Providing VCR Capabilities in Large-Scale Video Servers

Jayanta K. Dey-Sircar, James D. Salehi,
James F. Kurose and Don Towsley

CMPSCI Technical Report 94-53

July 5, 1994

Providing VCR Capabilities in Large-Scale Video Servers *

Jayanta K. Dey-Sircar James D. Salehi James F. Kurose Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

July 6, 1994

Abstract

Providing smooth playback capabilities for video servers, which must support potentially thousands of on-demand users, has been an area of active research. From a user's perspective, VCR functions of fast-forward and rewind (FF/Rew), are desirable features in video-on-demand. But FF/Rew at n times the regular playback rate requires n times the regular playback bandwidth from architectural components of the video server. Thus, guaranteeing sufficient bandwidth to enable users to perform FF/Rew reduces the number of supportable users by a factor of n. In this paper we propose an alternative, *effective FF/Rew service*, which provides FF/Rew capabilities with an associated statistical quality-of-service (QoS) guarantee. This service provides immediate access to full-resolution FF/Rew bandwidth with high probability. When bandwidth is not available, service is either delayed or provided immediately but with a loss in resolution. In addition, we specify several QoS metrics to characterize the delay or loss experienced by a FF/Rew request. We show that using effective FF/Rew with statistical guarantees on these QoS metrics results in a significant increase in the number of supportable users, when compared to systems in which FF/Rew bandwidth is statically reserved for each user. Moreover, a playback-only video server can be extended to provide FF/Rew service by reserving only a small portion of its total bandwidth, which is dynamically shared among FF/Rew requests.

Key Words: movie-on-demand, multimedia, QoS metrics, statistical guarantees, VCR functions, video-on-demand, video servers.

---

*This work was supported by the National Science Foundation under grant NCR-9116183 and by ARPA under ESD/AVS contract number F19628-92-C-0089. The authors can be contacted at [dey, salehi, kurose, towsley]@cs.umass.edu.

1 Introduction

Supporting retrieval of stored video from a video server for playback at a client, across a high speed network, has recently been the focus of several research and commercial endeavors [5, 9, 12, 15, 16, 21]. Video servers, capable of providing video-on-demand, have engendered a broad range of application areas, including digital libraries, archives and information services, on-demand television, on-demand movies, multimedia publishing and distance education. It is of no surprise, therefore, that video servers are receiving significant attention from the computer, telecommunication and entertainment industries.

The basic service provided by a video server is the smooth playback of video streams. Research on video-on-demand has concentrated on resource reservation mechanisms in the video server which provide "hard" (i.e. strict or absolute) quality of service (QoS) guarantees on bandwidth, delay, and jitter for playback. This has led to two approaches for allocating I/O, memory system and CPU bandwidth to a stream requiring variable-rate video. In the first approach, a video stream is allocated a fixed bandwidth, often according to its peak bandwidth requirements [1, 9, 12, 15, 21]. We refer to this as the *peak bandwidth allocation* method. The motivation behind this approach is that it leads to simple round-robin or rate-monotonic scheduling in the I/O subsystem or the CPU of the video server. The disadvantage of this method is that when stream rates vary as a function of time, video server bandwidth is wasted. To ameliorate this situation, the second technique employed has been to adapt the bandwidth allocation to the stream's dynamic rate requirements [2, 7, 18]. The bandwidth requirement of a stored video stream, known *a priori*, is used to construct a bandwidth allocation schedule. We call this the *planned bandwidth allocation* method. This allows resource reservation at a finer granularity than the peak allocation method, potentially admitting more users at a given QoS.

Previous research has concentrated on *playback-only* systems. However, in addition to the regular playback mechanism, a desirable feature in all of these video-on-demand applications is the capability of viewing a video stream at high speed, i.e. the VCR fast-forward and rewind (FF/Rew) mechanism. In digital archives, a user would like to be able to browse quickly through a video clip. In distance learning, a student can rewind a segment to revisit the important sections of a lecture. For viewing movies on-demand, VCR functionality is almost a requirement in order to be a viable alternative to video rentals. Researchers have not examined the mechanisms needed to support FF/Rew functionality in the design of large-scale video servers.

The main difficulty in handling FF/Rew is that it introduces a wide variability in the bandwidth requirement of a single video stream. We denote by $n \times$FF/Rew the accessing of a video sequence at n times the normal playback rate (30 frames/sec). $n \times$FF/Rew requires several architectural components of the video server to sustain n times the playback bandwidth. For example, the stream's video data may require storage-media retrieval at n times the playback rate. This is regardless of whether the video server transmits all the frames. For predictive-encoding algorithms with inter-frame dependencies, the bandwidth requirements of the network can increase by almost a factor of n as well. In general, in order for frames to be decoded at the receiver, the frames they depend on must also be received and decoded. Several current implementations of MPEG-decoders require all frames to decode correctly [6, 14].

In order to accommodate FF/Rew capabilities, one may take the same approach as in playback-only video servers, and provide users with hard QoS guarantees for bandwidth, delay, and jitter, by reserving FF/Rew bandwidth for each stream. We call this approach the *guaranteed FF/Rew* approach, since it provides hard guarantees for FF/Rew requests. There are two drawbacks to this approach. First, it is inefficient when used with the bandwidth allocation techniques mentioned above. Providing peak bandwidth allocation when admitting users into a video server which provides $n \times$FF/Rew capabilities reduces the number of supportable video streams by a factor of n. The assumption behind the planned bandwidth allocation technique, that the retrieval pattern is known ahead of time, conflicts with the inherently nondeterministic nature of users' FF/Rew behavior. Second, in VCRs there is a short delay in the initiation of a FF/Rew request, and typically the image resolution is reduced during FF/Rew. This suggests that users may be willing to tolerate a small latency or a small loss in resolution when requesting FF/Rew from a video server.

In this paper, we explore an alternative, called the *effective FF/Rew* service mechanism. The adjective "effective" indicates that instead of providing hard guarantees, this service comes with an associated statistical guarantee on a QoS metric. Our goals are twofold: to admit a high number of users into the server, while providing VCR-capability to users with minimized perceived difference from guaranteed FF/Rew service.

The primary results of this paper are twofold. First, we show that the number of users admitted with effective FF/Rew service is very close to the number admitted by the playback-only service. Second, we define several QoS metrics to characterize effective FF/Rew, and apply these metrics to evaluate the effective FF/Rew service mechanism. We show that, as an alternative to providing hard guarantees, giving statistical guarantees on the QoS metrics of FF/Rew yields a large increase in the number of users that can be admitted into the system at a "reasonable" level of service. A combination of analysis and simulation has been used to evaluate the performance of different mechanisms for supporting effective FF/Rew service.

