© ACM In: Proc. of 8th ACM Int. Conf. on Multimedia (ACM-MM) 2000 pages 117–126

Stochastic Resource Prediction and Admission for Interactive Sessions on Multimedia Servers

Matthias Friedrich[*] , Silvia Hollfelder, Karl Aberer
GMD–IPSI Integrated Publication and Information Systems Institute
Dolivostr. 15, 64293 Darmstadt, Germany
hollfelder@gmd.de

ABSTRACT

In highly interactive multimedia applications startup latency is significant, and may negatively impact performance and Quality of Service (QoS). To avoid this, our approach is to admit whole multimedia sessions instead of single media streams. For the prediction of the varying resource demands within a session, which are mainly correlated to user behavior, we model user behavior as Continuous Time Markov Chains (CTMCs). In this paper, we propose a mathematical analysis of the CTMC model. This allows to anticipate possible overload and in turn to plan an admission control policy. As a result, our approach provides better control on the tradeoff between server utilization and QoS. Simulation studies confirm this capability.

Keywords

Admission Control, Interactive Multimedia Applications, Continuous Time Markov Chains

1. INTRODUCTION

Interactive multimedia presentations are fundamental to many advanced application domains, like home entertainment (e.g., news on demand, interactive VoD, action games), e-commerce (e.g., electronic product catalogs), education and training (e.g., computer-based training - CBT), and multimedia production (e.g., video editing). In these applications, users interact frequently with the system for a variety of activities, including VCR-control and request for typically short media chunks, such as video scenes or text documents, and the selection of media combinations (e.g., synchronized audio and video). In the following, we understand a *multimedia session* as a user's successive requests of different media objects to fulfill his information needs for a specific content.

The presentation time of selected media objects is typically very short in these applications. High startup latency in between subsequent presentations are thus not tolerable [12], and intra- and intermedia synchronisation requirements must also be considered.

Admission control mechanisms are used to limit the number of clients to be served in order to meet each clients' Quality of Service (QoS) requirements and to achieve high resource utilization. Most classical admission control mechanisms, like those for video-on-demand applications, handle requests for a *single* media stream playout. In this case, the resource requirements are pre-specified in terms of constant rate or rate deviations [23] and calculated by stochastic ([18], [28]) or deterministic approaches ([27], [20]).

For highly interactive applications, the admission of single streams leads to intolerable startup delays *within* a multimedia session at high system load. We claim that admission control for such applications has to be granted at *session level* to reduce startup latency. But a session-oriented admission control schema needs to adequately take into account the *highly varying* resource requirements. These do not only vary due to bursty VBR-media presentations [21] and VCR-control, but also to users' selections of media objects (that can be encoded in different formats) and media object combinations to present.

To analyze the resource consumption of interactive multimedia sessions, we first develop a user behavior model which exploits domain specific knowledge. Second, we utilize the model to stochastically predict resource demands for admission control purpose. In our approach, we do not consider system component specific properties, such as disk bandwidth or buffer space, but we assume that system resources are available in terms of throughput of a multimedia system.

Our model restricts the subset of media that can be requested within a multimedia session. This is realistic, for example, when a user retrieves its media objects via a query. Another example is a preorchestrated SMIL document in which the temporal order and possible interaction points are predetermined [29]. In our approach, we represent an uncertain user behavior by means of a Continuous Time Markov Chain (CTMC) model [25], which enables to stochastically represent the presentation times of media objects and the interaction probabilities. In previous work, we demonstrated how the parameters of a CTMC can be deduced by employing application domain knowledge. For example, for video browsing the user behavior can be heuristically specified from a ranked query result list [2]. In [14], we specified how user behavior in electronic multimedia catalogs can be monitored and applied to data mining technologies. The main difficulty is that heuristic assumptions on how the users behave with regard to their interaction possibilities are required.

Future resource needs are stochastically predicted for a look-ahead time window based on the user behavior model.

