Modeling and Evaluation of an Online TV Recording Service

Tobias Hoßfeld[1] and Kenji Leibnitz[2]

[1] University of Würzburg, Institute of Computer Science
Am Hubland, 97074 Würzburg, Germany, *hossfeld@informatik.uni-wuerzburg.de*

[2] Osaka University, Graduate School of Information Science and Technology
1-5 Yamadaoka, Suita, Osaka, Japan, *leibnitz@ist.osaka-u.ac.jp*

Abstract

In this paper we investigate the performance of a video delivery service using the OnlineTVRecorder.com service in Germany as an example. We show that the request arrivals for file downloads play an important role and the system reacts differently when the arrivals are time-dependent. We consider two simple analytical models: a steady state Markov chain analysis for constant arrivals and a fluid model to capture flash crowd effects. Furthermore, our analytical approach also takes the distribution of the offered files into account, as well as the users' impatience which leads to aborted downloads.

1 Introduction

Recently, new services have emerged which utilize the Internet as a delivery mechanism for multimedia content. With the advent of broadband accesses, more users are willing to download large volume content from servers, such as video files of TV shows. While some popular video services (e.g. YouTube.com) or some broadcasting companies (e.g. ABC.com) use streaming data with Flash technology, some media distributors (e.g. iTunes) offer entire TV shows for download. In this study, we investigate the performance of the German site OnlineTVRecorder.com (OTR), which acts as an online video cassette recorder (VCR) where users can program their favorite shows over a web interface and download the recorded files from a server or its mirrors. These files are offered in different formats and can consist of several hundred megabytes up to 1 GB or more depending on the length of the TV show as well as the encoding format. OTR can, thus, be seen as an example for a server-based content distribution system with large data files.

However, as these server farms are often overloaded, new requests are queued when the provided download slots are full. The restriction to a maximum number of simultaneous downloads guarantees a minimal download bandwidth for each user. Additionally, the service offers premium users prioritized access to downloading. The download duration itself depends on the total capacity of the server and the number of users sharing this capacity. On the other hand, users who might encounter slow downloads may abort their downloading attempt if their patience is exceeded.

In this paper, we discuss the impact of the user's impatience on the performance of such an OTR server with different file size distributions. The paper is organized as follows. After describing the problem and formulating simple analytical models, we provide numerical results and compare their performance in terms of download duration and success ratio. Especially, we address the question of how to properly dimension the number of simultaneous downloads at a server in order to optimize the performance of the system and to maximize the user's satisfaction.

2 Problem Formulation and Analytical Model

Let us consider the following system. User requests arrive at the server with an arrival rate λ. While we will at first consider a fixed arrival rate in order to evaluate a steady state Markov model, we will also consider later a non-stationary arrival rate $\lambda(t)$. This is a more realistic scenario when looking at individual files, since the popularity of a TV show highly depends on the time it was recorded. Once a show becomes outdated, the interest for this file decreases. This phenomenon is usually referred to as *flash crowd arrivals* [1]. However, since a server may offer several different files, the overall arrival rate may remain nearly constant. The superposition of time-dependent arrival processes with different starting points can be modeled as stationary Poisson process for a sufficiently large number of offered files per server.

When a request arrives and there are free download slots, the client may proceed with the download. We assume that the server system has a total fixed capacity C which is shared among all simultaneously downloading clients $D(t)$ at time t. The maximum number of users served in parallel is restricted to n. Thus, the time-dependent download rate $\mu(t)$ is

$$\mu(t) = \frac{1}{f_s} \min\left\{ \frac{C}{\min\{D(t), n\}}, R \right\} \tag{1}$$

for a file size f_s and the download rate is limited by the physical rate R of each client.

As we need the distribution of the file sizes to compute the download rate $\mu(t)$, we investigated the actual file sizes of video files offered at OTR. The measurements which were made in April 2007 show that the actual file size distribution over 11563 file samples from 19 different TV channels has a mean of 368.31 MB and standard deviation of 196.82 MB. It can be well fitted by an Erlang-k distribution with $k = 3.34$ phases and an average volume of B =107.67 MB per phase,

i.e., it is the sum of $\lfloor k \rfloor$ independent identically distributed random variables each having an exponential distribution with mean B and an exponential distribution with mean $(k - \lfloor k \rfloor)\, B$.

2.1 Discussion of the Model

In general, with a slight abuse of the Kendall notation for queuing systems, the model as described above can be expressed as M(t)/GI/1^n-PS with user impatience θ, an unlimited waiting queue, and a server capacity which is shared among n users at maximum. Thus, the service rate is influenced by μ and θ and depends on the number of currently served users.