The rest of the paper is organized as follows. Section 2 describes the effective FF/Rew service. Section 3 discusses the QoS metrics used to measure this service. Section 4 contains models of user behavior and of the video server during video retrieval. Section 5 describes the results, and section 6 concludes the paper.

2 Mechanisms supporting effective FF/Rew

The motivations behind effective FF/Rew derive from two simple observations: *i)* users spend most of their time in playback mode while watching a video sequence; and *ii)* the duration of time spent in FF/Rew mode is relatively small. Providing fixed resource bandwidth reservation for such infrequent requests, as in the guaranteed FF/Rew mechanism, thus leads to an immense underutilization of resource bandwidth.

We propose the following mechanism as the basis for providing effective FF/Rew service.

- Since playback is the primary functionality in video-on-demand, we require that every user be guaranteed the required playback bandwidth.

- In addition, a small fraction of the server bandwidth is reserved for FF/Rew, and is used to serve all of the FF/Rew requests, as discussed below.

We propose and evaluate two schemes for sharing the reserved FF/Rew bandwidth among FF/Rew requests.

- Under the first scheme, if the bandwidth required for a FF/Rew request is unavailable (i.e. cannot be allocated), the request is delayed until the bandwidth becomes available. This scheme is referred as the Delay Scheme (DS) in the rest of the paper.

  Within DS, we consider two service policies. The first is that a user holds its playback bandwidth while waiting for FF/Rew to be initiated, and continues to receive video at the normal rate. This is referred to as the Hold Bandwidth Delay Scheme (HD). The advantage of this policy is that the user continues to see motion while awaiting a FF/Rew response from the server. In the second policy, when a FF/Rew request is delayed, the server stops sending video data to that user, and that user's playback bandwidth becomes available to serve FF/Rew requests for streams which already await FF/Rew service. We call this the Release Bandwidth Delay Scheme (RD), since users release their playback bandwidth at the time they make a FF/Rew request. The potential advantage of RD over HD is reduction in FF/Rew waiting times, because when users wait, the FF/Rew bandwidth available in RD exceeds that available in HD. We assume that FF/Rew requests are satisfied in FIFO order.

- Under the second scheme, when FF/Rew bandwidth is unavailable, the bandwidth of each FF/Rew stream is lowered to accommodate the new FF/Rew request. Each FF/Rew stream thus suffers a loss of resolution in the video it receives. We refer to this as the Loss Scheme (LS). Under LS, users never wait for initiation of a FF/Rew request.

  Several techniques have addressed recovery at the client from intra-frame data loss. With these techniques at the client, the server can discard data in response to FF/Rew bandwidth contention. In a block-by-block frame compression algorithm like MPEG, losing a pixel is equivalent to losing a portion of the frame, and one approach is for the decoder to substitute the corresponding portion of the previous frame [3]. Alternatively, the decoder can reconstruct this portion by combining the corresponding portion from the previous frame and interpolations from the adjacent blocks of the current frame [22]. A second approach divides the discrete cosine transform (DCT) coefficients of a frame into layers so that in overload situations the lower layers can be discarded, resulting in an image of a lower quality [4, 10]. The third technique involves sending adjacent pixels in different packets, so that lost packets do not imply localized pixel loss, and each lost pixel usually has undamaged neighbors [19, 20].

3 QoS Metrics

This section describes the performance metrics we have used to evaluate the proposed schemes. One of our goals is to compute the maximum number of users that can be admitted into the system under various FF/Rew schemes for a given constraint on QoS. Table 1 summarizes these metrics. We describe these measures in more detail below.

In HD and RD, we define W to be the time that a random customer who selects fast forward or rewind must wait until the request is satisfied (i.e. fast-forwarding or rewinding actually begins). We want $E[W]$, the average "response time" of the system to a user's FF/Rew request, to be small. The delay distribution, $P[W > t]$, can be

Notation	*Definition*	*DS*	*LS*
$E[W]$	Expected waiting time for a FF/Rew request to be serviced	√	
$P[W > t]$	Probability a FF/Rew request waits for time greater than t	√	
$E[L]$	Expected fractional loss experienced by a FF/Rew request		√
$E[L_{max}]$	Expected maximum fractional loss experienced by a FF/Rew request		√
$P[L_{max} > f_l]$	Probability maximum fractional loss of a FF/Rew request exceeds f_l		√

Table 1: QoS Metrics for DS and LS as a function of number of users

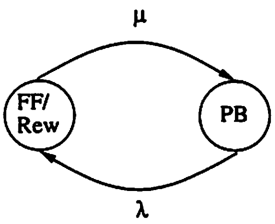
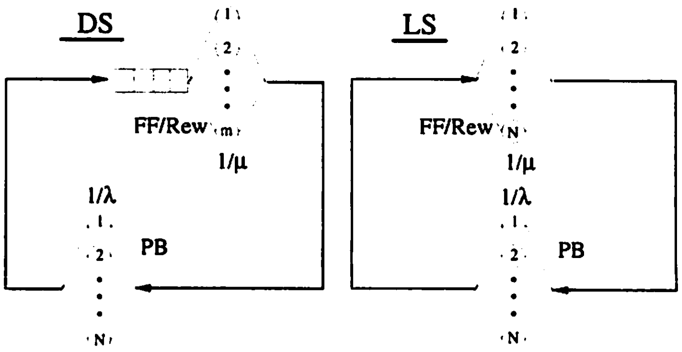


Figure 1: Model of a user watching a video



(a)

(b)

Figure 2: Video server models

used to evaluate the worse-case delay seen by the user. For example, given the delay distribution and a threshold t beyond which waiting is viewed as unacceptable by the user, we can limit the maximum number of users such that $P[W > t] < Q$, where Q is some small probability (e.g. 0.01 or 0.001).

In LS, we define L to be the fractional loss experienced by a FF/Rew request at a random point in time. The loss experienced by a FF/Rew request depends upon the number of concurrent FF/Rew requests, since all such requests must share the available bandwidth. Since the number of users simultaneously performing FF/Rew changes in a stepwise fashion over time, a FF/Rew request may experience a series of loss levels. As mentioned in [8], a user often measures the overall quality of a video sequence by its worst resolution. We thus define L_{max} to be the maximum fractional loss experienced by a FF/Rew request. L_{max} captures the user's perspective of worst-case loss, and thus, we want to size the system such that $E[L_{max}]$ is low. Alternatively, given the loss distribution and a threshold f_l, we can limit the maximum number of users such that $P[L_{max} > f_l] < Q$.