[*] New address: Mummert und Partner Consulting AG, Kölner Strasse 44, 60327 Frankfurt, Germany

The time window can be analyzed at various granularity levels. This prediction is then employed for admission control, when the server tests whether an overload situation occurs due to the admission of a pending client.

The level of granularity depends on various factors, like the ratio of total system resources and clients' resource requirements, the distribution of the resource requirements within a session, and the stability of the deviation patterns. For a large number of clients with uniform behavior, a simple method based on average resource consumption may suffice. Such a method is proposed in [2], where equilibrium analysis for the video browsing scenario is employed. However, equilibrium analysis can only be applied to specifically structured, so-called closed, CTMCs and is therefore more useful for long-lasting sessions.

For fewer clients with large deviations in consumptions, a more fine-grained analysis can be beneficial or even crucial. In this paper, we focus on a fine-grained admission control strategy. We perform a transient analysis on the CTMC that is much more precise than the equilibrium analysis, and is applicable to all types of CTMCs. As an additional advantage, the method is also flexibly adaptable to different application requirements by adjusting the time granularity in which the analysis is performed. Large timesteps smooth out the data rates and are adequate in scenarios with more uniform data rate demands, while decreasing the timesteps “zooms into” the temporal structure of the multimedia presentations, thus allowing more detailed predictions on resource consumption. In this paper, we focus on the mathematical analysis of the model.

The further content of the paper is structured as follows: In Section 2, we give a survey of the related work. We present our approach in detail, in Section 3, and show simulation results obtained by using our admission control strategy in Section 4. In Section 5, we conclude the paper with a result summary and some open issues.

2. RELATED WORK

Typically, admission control mechanism are classified by the service guarantees (i.e., predictive, stochastic, deterministic), and the system resources that are taken into account (e.g., buffer space, disk I/O, etc.). In the following, we take another focus and classify related work by the general idea behind. We evaluate approaches that support VCR-interactions or multimedia sessions with frequent changes in resource demands. However, there is no related work that covers both, interactivity and media object compositions, appropriately.

VCR-support for single streams can be given by a *priori reservation* of separate server bandwidth ([9], [3]), but leads to low system utilization or low QoS in case of highly varying resource demands. The *smoothing* of data rates for VCR-interactions aims to achieve a relatively constant workload ([24], [6], [5]). This proceeding is restricted to VCR-interactions, since it would lead to unacceptable quality degradations of media with originally high resource demands. A straightforward way for VCR-support is to *re-admit by priorities at interaction points* to reduce startup latency in case of high system load ([22], [8]). The problem here is that high variations in consumption rates due to media switches are neglected. *Server caching* for VCR-interactions ([30], [3]) needs applications with high probabilities for the access of same media data, such as news on demand.

Re-admission at media switches is proposed in [12] by adressing interactive hypermedia sessions consisting of discrete and continuous data requests. This works well for low workload (e.g., 50 percent), but startup latency gets intolerable for higher workload.

From **pre-orchestrated presentations** knowledge on the exact resource demands can be employed when the users are not allowed to control the presentation process interactively. This information can be used for admission of whole multimedia sessions, as realized in the following. Prefetching and replication heuristics in multidisk environments are proposed in [10]. The goal is to achieve low latency for few additional memory requirements, assuming that disk bandwidth is the scarce system resource. This idea is not applicable for our purpose, since interactions are not allowed at all. The goal in [31] is to determine a starting point for a session so that no bottlenecks occur. The basic reservation model does not consider user interactions, but following extensions for *interactions* are suggested: (1) the specification of a minimum upper bound which is not efficient for our purpose and (2) re-admission at media switches as discussed previously. Another schema for the admission of composite multimedia presentations is proposed in [4]. The servers' buffer space is the critical, limited system resource. They briefly propose two extensions for interactive applications, namely re-admission at media switches and a priory reservation.

