments.

C. *Conclusions for Multiple Segment Reception*

We conclude that for delayed playback schemes it is not possible to choose, as a destination, the server whose transmission satisfies the following offset condition $t_2 - t_1 \leq \Delta$ for all affected segments. However, it is possible to choose a server such that one of the following two conditions is true for all segments: 1) $t_2 - t_1 \leq \Delta$ or 2) $t_1 - t_2 \geq T - \Delta$. From the analysis of the single segment reception we know that the delay required to avoid playback disruption in this case is bounded by Δ (cases 1)-3) for $|t_2 - t_1| \leq \Delta$ and cases 4)-6) for $|t_2 - t_1| \geq T - \Delta$ in Table II).

The same conclusion holds for the case when the destination server is selected to satisfy less strict offset condition: $|t_2 - t_1| \geq \Delta$. The condition may not be satisfied for some segments. However, in this case the reversal of the condition to $|t_2 - t_1| \leq \Delta$ may lead to a large delay if $t_2 < t_1$.

Hence, a way to avoid playback disruption is to choose a destination server whose transmission is delayed by at most Δ with respect to the source server transmission and delay the playback by Δ.

VIII. Proxy Caching

In this section we present an alternative way to avoid playback disruption due to server switch. The method presented so far requires that the destination server is selected to satisfy certain criterion. The selection is based on the knowledge of the transmission offset between the server the client is currently receiving data from and a group of potential destination servers.

The alternative solution relies on the assistance of a proxy server. We assume that proxy server can cache part of each video and provide this data to clients allowing them to patch missing frames on time. The cached frames are delivered on-demand in the unicast mode. Thus, there are no timing constraints imposed by a PB scheme. We propose two different caching schemes with different storage space and I/O bandwidth requirements at the proxy: *static* and *dynamic* caching.

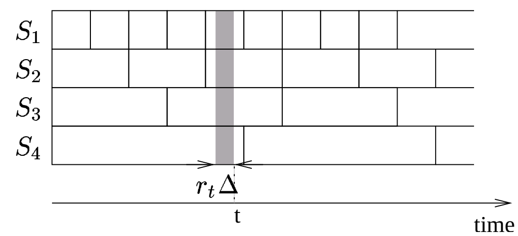


Fig. 12. Dynamic proxy cache content

A. *Dynamic Caching*

The idea of dynamic proxy caching is based on the observation that in the cases when the delay needed to patch the missing frames on time is proportional to the segment transmission time, the missing part is transmitted by the destination server just prior to the server switch time t_c. There are five such cases (Table II) and the proof of this behavior in each case is provided in Appendix III.

Given the fact that the amount of data the client misses due to server switch is no larger than $r_t\Delta$, i.e., proportional to the server offset upper bound, the missed part is transmitted during time interval $(t_c - \Delta, t_c)$. Thus, in order to provide missing data to the client, proxy needs to cache only $r_t\Delta$ most recently transmitted bits of each segment[1]. Since the buffer content at the proxy changes constantly, the client has to request the missing data from the proxy at the time of server switch, when the buffer contains data the client needs.

Such an arrangement requires that there are a number of proxies (at least one) associated with each server. Each proxy continuously receives all segments of a number of videos from its server and caches part of each segment as illustrated in Figure 12. The gray area in the Figure represents cache content at time t. When a given server is selected as a destination server by the client, the appropriate proxy is contacted with the request for data provided that the client cannot patch missing frames on time. The problem of proxy location is out of scope of this paper.

The storage space required at the proxy is equal to $\Delta \sum_{j=1}^{N} \sum_{i=1}^{n_j} r_t j^i$, where n_j is the number of segments of the jth video, r_t^j is the transmission rate, and N is the number of videos. Given that Δ is small, the buffer space requirement is not significant. However, the network and I/O bandwidth requirements are more significant: $\sum_{j=1}^{N} \sum_{i=1}^{n_j} r_t j^i$.

We assume that I/O bandwidth available to the proxy is limited and it may not be able to provide caching for all segments. Therefore, it is important to utilize storage space at the proxy in the most efficient way, i.e., cache segments that are more popular than others and that are more vulnerable. We now formulate a problem of efficient use of the proxy space given limited I/O capacity. We introduce the following set of variables $\{x_i^j\}$: x_i^j is equal to 1, if the ith segment of the jth video is cached by the proxy and 0 otherwise. The optimization problem is formally defined as follows:

[1] In reality this amount will be slightly larger to establish some safety margin

$$\begin{array}{lll} \text{minimize} & \sum_{j=1}^{N}\sum_{i=1}^{n_j}\frac{\alpha_j}{\alpha}P_i^j(1-x_i^j) & \\ \text{subject to} & \sum_{j=1}^{N}\sum_{i=1}^{n_j}x_i^j r_t^j + x_i^j r_t^j\frac{\alpha_j}{\alpha}P_i^j \le R & (5) \\ & x_i^j \in \{0,1\} & \end{array}$$

where α_j is the popularity of the jth video, P_i^j the length of the ith segment relative to the reception length of the entire jth video (e.g. Equation 7), $\frac{\alpha^j}{\alpha}$ is interpreted as a relative popularity of a video. Then $P_i^j\frac{\alpha^j}{\alpha}$ can be interpreted as a relative popularity of the ith segment of the jth video. We assume that the relative popularity determines the probability that the segment is affected by the server switch. Thus, the optimization function expresses the cumulative "probability" of disruption. Overall, the longer the segment and the more popular, the more likely that it will be cached.