4 Model

In this section, we describe the model of a user watching a video sequence and the model of the video server accepting FF/Rew requests from a user.

4.1 Model of a user

The behavior of each user is modeled by a two-state Markov process with states PB (playback) and FF/Rew, as shown in Figure 1. The times spent by the user in playback and FF/Rew modes are exponentially distributed with means $1/\lambda$ and $1/\mu$ respectively. The bandwidths required by the user in playback and FF/Rew modes are Φ_P and Φ_F respectively. $\Phi_F/\Phi_P = n$ corresponds to $n\times$FF/Rew. We assume the bandwidth requirements of all users are identical.

4.2 Model of the video server system

A video server has multiple resources, e.g. I/O subsystem bandwidth, network access bandwidth, system bus bandwidth, memory bandwidth and CPU processing. Our model of this multiple-resource system is a simple one–we focus only on that resource of the system which most severely limits throughput. The bandwidth of this bottleneck component, B, determines the maximum number of users that can be admitted into the video server. Based on B, the maximum number of users that can be supported in a playback-only video server is $\lfloor B/\Phi_P \rfloor$. In a guaranteed FF/Rew video server, the maximum number of supportable users is $\lfloor B/\Phi_F \rfloor$.

A user requesting FF/Rew under HD requires an additional bandwidth of $(\Phi_F - \Phi_P)$. When N users are admitted into the server, $N \leq \lfloor B/\Phi_P \rfloor$[1], the available bandwidth for FF/Rew is $(B - N\Phi_P)$. Thus the number of FF/Rew requests under HD that can be supported simultaneously is given by:

$$m_{HD}(N) = \left\lfloor \frac{B - N\Phi_P}{\Phi_F - \Phi_P} \right\rfloor . \tag{1}$$

Under RD, the bandwidth of a FF/Rew request that has to wait is used to satisfy the earliest waiting FF/Rew request. With N users in the system, $N \leq \lfloor B/\Phi_P \rfloor$, of which N_F have requested FF/Rew service, $0 \leq N_F \leq N$, the number of FF/Rew requests that can be simultaneously supported is given by:

$$m_{RD}(N, N_F) = \left\lfloor \frac{B - (N - N_F)\Phi_P}{\Phi_F} \right\rfloor , 0 \leq N_F \leq N. \tag{2}$$

When all the users are in playback mode, there is enough bandwidth to support $m_{RD}(N, 0)$ FF/Rew requests. Similarly, $m_{RD}(N, N)$ FF/Rew requests can be concurrently supported when every user has requested FF/Rew service. $E[m_{RD}]$ denotes the expected number of FF/Rew servers with N users in the system.

For LS, $m_{LS}(N)$ is the number of FF/Rew requests that can be concurrently supported without any loss in resolution. Note that $m_{LS}(N) = m_{HD}(N)$. When N_F of N users are in FF/Rew mode, the bandwidth available for FF/Rew is $(B - (N - N_F)\Phi_P)$, while their bandwidth requirement is $N_F\Phi_F$. Hence the fractional loss in resolution is:

$$\begin{aligned} f_l(N, N_F) &= \max\left(0, 1 - \frac{B - (N - N_F)\Phi_P}{N_F\Phi_F}\right), \\ & \quad 0 \leq N_F \leq N. \end{aligned}$$

For the remainder of the paper, we simplify the notation to m_{HD}, m_{RD}, m_{LS} and f_l.

---

[1] Since *effective FF/Rew* assumes that playback bandwidth is assigned to every user, the number of users in the system cannot exceed $\lfloor B/\Phi_P \rfloor$.

We are interested in computing the maximum number of users that can be admitted, while maintaining a certain level of performance. Thus we model the video server as a closed queueing network with 2 queues and N customers, as shown in Figure 2. Figure 2(a) represents DS, and Figure 2(b) represents LS. The lower queue is the playback queue. A customer in the playback queue models a user receiving video at the normal playback rate. Given our user model, the service time in this lower queue is exponentially distributed with mean $1/\lambda$. The upper queue is the FF/Rew queue. A customer in this upper queue represents a user that has requested, and is either queued for, or receiving FF/Rew service. The time spent in this queue is exponentially distributed with mean $1/\mu$. Users cycle through these two queues. There is never any queueing in the lower queue because there is always enough playback bandwidth. However, the HD and RD mechanisms, depicted in Figure 2(a), require that a FF/Rew request wait in a queue if there is insufficient bandwidth. On the other hand, LS, as shown in 2(b), does not require FF/Rew requests to wait for service. When In Figure 2(a), $m =$ is either m_{HD} or m_{RD}. When Figure 2(a) represents HD (RD), $m = m_{HD}$ ($m = m_{RD}$).

The behavior of each of these two closed queueing networks is described by simple Birth-Death processes with state-dependent transition rates. The state variable of each Birth-Death process is the number of users in the FF/Rew queue.

Details of the Birth-Death processes for the HD, RD and LS systems are described in the Appendix. In the appendix, we also derive the QoS performance measures that can be computed analytically. The rest of the metrics are obtained using simulation.

5 Results

In this section, we examine and compare the behavior of HD, RD, and LS. Throughout the discussion, we focus our attention on the parameter values $B/\Phi_P = 1000$ (a system which can support 1000 users concurrently in playback mode), $\Phi_F/\Phi_P = 3$ (the user requests FF/Rew at 3 times normal playback rate), $1/\lambda = 1200$ s, and $1/\mu = 20$ s. These parameters were chosen to approximate the behavior of users watching movies delivered by a video server in which FF/Rew activity is relatively rare and of comparatively short duration. In addition, we have examined a range of parameter values with Φ_F/Φ_P varying from 2 to 6, $1/\mu$ from 10 s to 60 s, and $1/\lambda$ from 480 s to 1200 s, to capture other video-on-demand application environments. Similar behavior is seen across this range of parameter values. In the course of discussing each of the HD, RD and LS systems, we will summarize the trends exhibited, and compare the base system to a system in which users exhibit greater FF/Rew activities (the "4-60" system, in which $\Phi_F/\Phi_P = 4$ and $1/\mu = 60$ s), and a system in which users exhibit lower FF/Rew activities (the "2-10" system, in which $\Phi_F/\Phi_P = 2$ and $1/\mu = 10$ s).

Each data point derived by simulation resulted from a batch of 20 runs, with a run length of 200,000 FF/Rew requests. The 95% confidence interval half widths were below 5% for all data presented and discussed in this section.