Observation-based admission. The general idea here is to assume that the past behavior is an indicator for the future. This approach can be employed for single media streams [26] as well as for whole multimedia sessions ([16], [13]). It is applicable for a large number of parallel sessions with even access patterns, but problems occur at low capacity with bursty or unpredictable client behavior.

A more detailed analysis of related work can be found in [11].

3. APPROACH

In the following we describe a stochastic admission control strategy for multimedia presentation scenarios. Our approach, called *Admission Control Based On Stochastic Prediction (ACSP),* rests upon the prediction of the overload probability at the server resulting from the admission of an additional client. The precision of the prediction in terms of fine-granular time-intervals can be flexibly adapted by choosing the length of the *rounds* which will be statistically analyzed one by one. In this section, we describe the user behavior model representing an interactive session, the admission control system architecture, and the statistical approach to resource usage prediction.

3.1 Modelling of interactive client sessions

For modelling of user behaviors, we assign to each client an application class c $(c = 0, ..., C - 1)$ which is further differentiated into a finite number of states i $(i = 0, ..., S_c - 1)$ representing the different presentation modes. Each client is allowed to switch between the states of his class. At each point in time, a client is assigned to a unique current state. In our context a transition from one state to another is interpreted as an interaction of the client leading to another presentation mode.

Each class c models a set of clients with similar behaviors. This means that the *duration times in the states,* the *data rates requested on average* and the *transition behavior* of the clients belonging to class c are either identical or similar. The model can be flexibly used for different desired levels of accuracy, by combining subsets of clients according to different degrees of similarity in their behaviors.

For illustration purposes, we give the concrete example for a preorchestrated interactive multimedia application, where the temporal and spatial relationships of components within a session are modelled. In Figure 1 the nodes relate to presentation of media objects and the arrows represent the possible user interactions, i.e., a user selects another media object for presentation. Since our main problems to be solved in this paper are user interactions in general, we will neglect VCR-modelling, which is only one specific case. However, VCR-interactions can be simply added to each playback state by additional VCR-mode states and the corresponding transition probabilites to these states.

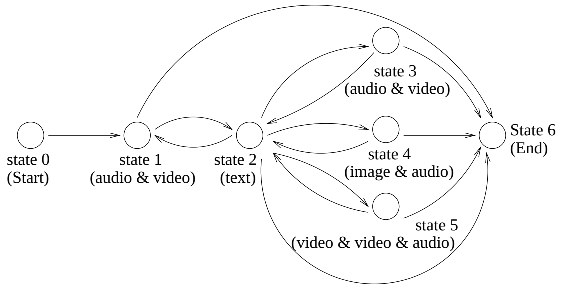

Figure 1: Example of a Preorch. MM Document

3.2 State Transitions and Data Rates

The transitions between the states of a client class are modelled as a stochastic process $X^c(t)$. $X^c(t)$ indicates the state of a client of class c at time t. For statistically modelling the state transition system, we use the Continuous Time Markov Chain model (CTMC). Such a process is stationary, time-continuous, and has the Markovian Property, i.e., it is memoryless. In scenarios with high user interaction probabilities, the CTMC assumption is realistic. For our model, the Markovian property means the following: If a client of class c leaves state i, the probability of his moving to state j is always p^c_{ij}, no matter how he attained state i. For the p^c_{ij}, which will subsequently be called *time-independent transition probabilities*, the following equations hold:

$$\sum_{j=0, j\neq i}^{S_c-1} p^c_{ij} = 1. \tag{1}$$

If a client moves to state j of class c, he stays there for a time interval being exponentially distributed with parameter v^c_j, independent of how he reached state j. We will subsequently call the v^c_j *leaving rates*.

The implementation of the ACSP requires the identification of the classes and their corresponding states, as well as the determination of the parameters p^c_{ij} and v^c_j. The latter could possibly be carried out by means of observation. In the subsequent part of this paper, we therefore assume that the time-independent transition probabilities as well as the leaving rates are known.