The first constraint ensures that the I/O bandwidth capacity of the proxy is not exceeded. The first element of the summation in this constraint expresses I/O bandwidth usage due to *write* operations and the second due to *read* operations. The latter is dependent on the popularity of the video and the length of segment reception relative to the length of video reception. We assume that the storage space at a proxy is not an issue in this case due to a small amount of the data cached for each segment. The above problem is a typical 0-1 knapsack problem. Hence, it can be solved by dynamic programming method with computational complexity $\mathcal{O}(NR)$.

B. Static Caching

Given potential high I/O bandwidth required at the proxy due to dynamic caching, we consider also static caching, where frames are selected for caching in advance and the buffer content does not change. Static caching network and I/O bandwidth requirements are only due to *read* operations triggered by client requests.

Frames that a client can miss during regular reception time can be located anywhere within the segment. The offset of the missing part can take any value from 0 to T. Thus, in order to ensure uninterrupted playback we would have to cache the whole segment.

We assume that the storage space at the proxy is limited and that only part of a segment can be cached. In order to utilize the space efficiently we find which part of a segment is most likely to be missed. Such a part would be the best candidate for caching. Unfortunately all frames in a segment are almost equally likely to be missed.

However, we observe that in all schemes with delayed playback such that transmission rate is equal to or larger than the playback rate ($\alpha \ge 1$), a delay is required to avoid playback disruption in only four cases (Table II). In only two of these cases (5 and 6) the delay is proportional to the length of segment: $t_2 < t_1 < t_c$ and $t_2 < t_c < t_1$ for $t_2 - t_1 \le \Delta$. In the latter case it is the segment *prefix* that can be missed due to serve switch.. Thus, uninterrupted playback can be ensured by caching the prefix of size $r_t\Delta$. We show the in the former case a part of segment located close to the beginning is more likely to be missed. Therefore, we can reduce the probability of playback disruption the most by caching also segment prefix of the size as large as possible.

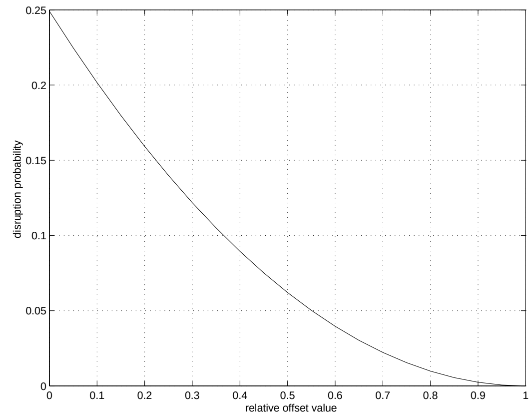


Fig. 13. Probability of playback disruption

Given that $t_2 < t_1 < t_c$ and $t_2 - t_1 \le \Delta$, the offset of the missing part $(t_c - t_1)$ can take any value from 0 to T. We now compute the probability that the offset of the missing portion of the segment is larger than y. Note that this probability is equal to the probability of disruption if the size of the cached prefix is equal to $y + \Delta r_t$. We assume that the source server can transmit beginning of the segment any time during the segment reception, i.e., let t_1 be a random variable uniformly distributed over interval $(0, T)$. Since $t_1 - t_2 \le \Delta$, we assume also that $t_1 - t_2$ is a random variable uniformly distributed over an interval of length $\min(t_e - t_1, \Delta) + \min(t_1 - t_0, \Delta) \le 2\Delta$. The probability that $t_2 \le t_1 \le t_c$ and that the offset is larger than y: $t_c - t_1 \ge y$, is computed as:

$$\begin{aligned} & P(t_2 \le t_1 \le t_c - y) = \\ & \quad \int_0^{\min(\Delta, T-y)} \frac{x}{T(x+\Delta)}\left(1 - \frac{x+y}{T}\right)dx \\ & + \int_{\min(\Delta,T-y)}^{\min(T-\Delta,T-y)} \frac{1}{2T}\left(1 - \frac{x+y}{T}\right)dx \qquad (6) \\ & + \int_{\min(T-\Delta,T-y)}^{T-y} \frac{\Delta}{T(T-x+\Delta)}\left(1 - \frac{x+y}{T}\right)dx \end{aligned}$$

The details of derivation are given in Appendix IV.

Figure 13 presents the above probability as a function of y expressed as a value relative to the segment length. We observe that indeed probability of disruption is larger for small values of y. The result is fairly intuitive, since for the offset $(t_c - t_1)$ to take a large value, t_1 must be close to t_0. Small value of y is possible for a larger range of t_1.

Given a small value of Δ, the probability that $t_2 \le t_1 \le t_c$ is equal to close to 0.25%. The same value expresses the probability of playback disruption without caching. We observe that larger value of Δ results in smaller probability since the chances that $t_2 \le t_c \le t_1$ are increased. As the value of y increases, the probability $P(t_2 \le t_1 \le t_c + y)$ decreases. With the size of the cached prefix equal to around 10% of the segment length, the probability of disruption is around 0.2. Given the probability of $t_2 \le t_1 \le t_c$ equal to 0.25, and probability of

$|t_2 - t_1| \leq \Delta$ equal to 0.5, the absolute probability of disruption is 0.025, provided that the server switch does take place.

Typical PB scheme has segment sizes forming an increasing sequence. In order to reduce disruption probability to the same value for all segments, the amount of data that must be cached is proportional to the segment size (provided that $r_t\Delta$ is much smaller than the smallest segment). Large segments are also more likely to be affected by the server switch since their reception is longer. We compute the probability that the ith segment is affected as a ratio of the segment reception length to the reception length of the whole video. For example, the following probability is derived for Fibonacci Broadcast:

$$P_i = \frac{T_i}{T_1 + \sum_{k=1}^{n-1} T_k} \tag{7}$$

Given a limited storage space at the proxy, we now formulate the problem of efficient usage of this space, i.e., the problem of selecting the prefix size for each segment of a video.