Throughout this section, we denote by $N^P(C)$ the maximum number of supportable users under policy P (HD, RD, LS) constrained by QoS metric C (e.g. $E[W] < 3$, or $P[L_{max} > 0.3] < 0.1$).

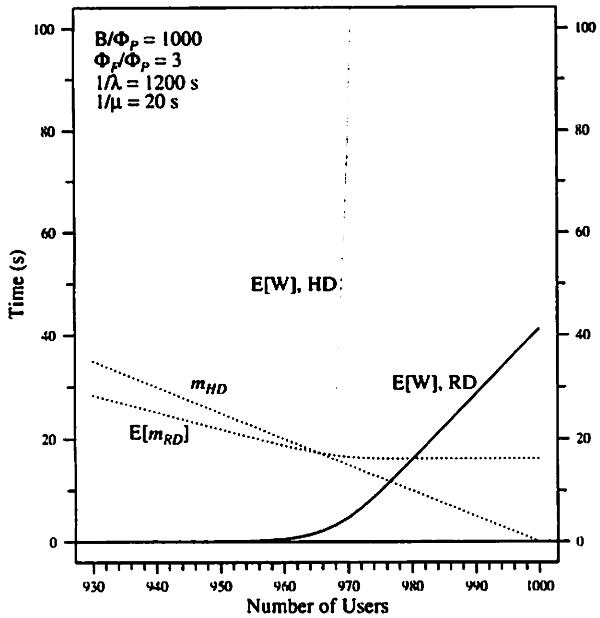


Figure 3: Waiting time of FF/Rew requests

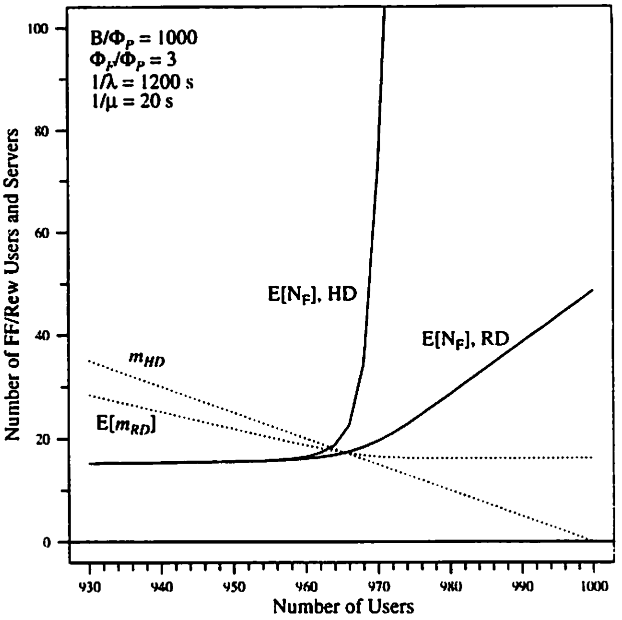


Figure 4: Number of FF/Rew users and servers

5.1 The Delay Scheme

Figures 3 and 4 show $E[W]$ and $E[N_F]$ respectively for HD and RD as a function of the number of admitted users N, along with m_{HD} and $E[m_{RD}]$ for the two systems. There are several notable aspects of Figures 3 and 4. The first is that, $E[W]$ becomes nonzero and starts increasing rapidly when $E[N_F]$ approaches the number of FF/Rew servers, for both HD and RD. The second is that for values of t as small as 1 second, both $N^{HD}(E[W] < t)$ and $N^{RD}(E[W] < t)$ exceed 950, in contrast to guaranteed FF/Rew service, which admits only $\lfloor B/\Phi_F \rfloor = 333$ users. Additionally, only 5% of the video-server bandwidth is required to provide effective FF/Rew service with this QoS guarantee, i.e. $N^{HD}(E[W] < t)$ and $N^{RD}(E[W] < t)$ are within 95% of the maximum number of users supported in a playback-only server. Figure 3 illustrates that RD performs uniformly better than HD, i.e. that for a particular QoS (based on $E[W]$), RD admits as many or more users as HD. This is a consequence of greater FF/Rew bandwidth under RD than HD when requests wait, as mentioned in Section 2.

In an overload condition, i.e. as N rises above the knee of the curves in Figures 3 and 4, HD and RD exhibit markedly different behavior. Beyond 960 users, $E[W]$ and $E[N_F]$ rise steeply under HD. Under RD, $E[W]$ and $E[N_F]$ increase linearly and thereby degrade more gracefully in response to the overload condition.

The explanation for this overload behavior can be discerned from the behavior of the m_{HD} and $E[m_{RD}]$. $E[m_{RD}]$ approaches a limit (in the Figure, about 16), and is insensitive to increases in N. After $E[m_{RD}]$ cross $E[N_F]$, i.e. in overload, additional users are simply queued in the FF/Rew system. Since these users have released their bandwidth, $E[m_{RD}]$ does not decrease further. In the limit, $E[W]$ grows linearly with slope $1/(\mu E[m_{RD}])$. (We can compute this limit by solving equation 2 for $E[m_{RD}]$ using $E[m_{RD}]\mu = \lambda(N - E[N_F])$. The second equation is given by Little's Law applied to the playback queue). However, m_{HD}, computed in equation 1, is a linearly decreasing

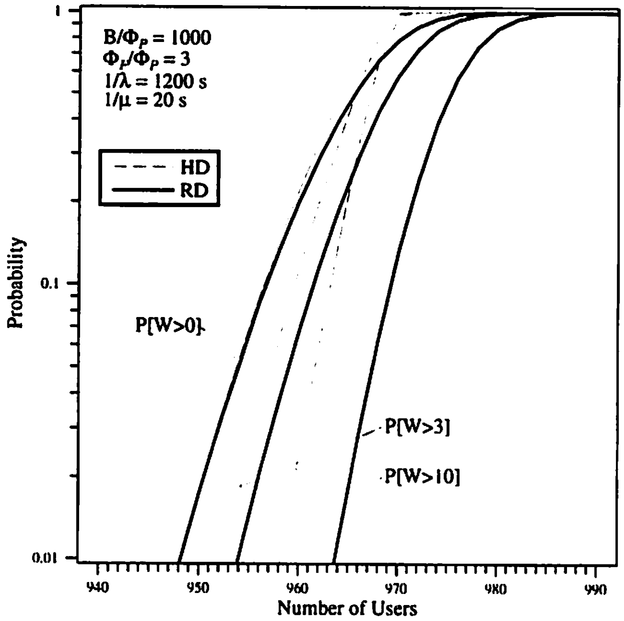


Figure 5: Probability FF/Rew delay exceeds t s

function of N with a slope of $-\Phi_P/(\Phi_F - \Phi_P)$. In Figure 3, where $\Phi_F = 3 \times \Phi_P$, the m_{HD} curve has a slope of -0.5. Therefore, the steepness of $E[W]$ curve as a function of N not only reflects the increase in users, but also a decrease in the number of available servers.