Admission control to the system can be taken into account by restricting the size of the waiting queue. However, in this paper we use the number of download slots n to guarantee the bandwidth per user and only investigate the impact of the user's impatience on the system's performance. While *reneging* is considered with an i.i.d. random variable θ, *balking*, i.e., taking back the download request if the waiting queue is too long, is neglected in this paper. We focus on the effect of wasted capacity due to users' impatience regardless of whether they are being served or not, and the impact of variability of the file size distribution, which is expressed by the service rate. Our findings show that the ratio of successful downloads increases with the variability of service time.

Basically, there are several approaches on how to analytically evaluate such a system depending on the number of available download slots n. If $n \leq \lfloor \frac{C}{R} \rfloor$, the user's access bandwidth limits the download rate. This effectively results in a M(t)/GI/n-FCFS system with independent service rates, since θ is an i.i.d. r.v. and μ is constant. An analytical evaluation is provided in [2]. For $n > \lfloor \frac{C}{R} \rfloor$, the download rate and therefore the service rates depend on the current state of the system. On the other hand, if the downlink of a user is not the limiting factor, i.e., a user can always utilize the offered bandwidth of the server ($C < R$), the system approaches a real processor sharing system with increasing n, which is investigated in [3, 4].

In order to emphasize the effects of the system, we consider in this paper only very simple models which are easily analytically tractable. It is well known that for systems of type M/GI/n only approximative evaluations can be performed for metrics of interest [5]. Several problems arise when an evaluation is performed at a higher level of detail. Firstly, this is because we consider time-dependent flash crowds arrivals requiring a transient analysis as described later in Section 2.3. Furthermore, several (virtual) service units ($n > 1$) with general service time and general impatience make it difficult to provide an exact analysis.

2.2 Steady State Analysis with Markov Model

We now consider a steady state analysis for evaluating the performance of the server system with aborted downloads due to impatience. We assume homogeneous users with equal access bandwidths R and generally independent patience time θ. In our model, θ is the time threshold after which a user aborts his download attempt if the download time[1] takes longer than that. However, this GI assumption is not an accurate model for the actual users' behavior. In reality, a user will have a state-dependent patience, since he is more willing to wait if the file is nearly completed, cf. [6]. However, in order to make the model analytically tractable, we consider an exponentially distributed θ. The model will be denoted M/M/1^n-PS. Thus, we have a homogeneous Poisson arrival process, exponential service time, a single server unit which services up to n clients and operates with the processor sharing regime. Note that bandwidth restrictions of the users' downlink capacity are taken into account. The queue length for waiting users is assumed to be infinite.

The model itself is a simple birth-death process where only transitions between neighboring states are possible. The service rates μ_i are dependent on state i and are expressed in (2). With the resulting state probabilities the waiting time, sojourn time, and success ratio can be obtained.

$$\mu_i = \frac{i}{\theta} + \min\{i, n\}\, \frac{1}{f_s} \min\left\{ \frac{C}{\min\{i, n\}}, R \right\} \qquad i = 1, 2, \ldots \tag{2}$$

2.3 Time-Dynamic Evaluation with Fluid Model

The Markov model described in the previous section only allows to investigate the steady state behavior. In order to also consider the flash crowd arrivals mentioned above, we use a fluid analysis technique, see (3).

$$\dot{W} = \begin{cases} 0 & \text{if } D < n \\ \lambda - D\,\mu - \nu\, W & \text{otherwise} \end{cases} \quad \dot{D} = \begin{cases} \lambda - D\,\mu & \text{if } D < n \\ 0 & \text{otherwise} \end{cases} \quad \dot{A} = D\, p\, \mu + \nu\, W \quad \dot{F} = D\ (1 - p)\ \mu \tag{3}$$

Arrivals enter the waiting population W with rate λ or directly the downloading population D, if the number of slots n is not full. If the slots are full, waiting users simply proceed to the downloading state with rate $\mu\, D$. After entering state

[1] In this work the sojourn time of a user in the system, i.e., the sum of the waiting and the service time, is referred to as download time.