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} P_i P(t_2^i \leq t_1^i \leq t_c - y_i) \\ \text{subject to} & \sum_{i=1}^{n} \min(y_i + r_t^i \Delta, S_i) \leq B \\ & 0 \leq y_i \leq S_i \end{array} \tag{8}$$

The optimization function expresses the aggregate vulnerability to disruption given the size of the prefix cache for each segment.

Given a set of videos, we divide the total space at the proxy among the videos proportionally to their popularity. Thus, B is the space assigned to a single video. The above problem is a non-linear optimization problem. We solve the problem using numerical method to obtain a guideline and as a base for evaluation of the heuristic algorithm. We observe that prefix sizes selected by the numerical method are proportional to the segment lengths. Thus, we use this rule in the heuristic solution:

$$y_i = B \frac{S_i}{\sum_{i=1}^{n} S_i} \tag{9}$$

The partition of the buffer space among a number of videos is performed proportionally to the video popularity, i.e., the higher the popularity, the more space the video is assigned.

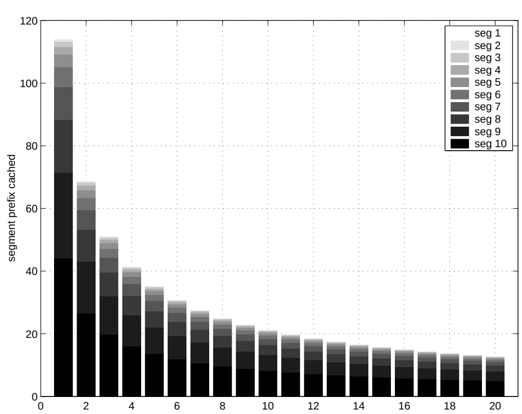
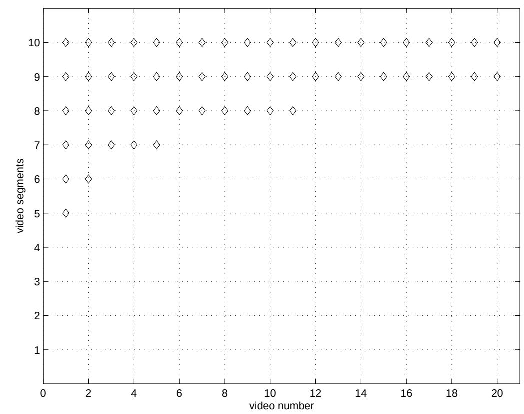


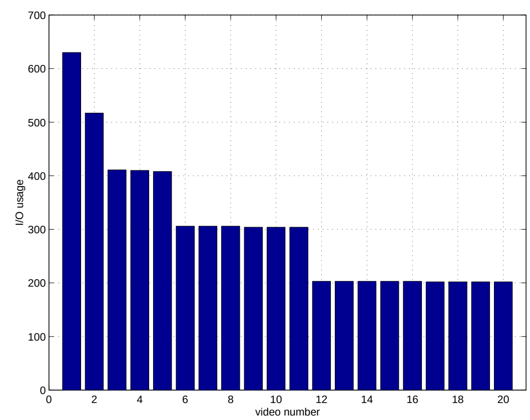
Fig. 14. Static Caching Solution

C. Numerical Tests Results

We evaluate the solution proposed to the static and dynamic caching through a number of numerical test. We consider a set of 20 videos, whose probability is drawn from Zipf distribution. For simplicity, we assume that all videos are of equal length 300. Fibonacci Broadcast is used to transmit each video.



(a) segment selection



(b) I/O bandwidth per video

Fig. 15. Dynamic Programming Solution

1) Static Caching: In the static caching settings the storage space available is sufficient to cache 10% of the aggregate video length. The storage space is divided among videos proportionally to their popularity. The prefix size for each segment of each video is chosen proportional to the segments length. Figure 14 presents prefix sizes selected, the videos are in order determined by their popularity.

2) Dynamic Caching: The solution for the dynamic caching problem is solved by exploring the trade-off between the I/O bandwidth required for staging a given segment and the probability of playback disruption without staging. Figure presents solution obtained with dynamic programming method for the proxy I/O capacity equal to 30% of the capacity required to stage all segments. Each of the 20 videos have 10 segments. A square in the plot indicates that a segment with a number of

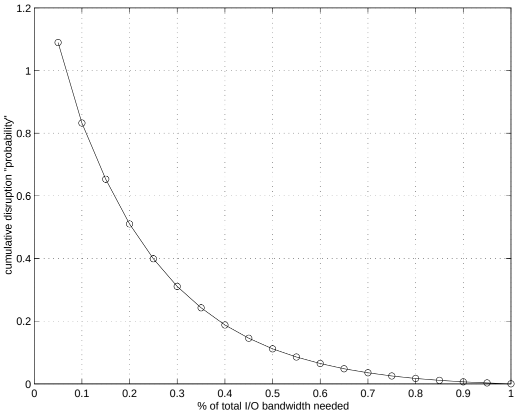


Fig. 16. Cumulative disruption "probability"

y-axis of the video whose number is marked on the x-axis is selected for staging. We observe that for the popular videos, short segments (with small index numbers) are selected, while for less popular videos the long segments are selected. Figure 15(b) presents the I/O bandwidth usage due to each video according to the segment selection given by dynamic programming. We present also the value of the objective function, the cumulative "probability" of disruption, as a function of the I/O bandwidth available at the proxy in Figure 16. The I/o bandwidth is specified as a value relative to the I/O bandwidth required to cache all segments of all videos.