Let us now consider the issue of sizing the two systems with the goal of keeping $P[W > t]$, the probability that a FF/Rew request exceeds t seconds, below a QoS probability threshold Q. Figure 5 graphs $P[W > t]$ for $t = 0, 3$, and 10 seconds. Here, $t = 3, 10$ has been chosen to approximate thresholds beyond which users would characterize the service as intolerable. $t = 0$ is plotted for reference purposes. The y-axis of figure 5 is graphed in log scale in order to amplify the probabilities in the low range of 0.01 to 0.1, i.e. that the user receives service within time t with high probability.

There are several interesting aspects of Figure 5. The first is $N^{HD}(P[W > t] < 0.01)$ is slightly lower than $N^{HD}(E[W] < t)$. This also holds for RD. For example, when $t = 3$, $N^{RD}(E[W] < 3) = 968$, whereas $N^{RD}(P[W > 3] < 0.01) = 953$. However, note that if 968 users are admitted following the $E[W]$ metric under RD, $P[W > 3] \simeq 0.45$.

Figure 5 shows negligible difference between the $P[W > 0]$ curves under RD and HD when N is in the range $[948, 956]$. This is explained by noting from Figure 4 that $E[N_F]$ is less than the number of FF/Rew servers, and thus there is very little waiting for FF/Rew service in either of the systems. Similarly, there is little difference in the $P[W > 3]$ curves under HD and RD in the probability range $[0.01, 0.2]$. This trend holds for any parameter values where the server is sized for small waiting times.

However, as the waiting time and probability thresholds t and Q increase, there is a larger difference between

Metrics	*4-60*	*3-20*	*2-10*
$N^{RD}(P[W > 3] \leq 0.01)$	839	953	987
$E[W]$ slope in overload	1.45	1.24	1.11

Table 2: Comparison between RD systems 4-60, 3-20 and 2-10

$N^{RD}(P[W > t] < Q)$ and $N^{HD}(P[W > t] < Q)$. These trends hold not only because an increase in t or Q results in increased waiting and therefore increased differences in performance of the two systems, but also because RD is more stable than HD in an overload situation.

The system 4-60 (in which $\Phi_F/\Phi_P = 4$ and $1/\mu = 60$ s) and 2-10 (in which $\Phi_F/\Phi_P = 2$ and $1/\mu = 10$ s) exhibit the same general behavior, and the relative comparisons between RD and HD noted above are valid for them as well. In Table 2, we give two measures for comparison among the three sets of parameter values. The measures are $N^{RD}(P[W > 3] \leq 0.01)$ and the slope of $E[W]$ (roughly, an overload protection measure) under RD. The trends in both measures are consistent with the fact that 4-60 is more congested than 3-20, which is more congested than 2-10.

In conclusion, HD and RD support nearly the same number of users under the QoS constraints we consider. Both are vastly superior to guaranteed FF/Rew service in this respect, and only require a small bandwidth reservation (less than 1% for 2-10, 2.5% for 3-20, and only 16% in the 4×FF/Rew, 60-second duration FF/Rew extreme.) When the video server experiences an overload situation, degradation of FF/Rew service under RD is much more graceful than under HD.

5.2 The Loss Scheme

Figure 6 graphs the expected maximum loss fraction $E[L_{max}]$, as well as the expected loss fraction $E[L]$, as a function of N. The figure also plots the number of full-resolution FF/Rew servers m_{LS}, the expected number of FF/Rew users $E[N_F]$, and the maximum possible loss in the system when all N users are in FF/Rew mode.

Because there is no queueing under LS, $E[N_F]$ increases slowly as a function of N. The maximum loss fraction possible in the system is also a slowly increasing function of N. The number of full-resolution FF/Rew servers, defined in equation 1, is a linearly decreasing function of N with slope $-\Phi_P/(\Phi_F - \Phi_P)$, or -0.5 for Figure 6.

The graph indicates that $N^{LS}(E[L_{max}] < 0.1) \simeq 966$. This again is a large improvement over the 333 users admitted under guaranteed FF/Rew service, and requires the reservation of only 4% of the total playback bandwidth. This graph also indicates that users begin experiencing loss when m_{LS} is within approximately 50% of $E[N_F]$, i.e. around $N = 945$. A FF/Rew request experiences several loss levels over its duration. Thus, when N exceeds 950, $E[L_{max}]$ is higher than $E[L]$, because users are experiencing multiple loss levels. However, when N is below 950, $E[L_{max}]$ and $E[L]$ are close because users experience few loss levels. With 1000 users, there is no extra FF/Rew bandwidth, and each user has only its playback bandwidth for FF/Rew. Thus, both curves meet the maximum loss fraction curve at 0.67 (because $\Phi_P/\Phi_F = 0.33$) for $N = 1000$.

Figure 7 graphs the tail of the L_{max} probability distribution as a function of N. We consider loss fraction