We distinguish two different types of states the client may reside in, namely *active states*, in which they request data at a specific data rate, and *idle states*, in which they are inactive. In active states, the requested average data amount per round for a client of class c is described by the random variable N_{ci}. We assume N_{ci} to be rectangularly distributed, with minimum u_{ci} and maximum v_{ci}.

We consider the rectangular distribution for modelling the active states to be appropriate as it describes a corridor which the requested data amounts fall into. The width of this corridor $v_{ci} - u_{ci}$ captures the variability of the data rates requested by the clients. In the idle states, the clients request with probability 1 no data.

3.3 Implementation of the admission control module

Within the admission control module, we predict the probability of a situation in which the given server resources are smaller than the amount of resources requested by the clients. Such a situation is called *overload*. Our prediction is performed for a look-ahead time window W starting at time t_{start}. We further divide W into rounds which all have the same length l_r (see Figure 2). The statistical analysis in the prediction is performed for each round separately.

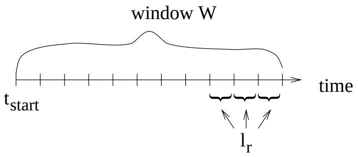

Figure 2: Time Window for Resource Prediction

In this analysis, the random variable T^t_s is the time the server needs in the round starting at time $t_{start} + t$ for serving the data requested by all clients to be served on average. Then an overload is the probability that T^t_s is greater than l_r. Thus, for each of the rounds in W, we calculate an upper bound $ub(t)$ for the probability that, within this round, an overload situation occurs. For $ub(t)$ the following holds:

$$ub(t) \geq P(T^t_s > l_r) \tag{2}$$

The calculation of $ub(t)$ is the essential part in the ACSP. It is performed in two separate steps (see Figure 3). First, in a prognosis module 1, a matrix $M(t)$ is calculated which describes the predicted number of clients in each state of each class for the round beginning at time t including the client to be admitted. Second, in a prognosis module 2, $M(t)$ is used to calculate $ub(t)$. Note that this analysis can be precomputed and thereby computational overhead at admission time is reduced.

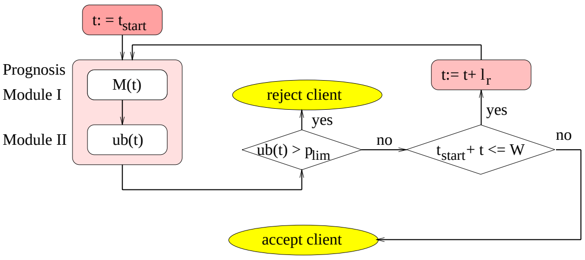

Figure 3: Flow Diagram of the ACSP Mechanism

Figure 3 shows how the admission control within the ACSP proceeds on the arrival of a new client. At the beginning of the ACSP process the time parameter t is set to the value t_{start}, i.e., to the beginning of the first round within W. Next, within the prognosis modules 1 and 2, the upper bound $ub(t)$ for that first round is calculated. This value is then compared with a value p_{lim} representing the maximal overload probability the system is willing to accept without rejecting the client. The value p_{lim} is a configuration parameter showing how optimistic or pessimistic the admission control proceeds. High values for p_{lim} lead to a larger number of admitted clients and therefore to more frequent overload situations. If $ub(t)$ is greater than p_{lim}, the client is rejected, otherwise the time parameter t is increased by l_r, the matrix $M(t+l_r)$ and the upper bound $ub(t+l_r)$ are calculated, and the algorithm proceeds as described before. If, for none of the rounds within W, the calculated upper bound of the overload probability is greater than p_{lim}, the new client is accepted to service. Otherwise, if $ub(t)$ exceeds p_{lim} for a single round, the client is rejected.