IX. Conclusions

We have examined the possibility of client switching to a different server during an on-going reception of a periodically broadcasted video. Such a possibility allows the client to adjust to changing network conditions, which is a desirable feature given the length of a typical video transmission.

We found that the server switch may cause disruption of playback due to the fact that some frames may arrive past their playback deadlines and some frames are not received during the regular reception time. The magnitude of both problems is bounded by a function of the maximum offset between transmissions from two servers.

Both problems can be avoided by: 1) selecting, as a destination, the server whose transmission is delayed, from the client point of view, by no more than Δ, the maximum server offset and 2) delaying the playback by Δ relative to the original beginning of playback.

We have also proposed an alternative solution that relies on proxy caching. A proxy server caches part of each segment and delivers to the client when the missing part cannot be patched on time. In addition, the client still has to delay the video playback by Δ.

References

[1] C. Fung and S. Liew, "End-to-end frame-rate adaptive streaming of video data," in *IEEE International Conference on Multimedia Computing and Systems*, 1999, p. 67.

[2] N. Duffield, K. Ramakrishnan, and A. Reibman, "Save: An algorithm for smoothed adaptive video over explicit rate network," *IEEE/ACM Transactions on Networking*, December 1999.

[3] Z. Lei and N.D.Georganas, "Rate adaptation transcoding for video streaming over wireless channels," in *IEEE International Conference on Multimedia and Expo*, 2003.

[4] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for congestion controlled video playback over the Internet," in *Proceedings of ACM SIGCOMM*, 1999, pp. 189–200.

[5] E. Kusmierek and D. H. Du, "Streaming video delivery over Internet with adaptive End-to-End QoS," *Journal of Systems and Software, Special Issue on Adaptive Multimedia Computing*, July 2004.

[6] Z.-L. Zhang, Y. Wang, D. H. C. Du, and D. Shu, "Video staging: a proxy-server-based approach to end-to-end video delivery over wide-area networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 429–442, 2000.

[7] Z. Miao and A. Ortega, "Scalable proxy caching of video under storage constraints," *IEEE J. Selected Areas in Communications, Special issue on Internet Proxy Services*, vol. 20, no. 7, pp. 1315– 1327, September 2002.

[8] W. Ma and D. H. C. Du, "Reducing bandwidth requirement for delivering video over wide area networks with proxy server," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 539–550, December 2002.

[9] R. L. Carter and M. Crovella, "Server selection using dynamic path characterization in wide-area networks," in *INFOCOM*, no. 3, 1997, pp. 1014–1021.

[10] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, "A novel server selection technique for improving the response time of a replicated service," in *INFOCOM*, no. 2, 1998, pp. 783–791.

[11] Z. Fu and N. Venkatasubramanian, "Combined path and server selection in dynamic multimedia environments," in *ACM Multimedia (1)*, 1999, pp. 469–472.

[12] L. Amini and H. Schulzrinne, "On probe selection for dynamic multimedia selection," in *IEEE International Conference on Multimedia and Expo*, vol. 1, 2003, pp. 393 – 396.

[13] S. Viswanathan and T. Imielinski, "Metropolitan area video-on-demand service using pyramid broadcasting," *Multimedia Systems*, vol. 4(4), pp. 197–208, 1996.

[14] L. Juhn and L. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 44, no. 3, Sep 1997.

[15] K. A. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," in *Proceedings of ACM SIGCOMM*, 1997, pp. 89–100.

[16] J. Paris and D. Long, "A low bandwidth broadcasting protocol for video on demand," in *Proceedings of IEEE Int'l Conference on Computer Communications and Networks*, 1998.

[17] D. L. Eager and M. K. Vernon, "Dynamic skyscraper broadcasts for video-on-demand," *Lecture Notes in Computer Science*, vol. 1508, 1998.

[18] L. Gao, J. Kurose, and D. Towsley, "Efficient schemes for broadcasting popular videos," in *NOSSDAV*, 1998.

[19] L. Juhn and L. Tseng, "Fast data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 44, no. 1, Mar 1998.

[20] A. Hu, "Video-on-demand broadcasting protocols: A comprehensive study," in *Proceedings of IEEE Infocom*, vol. 1, April 2001.

APPENDIX I
PERIODIC BROADCAST SCHEMES SUMMARY

PB scheme	parameter	segmentation	transmission rate	# of seg. received simultan.
Pyramid	$\alpha > 1$	$S_i = \begin{cases} S\frac{\alpha-1}{\alpha^n-1} & \text{if } i=1 \\ \alpha S_{i-1} & \text{else} \end{cases}$	$r_t > r_p$	≤ 2
Skyscraper		$S_i = \begin{cases} 1 & \text{if } i=1 \\ 2 & \text{if } i=2,3 \\ 2S_{i-1}+1 & \text{if } i>3,\, i \bmod 4 = 0 \\ 2S_{i-1}+2 & \text{if } i>3,\, i \bmod 4 = 2 \\ S_{i-1} & \text{if } i>3,\, i \bmod 2 = 1 \end{cases}$	$r_t = r_p$	≤ 2
GDB	$k > 3$	$S_i = \begin{cases} 2^{i-1} & \text{if } i \leq k \\ \lfloor \sum_{j=i-k+1}^{i-1} \frac{S_j}{S_{i-k+1}} \rfloor S_{i-k+1} & \text{else} \end{cases}$	$r_t = r_p$	$k-1$
Polyharmonic	$m \geq 1$	$S_i = \frac{S}{n}$	$r_t = \frac{r_p}{m+i-1}$	n
Server Optimal	w (delay)	$S_i = w\alpha(1-\alpha)^{i-1}$	$r_t = \left(\sqrt[n]{\frac{S}{w}+1}-1\right) r_p$	n
Fibonacci		$S_i = \begin{cases} 1 & \text{if } i=1,2 \\ S_{i-2}+S_{i-1} & \text{else} \end{cases}$	$r_t = r_p$	2

n denotes number of segments and S length of a video.