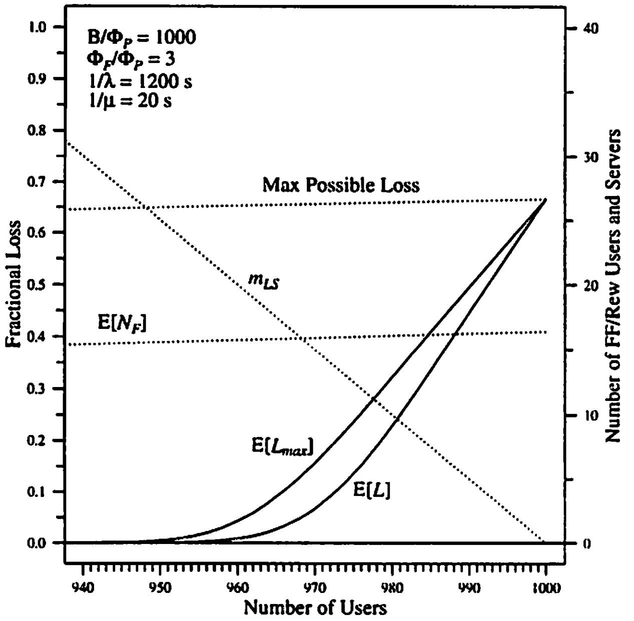


Figure 6: Averages in the Loss System

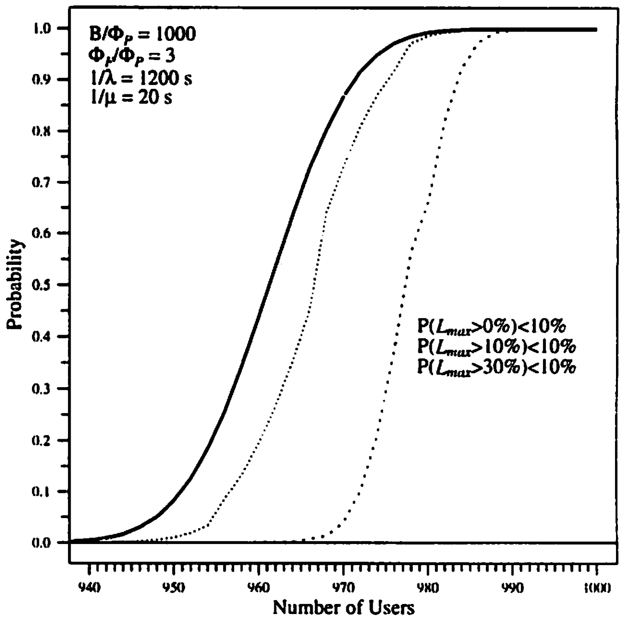


Figure 7: Probability that L_{max} exceeds f_l

Metrics	*4-60*	*3-20*	*2-10*
$N^{LS}(P[L > 0.1] < 0.1)$	854	956	987
ΔN^{LS}	53	20	6

Table 3: Comparison of Loss Systems 4-60, 3-20 and 2-10

thresholds f_l of 0, 0.1 and 0.3 in the figure. Note that $N^{LS}(P[L_{max} > f_l] < Q)$ is slightly lower than $N^{LS}(E[L_{max}] < f_l)$ for probability thresholds Q. For example, $N^{LS}(P[L_{max} > 0.1] < 0.1) = 957$, whereas $N^{LS}(E[L_{max}] < 0.1) = 968$. On the other hand, if 968 streams are admitted using $E[L_{max}] < 0.1$, $P[L_{max} > 0.1] \simeq 0.70$. We can also see from the figure that the differences in the number of supportable users for a given QoS tolerance threshold Q is a decreasing function of Q and f_l. These trends also hold for the 4-60 and 2-10 parameter sets. Table 3 compares these systems, using $N^{LS}(P[L > 0.1] < 0.1)$ and ΔN^{LS}, the difference between $N^{LS}(P[L > 0.3] < 0.2)$ and $N^{LS}(P[L > 0] < 0.2)$.

When users are more tolerant of loss than delay, LS allows the video server to support a higher number of users. For example, for the 3-20 system, $N^{LS}(P[L > 0.3] < 0.01) = 963$, whereas $N^{RD}(P[W > 3] < 0.01) = 952$.

5.3 Sensitivity Analysis

In this section, we examine the sensitivity of the maximum number of supported users under HD, RD and LS, to parameter fluctuations. We vary the mean FF/Rew holding time $1/\mu$ from 5 to 60 seconds, and Φ_F/Φ_P from 2 to 6. Variations in Φ_F/Φ_P reflect video server support for variable-speed FF/Rew, a desirable feature from the user's

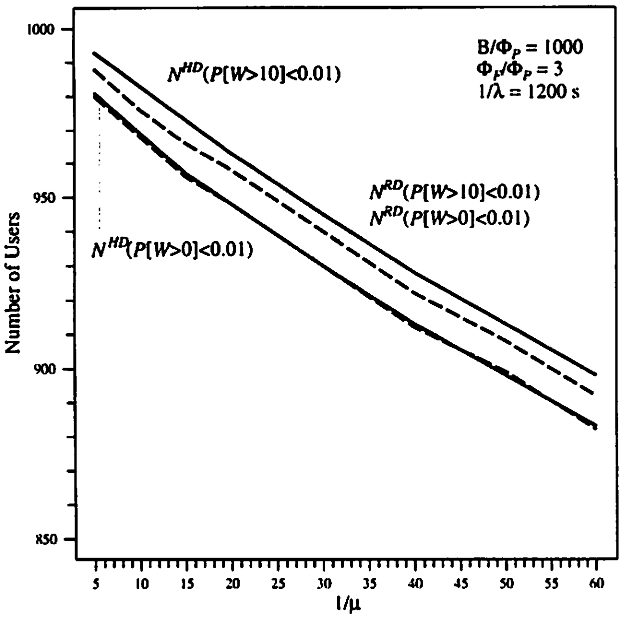


Figure 8: Sensitivity to variation in $1/\mu$ (DS)

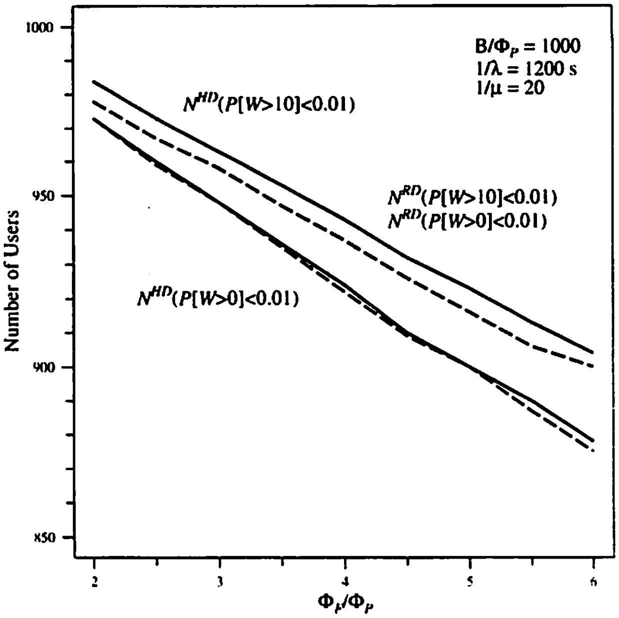


Figure 9: Sensitivity to variation in Φ_F/Φ_P (DS)

perspective and one found on many VCRs and in several continuous media systems under development, e.g. [17]. We graph $N^{HD}(P[W > t] < 0.01)$ and $N^{RD}(P[W > t] < 0.01)$ for $t = 0, 10$ s and $N^{LS}(P[L_{max} > f_l] < 0.1)$ for $f_l = 0, 0.1, 0.3$.