3.4 The determination of $M(t)$ in prognosis module 1

The objective of the calculations within prognosis module 1 is to determine the matrix $M(t)$. $M(t)$ is a $C \times S$ matrix with non-negative integers. S is the maximum number of states a class contains. An element m^t_{ci} of $M(t)$ represents the estimated number of clients being in state i of class c at the round beginning at time t. For the calculation of $M(t_{start})$, all admitted clients as well as the actual client requesting for admission are taken into account. We assume for the calculation that none of the clients will leave the system during the time window W, i.e., the following equations must hold:

$$\sum_{i=0}^{S_c-1} m^t_{ci} = \sum_{i=0}^{S_c-1} m^{t_{start}}_{ci} \tag{3}$$

If, contrary to our assumption, one client stops requesting the server during the time window W, the predictions do not become invalid but only more conservative.

To calculate $M(t)$, we first determine the *time-dependent transition probabilities* $p^c_{ij}(t)$.

$$p^c_{ij}(t) = P(X^c(t) = j | X^c(0) = i) \tag{4}$$

As the transition process $X^c(t)$ is stationary and has the Markovian Property, the $p^c_{ij}(t)$ can be interpreted as the probability that the client moves from state i to state j within t time units.

The $p^c_{ij}(t)$ must be clearly distinguished from the known p^c_{ij}. Whereas p^c_{ij} indicates the probability that a client of class c will eventually move from state i to state j, $p^c_{ij}(t)$ indicates the probability that a client of class c will move from state i to state j within time t.

The $p^c_{ij}(t)$ can be calculated via the *uniformization method* [25] using the given p^c_{ij} as well as the given leaving rates v^c_j as follows:

$$\overline{p}^c_{ij} = \begin{cases} \frac{v^c_i}{v^c} p^c_{ij}, j \neq i \\ 1 - \frac{v^c_i}{v^c}, j = i \end{cases} \tag{5}$$

$$p^c_{ij}(t) = \sum_{k=0}^{\infty} e^{-vt} \frac{(vt)^k}{k!} \overline{p}^{c(k)}_{ij} \tag{6}$$

$$\begin{aligned} &\text{with } \overline{p}^{c(k)}_{ij} = \sum_{s=0}^{S_c-1} \overline{p}^{c(k-1)}_{is} \overline{p}^c_{sj} \\ &\overline{p}^{c(0)}_{ss} = 1 \\ &\overline{p}^{c(0)}_{sj} = 0 \text{ for } j \neq s \\ &v^c \geq v^c_i \ \forall \ i = 0, \cdots S_c - 1 \end{aligned}$$

Next, we use the $p^c_{ij}(t)$ to calculate the matrix $\overline{M}(t)$ giving the expected values for the numbers of clients in the individual states of the different classes.

$$\overline{M}(t) = \begin{pmatrix} \overline{m}^t_{0,0} & \cdots & \overline{m}^t_{0,S-1} \\ \cdots & \cdots & \cdots \\ \overline{m}^t_{C-1,0} & \cdots & \overline{m}^t_{C-1,S-1} \end{pmatrix} \tag{7}$$

$$\begin{aligned} &\text{with } \overline{m}^t_{ci} = \sum_{k=0}^{S_c-1} m^{t_{start}}_{ck} p^c_{ki}(t) \\ &m^{t_{start}}_{ci} \in M(t_{start}) \\ &t \in \{0, l_r, 2l_r, \cdots, (W-1)l_r\} \end{aligned}$$

Finally, we obtain $M(t)$ by rounding the values of the elements of $\overline{M}(t)$.