APPENDIX II
SINGLE SEGMENT ANALYSIS FOR DELAYED PLAYBACK PB

In the Figures illustrating each of the case the dark gray area represents data received from the source server, medium gray area data received from the destination server and light gray area data missed during regular reception interval.

Case 1): $\mathbf{t_1 < t_c < t_2}$

The offset of the missed portion is: $f = \max(t_c - t_1, t_e - t_2)$.

1) If $t_e - t_2 \geq t_c - t_1$: the missed portion is transmitted at $t_2 + t_e - t_2 = t_e$. The transmission ends at $t_e + \frac{S_m}{\alpha r_p}$ and the playback ends at $t_p + \alpha(t_e - t_2) + \frac{S_m}{r_p}$. Therefore, the delay required is:

$$d_m = \frac{S_m}{\alpha r_p}(1-\alpha) - \alpha(t_e - t_2) \leq \Delta(1-\alpha)$$

2) If $t_c - t_1 > t_e - t_2$: the missed portion is transmitted at $t_2 + t_c - t_1 \geq t_2 + (t_e - t_2) = t_e$. The transmission ends at $t_2 + (t_c - t_1) + \frac{S_m}{\alpha r_p}$ and the playback ends at $t_p + \alpha(t_c - t_1) + \frac{S_m}{r_p}$.

The size of the missed portion is given by $\frac{S_m}{\alpha r_p} = \min(t_2 - t_1, t_e - t_c)$. Since $t_c - t_1 > t_e - t_2$, then $t_c - t_1 + t_2 - t_c \geq t_e - t_2 + t_2 - t_c$, which means that $t_2 - t_1 > t_e - t - c$ and $\frac{S_m}{\alpha r_p} = t_e - t_c$.

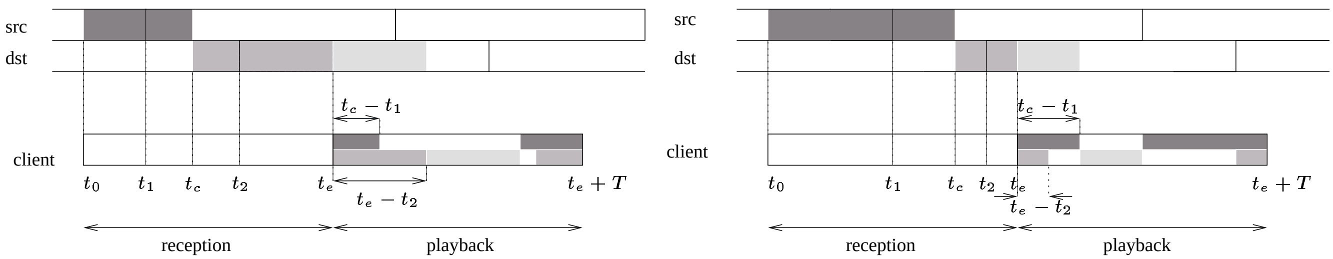
Let $t_c - t_1 = x$, $t_2 - t_c = y$ and $t_e - t_2 = z$.

a) for $t_2 - t_1 \leq \Delta$: the delay required is equal to:

$$\begin{aligned} d_m &= t_2 + t_c - t_1 - t_p - \alpha(t_c - t_1) + (t_e - t_c)(1-\alpha) \\ &= (x+y) - (y+z) - \alpha x + (y+z)(1-\alpha) \\ &= (x+y)(1-\alpha) - \alpha z \\ &\leq \Delta(1-\alpha) \end{aligned} \tag{10}$$

b) for $T - \Delta \leq t_2 - t_1 \leq T$: the delay required is:

$$\begin{aligned} d_m &= (x+y)(1-\alpha) - \alpha z \\ &= T(1-\alpha) - \alpha z \\ &\leq T(1-\alpha) \end{aligned}$$



(a) $t_e - t_2 \geq t_c - t_1$

(b) $t_e - t_2 < t_c - t_1$

Fig. 17. Case 1): $t_1 < t_c < t_2$

Case 2): $\mathbf{t_c < t_1 < t_2}$

The offset of the missed portion is $(t_e - t_2)$. The transmission of the missed portion starts at $t_2 + t_e - t_2 = t_e$. End of transmission of the missed portion is at $t_2 + (t_e - t_2) + \frac{S_m}{\alpha r_p}$. The end of playback is at time: $t_p + \alpha(t_e - t_2) + \frac{S_m}{r_p}$. The resulting delay is:

$$d_m = \frac{S_m}{\alpha r_p}(1-\alpha) - \alpha(t_e - t_2) \leq \Delta(1-\alpha)$$

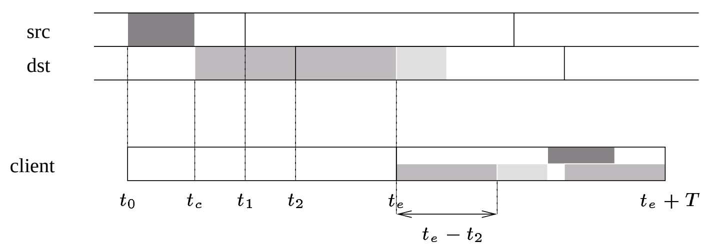


Fig. 18. Case 2): $t_c < t_1 < t_2$

Case 3): $\mathbf{t_1 < t_2 < t_c}$

The offset of the missing part is $\max(t_c - t_1, t_e - t_2)$.