Figures 8 and 9 compare the sensitivity of $N^{HD}(P[W > t] < 0.01)$ and $N^{RD}(P[W > t] < 0.01)$. In general, HD and RD behave similarly. The figures indicate that the server can tolerate large variation in user FF/Rew behavior (e.g. a increase in mean FF/Rew holding time from 20 to 40 s) without service degradation, by reducing the number of users admitted by only 5-10%. The difference between the $t = 0$ curves in both the figures is small for HD and RD because there is very little waiting for FF/Rew service. For the $t = 10$ curves, there is a difference between the curves and it remains relatively constant across variation of both parameters.

In Figures 10 and 11 we note that, under LS, the maximum number of supported users also decreases smoothly in response to increasing parameter values. As in DS, the figures indicate the server can accommodate these fluctuations without service degradation by by reducing the number of users admitted by 5%. The divergent behavior of the curves in both graphs indicates that the maximum number of supportable users is less sensitive to parameter variation for higher levels of loss tolerance.

6 Conclusions

In this paper, we address the issue of providing video-server customers with VCR functions of fast-forward and rewind. We introduce effective FF/Rew service, which provides users FF/Rew capabilities with associated statistical QoS guarantees. This service provides immediate access to full-resolution FF/Rew bandwidth with high probability.

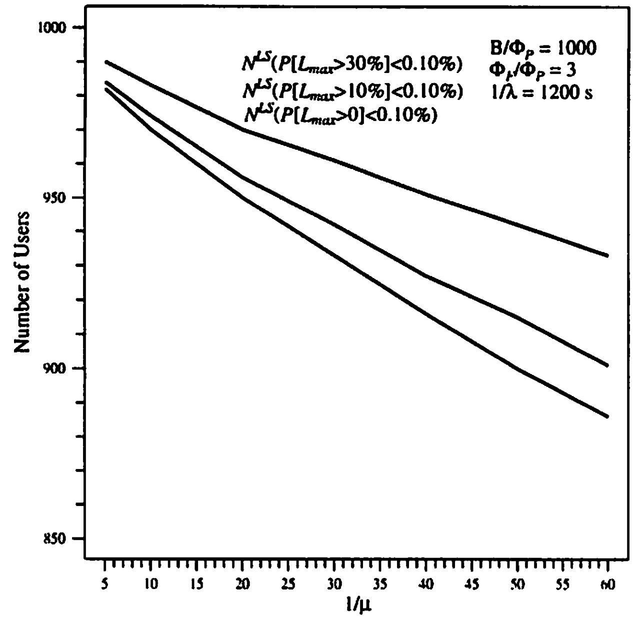


Figure 10: Sensitivity to variation in $1/\mu$ (LS)

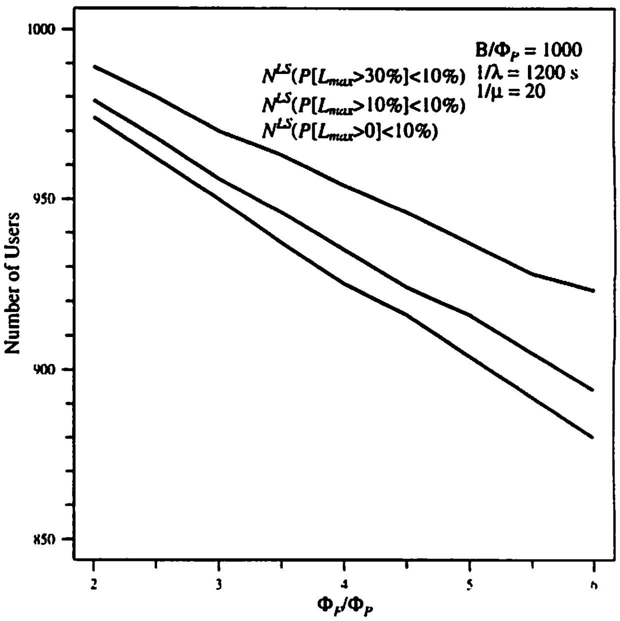


Figure 11: Sensitivity to variation in Φ_F/Φ_P (LS)

When bandwidth is not available, service is either delayed or provided immediately but with loss in resolution. In addition, we introduce several QoS measures to characterize the delay or loss experienced by a FF/Rew request.

We show that the number of users supported under effective FF/Rew service with statistical QoS guarantees is significantly larger than when FF/Rew bandwidth is statically reserved for each user. Moreover, a playback-only video server can be extended to provide this service by reserving only a small portion of its total bandwidth, which is dynamically shared among FF/Rew requests.

The technique of statistical resource sharing has been used in other areas. Human speech consists of talk spurts and silence periods. In intercontinental telephone circuits, when a speaker pauses, the channel can be reassigned to another user [13]. This is done to increase the effective capacity of the circuit. Similarly, processor time-sharing among multiple users takes advantage of the bursty nature of their processing requirements. In packet-based communication, statistical sharing of network resources is a common technique.

Acknowledgements

A number of discussions with David Yates were very helpful in developing the ideas presented in this paper; his comments and insights are gratefully acknowledged.

References

[1] ANDERSON, D. P., OSAWA, Y., AND GOVINDAN, R. A file system for continuous media. *ACM Transactions on Computer Systems 10*, 4 (Nov. 1992), 311–337.

[2] Dey, J. K., Shih, C. S., and Kumar, M. Storage subsystem in a large multimedia server for high-speed network environments. In *IS&T/SPIE Symposium on Electronic Imaging Science and Technology* (San Jose, CA, Feb. 1994).

[3] Emerson, C. A study of the MPEG video coder for use over ATM networks. Master's thesis, Lehigh University, 1993.

[4] Ghanbari, M. Two-Layer Coding of Video Signals for VBR Networks. *IEEE Journal on Selected Areas in Communications 7*, 5 (June 1989), 771–781.

[5] Haskins, R. The Shark continuous-media file server. In *Proceedings of IEEE COMPCON 1993* (San Francisco, CA, Feb. 1993).

[6] Hung, A. C. *PVRG-MPEG CODEC 1.1.* Department of Computer Science, Stanford University, June 1993.

[7] Kamath, M., Ramamritham, K., and Towsley, D. Buffer Management for Continuous Media Sharing in Multimedia Database Systems. Tech. Rep. 94-11, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, 1994.

[8] Kanakia, H., Mishra, P. P., and Reibman, A. An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. In *ACM SIGCOMM 93 Proceedings* (Ithaca, NY, Aug. 1993), ACM, pp. 20–30.