3.5 The determination of $ub(t)$ in prognosis module 2

Based on the matrix $M(t)$, we can calculate the upper bound $ub(t)$ according to equation (2). To achieve this, we first introduce a random variable N^t representing the amount of data the admitted clients and the new client request on average in the round starting at time $t_{start} + t$. We further assume that the following relationship between N^t and T^t_s holds:

$$T^t_s = \frac{N^t}{capacity} \tag{8}$$

Equation (8) expresses a linear relationship between the amount of data the server has to transmit to the client and the time it takes the server to execute this transmission. The parameter *capacity* is a constant that gives the data amount the server is able to deliver in a single round[1]. To calculate $ub(t)$, we use the *Chernov Inequality* [15] which has the form

$$P(Y \geq x) \leq \inf_{\theta \geq 0} e^{(-\theta x)} G_Y(\theta) \tag{9}$$

The Chernov inequality has been proven to be a good upper bound for many practical problems, such as in [19]. In this inequality, $G_Y(\theta)$ is the so-called *Moment Generating Function* of Y which is defined as [15]:

$$G_Y(\theta) = \int_{-\infty}^{\infty} e^{\theta y} f_Y(y) dy \tag{10}$$

with $f_Y(y)$: density function of random variable Y

[1] We abstract from resource specific parameters, like buffer size, disk seek and transfer time, and rotational latency.

Between the Moment Generating Function G_Y and the so-called *Laplace Transformation* $F^*_Y(\theta)$ the following relationship exists [15]:

$$G_Y(\theta) = F^*_Y(-\theta) \tag{11}$$

Inequality (9) applied to equation (2), leads to

$$ub(t) = \inf_{\theta \geq 0}(e^{-\theta l_r} G_{T^t_s}(\theta)) \tag{12}$$

$$with \quad inf_{\theta \geq 0}(e^{-\theta l_r} G_{T^t_s}(\theta)) \leq P(T^t_s > l_r)$$

Thus, to find $ub(t)$, we first have to solve the problem of how to determine $G_{T^t_s}(\theta)$, i.e., the Moment Generating Function of T^t_s. To achieve this, we proceed as follows: First, we express T^t_s in terms of known random variables. Then, we use this expression to calculate the Laplace Transformation of T^t_s and derive from this Laplace Transformation the desired Moment Generating Function using equation (11).

For the first step, we observe that the random variable N^t for the total data consumption can be computed from the matrix $M(t)$ and the known random variables N_{ci} for the amount of data a client on average requests in one round if he is in state i of class c, as follows:

$$N^t = \sum_{c=0}^{C-1} \sum_{i=0}^{S_c-1} \sum_{k=1}^{m^t_{ci}} N_{ci} \tag{13}$$

Thus, according to equation (8) T^t_s can be rewritten as follows:

$$T^t_s = \frac{1}{capacity} \sum_{c=0}^{C-1} \sum_{i=0}^{S_c-1} \sum_{k=1}^{m^t_{ci}} N_{ci} \tag{14}$$

Next, we define a random variable M_{ci} as follows:

$$M_{ci} = \frac{1}{capacity} N_{ci} \tag{15}$$

We have to distinguish between the idle and the active states of the clients. Within the latter, M_{ci} is rectangularly distributed, and the parameters u'_{ci} and v'_{ci} corresponding to the minimum and maximum of the rectangular distribution of M_{ci} as well as the distribution function can directly be deduced from those of N_{ci}:

$$P(M_{ci} \leq x) = \begin{cases} 0, & -\infty < x \leq u'_{ci} \\ \frac{x - u'_{ci}}{v'_{ci} - u'_{ci}} & u'_{ci} < x < v'_{ci} \\ 1, & v'_{ci} \leq x \end{cases} \tag{16}$$

$$\text{with } u'_{cs} = \frac{u_{cs}}{capacity},\ v'_{cs} = \frac{v_{cs}}{capacity}$$

For the idle states, the random variable M_{ci} has the value zero. Therefore, we can rewrite T^t_s in terms of known random variables as follows:

$$T^t_s = \sum_{c=0}^{C-1} \sum_{i=0}^{S_c-1} \sum_{k=1}^{m^t_{ci}} M_{ci} \tag{17}$$