1) If $t_e - t_2 \geq t_c - t_1$*:* the transmission of the missed part starts at $t_2 + t_e - t_2 = t_e$ and ends at $t_e + \frac{S_m}{\alpha r_p}$. The end of playback is at time: $t_p + \alpha(t_e - t_2) + \frac{S_m}{r_p}$. The resulting delay is:

$$d_m = \frac{S_m}{r_p}(1-\alpha) - \alpha(t_2 - t_2) \leq \Delta(1-\alpha)$$

2) If $t_c - t_1 > t_e - t_2$*:* the transmission of the missed part starts at: $t_2 + t_c - t_1 > t_2 + t_e - t_2 = t_e$ and ends at $t_2 + t_c - t - 1 + \frac{S_m}{\alpha r_p}$. The playback of the missed portion ends at $t_p + \alpha(t_c - t_1) + \frac{S_m}{r_p}$.

Similarly to case 1) the size of the missed portion is given by $\frac{S_m}{\alpha r_p} = t_e - t_c$.

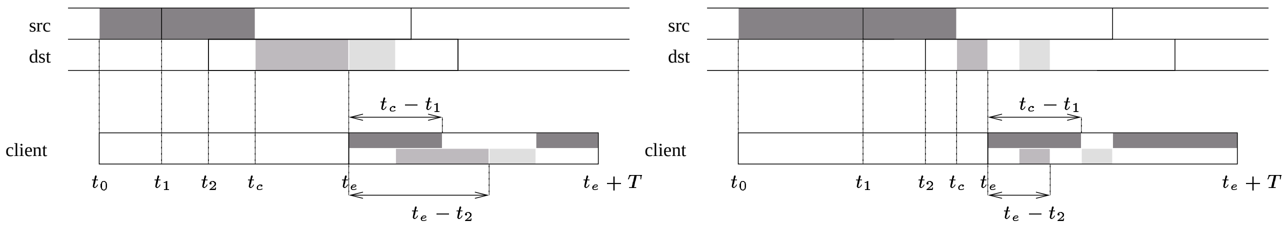
Let $t_2 - t_1 = x$, $t_c - t_2 = y$ and $t_e - t_c = z$.

a) for $t_2 - t_1 \leq \Delta$*:* the size of the missed portion is given by $\frac{S_m}{\alpha r_p} = T - (tc_t - 1) - (t_1 - t_0)$ and the delay required is:

$$\begin{aligned} d_m &= t_2 + t_c - t_1 - t_p - \alpha(t_c - t_1) + (t_e - t_c)(1-\alpha) \\ &= x - z - \alpha(x+y) + z(1-\alpha) \\ &= x(1-\alpha) - \alpha y - \alpha z \\ &\leq \Delta(1-\alpha) \end{aligned}$$

b) for $T - \Delta \leq t_2 - t_1 \leq T$*:* the delay required is:

$$\begin{aligned} d_m &= x(1-\alpha) - \alpha y - \alpha z \\ &\leq T(1-\alpha) \end{aligned}$$



(a) $t_e - t_2 \geq t_c - t_1$ (b) $t_c - t_1 > t_e - t_2$

Fig. 19. Case 3): $t_1 < t_2 < t_c$

Case 4)*:* $\mathbf{t_c < t_2 < t_1}$

The offset of the missed portion is $\max(t_e - t_2, t_e - t_1 + t_c - t_0)$.

1) If $t_e - t_2 \geq t_e - t_1 + t_c - t_0$: transmission of the missed part starts at $t_2 + t_e - t_2 = t_e$ and ends at $t_e + \frac{S_m}{\alpha r_p}$. The end of playback is at time: $t_p + \alpha(t_e - t_2) + \frac{S_m}{r_p}$. Therefore, the delay is:

$$d_m \leq \Delta(1 - \alpha)$$

2) If $t_e - t_1 + t_c - t_0 > t_e - t_2$: transmission of the missed part starts at $t_2 + t_e - t_1 + t_c - t_0 > t_2 t_e - t_2 = t_e$ and ends at $t_2 + t_e - t_1 + t_c - t_0 + \frac{S_m}{r_p}$. The playback ends at $t_p + \alpha(t_e - t_1 + t_c - t_0) + \frac{S_m}{\alpha r_p}$.

The size of the missed portion is given by $\frac{S_m}{\alpha r_p} = T - (t_e - t_1 + t_c - t_0) - (t_2 - t_c)$.

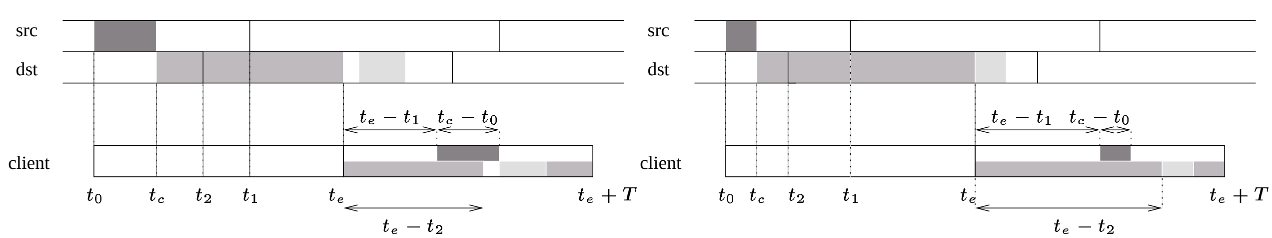
Let $t_2 - t_c = x$, $t_1 - t_2 = y$ and $t_e - t_1 = z$.