D, the client remains in this state until he either fully downloads the file and enters the finished state F or he aborts the download when the download duration exceeds his patience threshold θ. The latter is expressed by entering abort state A. In both cases the transitions are performed at rate μ multiplied with a probability p (when the download fails) or $1-p$ in the case of success. The probability p can be interpreted in the following way. An abort occurs when the patience of the downloading user is exceeded either during downloading or waiting. The patience in this model is characterized by the exponential random variable θ with rate $\nu = 1/\mathrm{E}[\theta]$ and the downloading time is exponentially distributed as well with rate $\psi = \frac{C(t)}{\mathrm{E}[f_s]}$. The variable $C(t)$ denotes the time-dependent capacity per user, i.e., $C(t) = \min\left\{\frac{C}{\min\{D(t),n\}}, R\right\}$, and $\mathrm{E}[f_s]$ is the mean file size. Thus, the probability that the patience is exceeded at time t can be expressed as

$$p(t) = \frac{\nu}{\nu+\psi} = \frac{\mathrm{E}[f_s]}{\mathrm{E}[f_s] + C(t)\,\mathrm{E}[\theta]}. \tag{4}$$

Note that in the case of a single downloading state D, exponential file sizes f_s and thus exponentially distributed rates μ are assumed. If we consider Erlang-k distributed file sizes as obtained in our measurements, the state D must be expanded to several intermediate states $D_0, D_1, \ldots, D_k$. For $k \to \infty$ this approaches deterministic values.

With the computation of the population dynamics of the downloading users, we obtain the dynamics of the download rates from Eqn. (1). In particular, for a starting time t_0 the duration $d(t_0)$ can be computed by integrating μ over time, i.e., $\int_{t_0}^{d(t_0)} \mu(t)\,dt = 1$.

3 Numerical Results

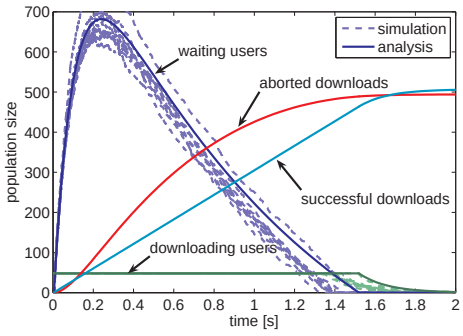

Figure 1: Population changes with fluid model

Due to the space limitations we focus on the flash crowd scenario with the fluid model. We assume an exponentially decreasing arrival rate $\lambda(t) = \beta e^{-\alpha t}$ with $\beta = 1$ and $\alpha = 10^{-3}$. Thus, the total number of arriving users in the system is limited to $\lim_{t\to\infty} \lambda(t)dt = \frac{\beta}{\alpha} = 1000$. Fig. 1 shows the time-dynamic evolution of the population size in the flash crowd scenario. We compare the population sizes from several simulation runs with the numerical solution of the differential equation system (3).

In the following we look at the different behavior of the system when there are constant and flash crowd arrivals. In order to compare systems with both types of arrivals, we match the arrival rate for the constant case to get the same number of arrivals as in the case of flash crowds. Here, we use the parameters $\beta = 1$, $\alpha = 10^{-4}$, as well as the server capacity $C = 100\,\mathrm{Mb/s}$, user bandwidth $R = 2\,\mathrm{Mb/s}$ and patience threshold $\theta = 200\,\mathrm{min}$, and the file size distribution is taken from measurement values, as well as $n = \lfloor C/R \rfloor$ download slots.

Fig. 2 and Fig. 3 depict the two measures of interest to us, the download time and success ratio for two exemplary simulation runs. We take a look at the temporal evolution using a moving average with a window size of 100. Both figures show that there is a significant difference when constant or time-dependent arrivals are considered. With a constant arrival rate, after an initial transient phase, both the download duration and the success ratio become constant. With flash crowds, there is a higher variation of both values as the arrivals rapidly decrease over time from which later arrivals benefit. The figures show that it is very important to consider if the arrivals are time-dependent or not, as they yield quite different results.