[9] Kandlur, D., Chen, M. S., and Shae, Z. Y. Design of a multimedia storage server. In *IS&T/SPIE Symposium on Electronic Imaging Science and Technology* (San Jose, CA, Feb. 1994).

[10] Karlsson, G., and Vetterli, M. Packet Video and its Integration in to the Network Architecture. *IEEE Journal on Selected Areas in Communications 7*, 5 (June 1989), 739–751.

[11] Lavenberg, S. S., Ed. *Computer Performance Modeling Handbook.* Academic Press, 1983.

[12] Lougher, P., and Shepherd, D. The Design of a Storage Server for Continuous Media. *The Computer Journal (special issue on multimedia) 36*, 1 (Feb. 1993), 32–42.

[13] Martin, J. *Communications Satellite Systems.* Prentice Hall, 1978.

[14] Patel, K., Smith, B. C., and Rowe, L. A. Performance of a Software MPEG Video Decoder. In *Proceedings of ACM Multimedia 1993* (Anaheim, CA, Aug. 1993).

[15] Ramakrishnan, K. K., and et al. Operating system support for a video-on-demand file service. In *Proceedings of 4th International Workshop on Network and Operating Systems support for Digital Audio and Video* (Lancaster, UK, Nov. 1993).

[16] Rangan, P. V., Vin, H. M., and Ramanathan, S. Designing an on-demand multimedia service. *IEEE Communications Magazine 30*, 7 (July 1992), 56–64.

[17] Rowe, L. A., and Smith, B. C. A Continuous Media Player. In *Proceedings of 3rd International Workshop on Network and Operating Systems support for Digital Audio and Video* (San Diego, CA, Nov. 1992).

[18] Staehli, R., and Walpole, J. Constrained-Latency Storage Access. *IEEE Computer Magazine* (Mar. 1993), 44–53.

[19] Tom, A. S., Yeh, C., and Chu, F. Packet video for cell loss protection using deinterleaving and scrambling. In *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Processing* (Toronto, Ont., Canada, May 1991), IEEE, pp. 2857–2860.

[20] Turner, C. J., and Peterson, L. L. Image Transfer: An End-to-End design. In *ACM SIGCOMM 92 Proceedings* (Baltimore, MD, Aug. 1992), ACM, pp. 258–268.

[21] Vin, H. M., and Rangan, P. V. Designing a multi-user HDTV storage server. *IEEE Journal on Selected Areas in Communications 11*, 1 (Jan. 1993).

[22] Wang, Y., Zhu, Q.-F., and Shaw, L. Maximally Smooth Image Recovery in Transform Encoding. *IEEE Transactions on Communications 41*, 10 (Oct. 1993), 1544–1551.

Appendix

In the appendix, we derive the equations for the Birth-Death processes representing each of the HD, RD and LS systems. We also describe the derivation of the QoS metrics that are computed analytically.

HD mechanism

Under HD with N users, when i users request FF/Rew, the arrival rate into the FF/Rew system is Poisson with rate $(N-i)\lambda$ and the rate at which users depart the system is $\min(m_{HD}(N), i)\mu$. If p_i is the probability that the system is in state i, $0 \leq i \leq N$, we have:

$$p_i \min(m_{HD}(N), i)\mu = \qquad p_{i-1}(N-i+1)\lambda, \quad 0 \leq i \leq N. \tag{3}$$

From equation 3, we can derive all the QoS metrics of interest. Let N_F be the random variable denoting the number of users in FF/Rew mode at any point in time. The expected number of users doing FF/Rew is given by:

$$E[N_F] = \sum_{i=0}^{n} i p_i. \tag{4}$$

The tail probabilities under HD are computed as follows. A FF/Rew request has to wait if it arrives into state $m_{HD}(N)$ or higher. The probability that a user has to wait for k users to complete FF/Rew, i.e. the probability that the user enters the system in state $(m_{HD}(N)+k-1)$, is:

$$w_k = \frac{(N - m_{HD}(N) - k + 1)p_{m_{HD}(N)+k-1}}{\sum_{i=0}^{N-1}(N-i)p_i}, \quad 1 \leq k \leq N - m_{HD}(N).$$

Hence, the average waiting time of a user is:

$$E[W] = \sum_{k=1}^{N-m_{HD}(N)} \frac{k w_k}{\mu m_{HD}(N)}.$$

The waiting time distribution of a user waiting for k users to complete FF/Rew is described by a k-th order Erlang random variable. If $F_{E_k}(t)$ is the probability distribution function for the random variable [11], the tail probabilities under HD are given by:

$$P[W > t] = \sum_{k=1}^{N-m_{HD}(N)} w_i F_{E_i}(t).$$

RD mechanism

Under RD with N users, when i users request FF/Rew, the arrival rate into the FF/Rew system is Poisson with rate $(N-i)\lambda$ and the rate at which users depart the system is $\min(m_{RD}(N,i), i)\mu$. If p_i is the probability that the system is in state i, $0 \leq i \leq N$, we have:

$$p_i \min(m_{RD}(N,i), i)\mu = p_{i-1}(N-i+1)\lambda, \quad 0 \leq i \leq N.$$

The expected number of users in FF/Rew mode is computed exactly as in equation 4. However, the amount of time a FF/Rew request has to wait depends upon both the number of FF/Rew requests ahead of it in the FF/Rew queue, but also upon the number of users who arrive after after it (since addition of users in FF/Rew queue increases the available bandwidth in the FF/Rew queue). Thus it is difficult to compute $E[W]$ and its tails analytically. We thus compute them through simulation.

The expected number of FF/Rew servers is:

$$E[m_{RD}] = \sum_{i=0}^{N} m_{RD}(N,i)p_i.$$

LS mechanism

Under LS, when i users request FF/Rew, the arrival rate into the FF/Rew system is Poisson with rate $(N-i)\lambda$ and the rate at which users depart the system is $i\mu$. Let p_i is the probability that the system is in state i, $0 \leq i \leq N$. Since no FF/Rew request has to wait before being serviced, the equation is simply:

$$p_i i\mu = p_{i-1}(N-i+1)\lambda, \quad 0 \leq i \leq N.$$

Once again, the expected number of users in FF/Rew mode is given by equation 4. The fractional loss seen in the system is:

$$E[L_{sys}] = \sum_{i=0}^{N} f_l(i)p_i.$$

As in HD, the fractional loss seen by a FF/Rew request depends upon both the number of earlier FF/Rew requests, as well as the number of subsequent FF/Rew requests. Thus the QoS metrics for L and L_{max} have been computed via simulation.