a) for $t_2 - t_1 \leq \Delta$: the delay is

$$\begin{aligned} d_m &= t_2 + t_e - t_1 + t_c - t_0 - t_p - \alpha(t_e - t_1 + t_c - t_0) + \frac{S_m}{\alpha r_p}(1 - \alpha) \\ &= T - y - (x + y + z) - \alpha(T - x - y) + (T - (T - x - y) - x) \\ &= T(1 - \alpha) - x(1 - \alpha) - y - z \\ &\leq T(1 - \alpha) \end{aligned}$$

b) for $T - \Delta \leq t_1 - t_2 \leq T$:

$$\begin{aligned} d_m &= T(1 - \alpha) - x(1 - \alpha) - y - z \\ &\leq T(1 - \alpha) - (T - \Delta) \\ &= \Delta - \alpha T \end{aligned}$$



(a) $t_e - t_2 \geq t_e - t_1 + t_c - t_0$ (b) $t_e - t_1 + t_c - t_0 > t_e - t_2$

Fig. 20. Case 4): $t_c < t_2 < t_1$

Case 5)*:* $\mathbf{t_2 < t_1 < t_c}$

The offset of the missed portion is: $t_c - t_1$. Note that $t_2 + t_c - t_1 < t_e$, thus, transmission of the missed part starts at $T + t_2 - t - c - t_1$ and ends at $T + t_2 + t_c - t_1 + \frac{S_m}{\alpha r_p}$. The playback ends at time: $t_p + \alpha(t_c - t_1) + \frac{S_m}{r_p}$.

Let $t_1 - t_2 = x$, $t_c - t_1 = y$ and $t_e - t_c = z$.

1) for $t_1 - t_2 \leq \Delta$: the size of the missed portion is given by $\frac{S_m}{\alpha r_p} = \min(t_1 - t_2, t_e - t_c)$ and the delay required is

$$\begin{aligned} d_m &= T + t_2 + t_c - t_1 - t_p - \alpha(t_c - t_1) + \frac{S_m}{\alpha r_p}(1-\alpha) \\ &= T - x - z - \alpha y + x(1-\alpha) \\ &= T - z - \alpha y \\ &\leq T \end{aligned} \tag{11}$$

if $\frac{S_m}{\alpha r_p} = t_1 - t_2 = x$. The same bound is obtained for $\frac{S_m}{\alpha r_p} = t_e - t_c = z$.
2) for $T - \Delta \leq t_1 - t_2 \leq T$: the size of the missed portion is given by $\frac{S_m}{\alpha r_p} = t_e - t_c$ and the delay required is

$$\begin{aligned} d_m &= T - x - z - \alpha y + z(1-\alpha) \\ &= T - x - \alpha y - \alpha z \\ &\leq T - (T - \Delta) \\ &= \Delta \end{aligned}$$

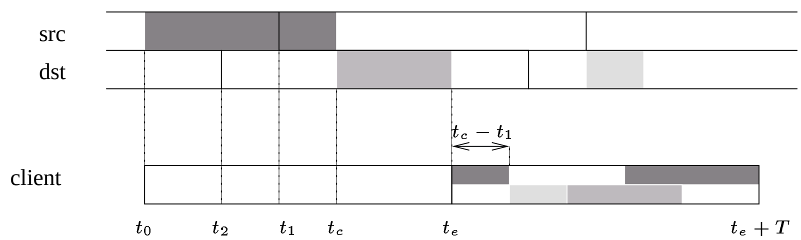


Fig. 21. Case 5): $t_2 < t_1 < t_2$

Case 6): $\mathbf{t_2 < t_c < t_1}$

The offset of the missed portion is: 0. Since $t_2 < t_e$ the transmission of the missed part starts at $T + t_2$ and ends at $T + t_2 + \frac{S_m}{\alpha r_p}$. The playback ends at $t_p + \frac{S_m}{r_p}$. The size of the missed portion is given by $\frac{S_m}{\alpha r_p} = \min(t_c - t_2, t_e - t_1)$. Let $t_c - t_2 = x$, $t_1 - t_c = y$ and $t_e - t_1 = z$.
1) If $t_1 - t_2 \leq \Delta$: the delay required is:

$$\begin{aligned} d_m &= T - (t_p - t_2) + \frac{S_m}{\alpha r_p}(1-\alpha) \\ &= T - (x + y + z) + x(1-\alpha) \\ &= T - y - z - \alpha x \\ &\leq T \end{aligned} \tag{12}$$

provided that $\frac{S_m}{\alpha r_p} = x$ (same result is obtained for $\frac{S_m}{\alpha r_p} = z$).
2) for $t_1 - t_2 \geq T - \Delta$: the delay required is:

$$\begin{aligned} d_m &= T - x - y - \alpha z \\ &= T - (x + y) - \alpha z \\ &\leq \Delta \end{aligned}$$

provided that $\frac{S_m}{\alpha r_p} = z$ (same result is obtained for $\frac{S_m}{\alpha r_p} = x$).

APPENDIX III
MISSING FRAMES TRANSMISSION TIMES FOR DYNAMIC CACHING

We show that for all cases of delayed playback that required long delay (proportional to the playback transmission time) to avoid playback disruption, the frames that the client does not received during regualr reception are transmitted by the destination server just prior to the server switch at t_c. We compute the latest time before t_c of the transmission and show that all missing frames are transmitted during time interval $(t_c - \Delta, t_c)$. Let $\hat{t_2} = t_2 - T$ denote the previous time the destintion server transmits beginning of the segment before t_2.