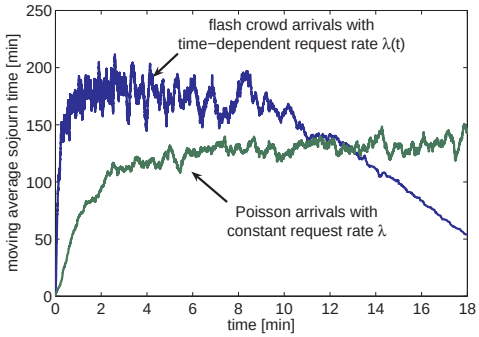

Figure 2: Sojourn times for flash crowd / Poisson arrivals

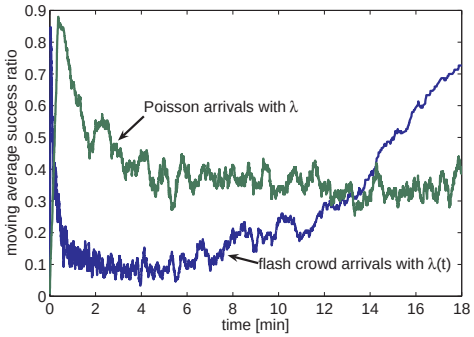

Figure 3: Moving average of success ratio

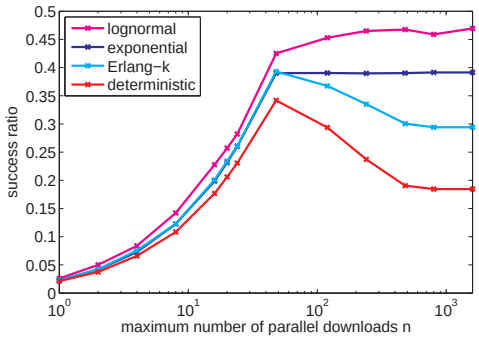

Figure 4: Success ratio for different file size distributions

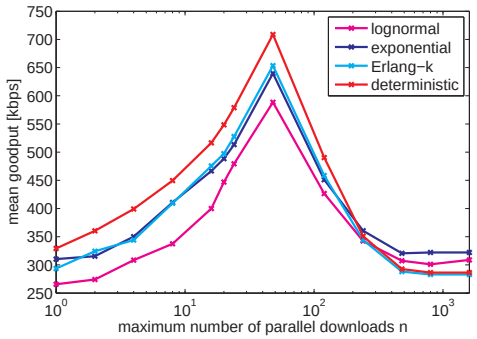

Figure 5: Goodput depending on file size variation

The next investigation aims at the optimal dimensioning of the number of download slots n for different file size distributions. We focus on the flash crowd scenario with the same parameters as above except for $\alpha = 10^{-3}$. While Fig. 4 shows the success ratio when the file size is distributed either deterministic, exponential, Erlang, or lognormal, Fig. 5 depicts the average goodput in kbps depending on the maximum number n of simultaneously served users. Both figures illustrate the influence of the coefficient of variance on the system behavior. What is remarkable is that for deterministic and Erlang-distributed file sizes a maximum success ratio exists, whereas for exponential and lognormal the success ratio remains nearly constant when n is larger than the optimal value $n = \lfloor C/R \rfloor$. However, this is caused by the fact that in systems with higher coefficients of variation smaller files are downloaded more often. In all four cases the goodput is highest at this value, as can be seen from Fig. 5. The goodput is defined as the ratio of the file size and the download time for successful downloads. For larger n the system capacity is wasted due to longer download times caused by capacity sharing and the aborting of a download due to the user's impatience.

4 Conclusion and Outlook

In this paper we discussed the performance of an online TV recording service for distributing large-volume video files. The user behavior was characterized with an impatience threshold after which the client aborts the download. We derived two simple analytical models, a stationary and a transient fluid flow model and compared their performance in terms of the mean download duration and success ratio.

In the future, we wish to perform a more detailed analysis which can be used for comparison to other content distribution methods, e.g. using peer-to-peer networks [7]. By utilizing the benefits of distributed serving nodes as in P2P with optimal strategies for caching contents, our goal is to design better content distribution networks with a higher reliability and scalability.

References

[1] T. Hoßfeld, K. Leibnitz, R. Pries, K. Tutschku, P. Tran-Gia, and K. Pawlikowski, "Information diffusion in eDonkey-like P2P networks," in *Proc. ATNAC 2004*, (Bondi Beach, Australia), 2004.

[2] N. Gans, G. Koole, and A. Mandelbaum, "Commissioned paper: Telephone call centers: Tutorial, review, and research prospects," *Manufacturing & Service Operations Management*, vol. 5, no. 2, pp. 79–141, 2003.

[3] J. E. G. Coffman, A. A. Puhalskii, M. I. Reiman, and P. E. Wright, "Processor-shared buffers with reneging," *Perform. Eval.*, vol. 19, no. 1, pp. 25–46, 1994.

[4] H. C. Gromoll, P. Robert, B. Zwart, and R. Bakker, "The impact of reneging in processor sharing queues," in *Proc. of SIGMETRICS '06/Performance '06*, (New York, NY, USA), pp. 87–96, ACM Press, 2006.

[5] B. Gnedenko and D. König, *Handbuch der Bedienungstheorie II*. Berlin: Akademie-Verlag, 1984.

[6] S.-C. Yang and G. de Veciana, "Bandwidth sharing: The role of user impatience," in *Proc. IEEE GLOBECOM*, (San Antonio, TX), pp. 2258–2262, Dec. 2001.

[7] K. Leibnitz, T. Hoßfeld, N. Wakamiya, and M. Murata, "Peer-to-peer vs. client/server: Reliability and efficiency of a content distribution service," in *20th International Teletraffic Congress (ITC-20)*, (Ottawa, Canada), June 2007.