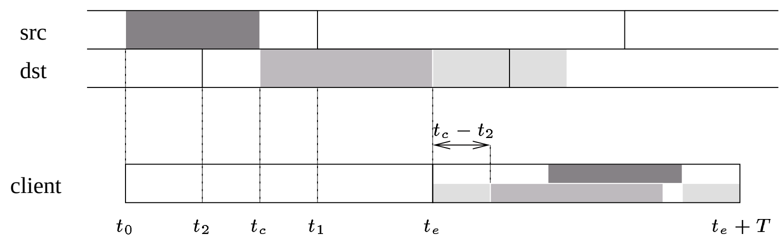


Fig. 22. Case 6): $t_2 < t_c < t_1$

Case 1): $t_1 < t_c < t_2$ and $t_2 - t_1 \geq T - \Delta$

The delay is bounded by $T(1-\alpha)$ when $t_c - t_1 > t_e - t_2$. Then the missed part is transmitted at time $\hat{t_2} + t_c - t_1 = t_c - (t_1 - \hat{t_2}) \geq t_c - \Delta$.

Case 3): $t_1 < t_2 < t_c$ and $t_2 - t_1 \geq T - \Delta$

The delay is bounded by $T(1-\alpha)$ when $t_c - t_1 > t_e - t_2$. Then the missed part is transmitted at time $\hat{t_2} + t_c - t_1 \geq t_c - \Delta$.

Case 4): $t_c < t_2 < t_1$ and $t_1 - t_2 \leq \Delta$

The delay is bounded by $T(1-\alpha)$ when $t_c - t_0 + t_e - t_1 > t_e - t_2$. The missed portion is transmitted at $\hat{t_2} + t_c - t_0 + t_e - t_1 = t_c + t_2 - t_1 \leq t_c + \Delta$.

Case 5): $t_2 < t_1 < t_c$ and $t_1 - t_2 \leq \Delta$

The missed portion is transmitted at $t_2 + t_c - t_1 \leq t_c - \Delta$.

Case 6): $t_2 < t_c < t_1$ and $t_1 - t_2 \leq \Delta$

The missed portion in transmitted at t_2. Note that $t_c - t_2 \leq t_1 - t_2 \leq \Delta$.

Appendix IV
Static Caching Probability Derivation

We assume that the source server can transmit beginning of the segment any time during the segment reception, i.e., let t_1 be a random variable uniformly distributed over interval $(0, T)$. We assume that $(t_2 - t_1)$ is a random variable uniformly distributed over an interval of length $\min(t_e - t_1, \Delta) + \min(t_1 - t_0, \Delta) \leq 2\Delta$. For simplicity we assume also that server transmission offset is equal to Δ. Note that if $t_1 \leq \Delta$ probability that $t_2 \leq t_1$ is equal to $\frac{t_1}{t_1+\Delta}$. Similarly if $t_1 \geq T - \Delta$, then $P(t_2 \leq t_1) = \frac{\Delta}{T - t_1 + \Delta}$. Hence, the overall probability is equal to:

$$P(t_2 \leq t_1) = \int_0^{\Delta} \frac{x}{x+\Delta}\frac{1}{T}dx + \int_{\Delta}^{T-\Delta} \frac{1}{2}\frac{1}{T}dx + \int_{T-\Delta}^{T} \frac{\Delta}{T-x+\Delta}\frac{1}{T}dx \tag{13}$$

The probability that server switch occurs after t_1 is equal to $P(t_1 \leq t_c) = 1 - P(t_c < t-1) = 1 - \frac{t_1}{T}$. Consquently $P(t_1 + y \leq t_c) = 1 - \frac{t_1+y}{T}$. Then we have:

$$P(t_1 + y \leq t_c) = \int_0^{T-y} (1 - \frac{x+y}{T})\frac{1}{T}dx + \int_{T-y}^{T} 0\frac{1}{T}dx = \int_0^{T-y} (1 - \frac{x+y}{T})\frac{1}{T}dx \tag{14}$$

Now the combined probability for $y \leq \Delta$ $(T - \Delta \leq T - y)$:

$$P(t_2 \leq t_1 \leq t_c) = \int_0^{\Delta} \frac{x}{x+\Delta}\left(1 - \frac{x+y}{T}\right)\frac{1}{T}dx + \int_{\Delta}^{T-\Delta} \frac{1}{2}\left(1 - \frac{x+y}{T}\right)\frac{1}{T}dx + \int_{T-\Delta}^{T-y} \frac{\Delta}{T-x+\Delta}\left(1 - \frac{x+y}{T}\right)\frac{1}{T}dx \tag{15}$$

Simlarly we can derive probability for $\Delta < y \leq T - y$ and $\Delta \leq T - y \leq y$. All three case are capture by the following equation:

$$\begin{aligned} P(t_2 \leq t_1 \leq t_c - y) \quad = \quad & \int_0^{\min(\Delta, T-y)} \frac{x}{x+\Delta}\left(1 - \frac{x+y}{T}\right)\frac{1}{T}dx + \int_{\min(\Delta,T-y)}^{\min(T-\Delta,T-y)} \frac{1}{2}\left(1 - \frac{x+y}{T}\right)\frac{1}{T}dx \\ & + \int_{\min(T-\Delta,T-y)}^{T-y} \frac{\Delta}{T-x+\Delta}\left(1 - \frac{x+y}{T}\right)\frac{1}{T}dx \end{aligned} \tag{16}$$