This is the form of T^t_s that we can use to calculate the Laplace Transformation of T^t_s. This calculation is based on the following observations [15]: The density function of a sum of random variables is equal to the convolution of the density functions of the individual terms of the sum. The Laplace Transformation of a convolution is equal to the product of the Laplace Transformations of the individual terms of the convolution, where the convolution is defined as [15]:

$$\begin{aligned} &conv(y) = \\ &\int_{-\infty}^{\infty} f_{X_1}(y - x_2 - \cdots - x_n) * \cdots \\ &\cdots * f_{X_2}(x_2) * \cdots * f_{X_n}(x_n) dx_n \\ &\text{with } f_{X_i} : \text{density function of term } i \\ &n : \text{number of terms} \end{aligned}$$

Using equation (17), we can therefore calculate the Laplace Transformation $F^*_{T^t_s}(\theta)$ of T^t_s as follows:

$$\begin{array}{lll} F^*_{T^t_s}(\theta) = & & \\ F^*_{M_{0,0}}(\theta)^{m^t_{0,0}} & \times \cdots \times & F^*_{M_{0,S_c-1}}(\theta)^{m^t_{0,S_c-1}} \\ & \times \cdots \times & \\ F^*_{M_{C-1,0}}(\theta)^{m^t_{C-1,0}} & \times \cdots \times & F^*_{M_{C-1,S_c-1}}(\theta)^{m^t_{C-1,S_c-1}} \end{array} \tag{18}$$

In this context $F^*_{M_{c,i}}(\theta)$ represents the Laplace Transformation of the random variable M_{ci}. As the exponents of the individual factors are the elements of the known matrix $M(t)$, the search for $F^*_{T^t_s}(\theta)$ can be reduced to determining $F^*_{M_{c,i}}(\theta)$, which has the following form:

$$\begin{array}{lll} F^*_{M_{c,i}}(\theta) = & \frac{e^{-\theta u'_{ci}} - e^{-\theta v'_{ci}}}{\theta(v'_{ci} - u'_{ci})} & \text{(active states)} \\ F^*_{M_{c,i}}(\theta) = & 1 & \text{(idle states)} \end{array} \tag{19}$$

Having determined $F^*_{T^t_s}(\theta)$ the Moment Generating Function $G_{T^t_s}(\theta)$ can be calculated using equation (11). Finally, we compute $ub(t)$ of equation (12) by a polynomial approximation.

4. SIMULATIONS

4.1 Simulation scenarios

For evaluation purposes, we model preorchestrated interactive multimedia presentations with highly varying resource requirements. As components of the presentations, we use three different media formats, namely the video formats MPEG-1 and MPEG-2 and the audio format MP3 [1]. We assume that an MPEG-1 video has a mean data rate of 1.5 Mbit/s. For an MPEG-2 encoded video, we assume a mean data rate of 4.5 Mbit/s. Both video formats are considered to be hardware encoded with a fixed IPB-pattern. Therefore, we assume the rates within a video stream to be rectangularly distributed with maximum or minimum values 20 per cent above or below the average data rate. We assume audio streams with CD-quality to be MPEG-1 encoded. For MP3 encoded audio streams, we use a data rate of 130 Kbit/s. Since text documents, images and VRML scenes require a relatively low amount of data in comparison with time-dependent media like audio and video, we ignore them within our simulations. The resource requirements of the different media formats are summarized in Table 1.

In our simulations, we consider three different application classes. The first one consists of 9 different states (see Figure 4). The values next to the arrows represent the time-

Media Format	Mean Data Rate	Distribution	Rate Deviation
MPEG-1	1.5 Mbit/s	rectangular	0.2
MPEG-2	4.5 Mbit/s	rectangular	0.2
MP3	130 Kbit/s	rectangular	0.2

Table 1: Resource Requirements

independent transition probabilities. Table 2 includes the media combinations, the mean data rates and the mean holding times of the users, within the single states of class 1.

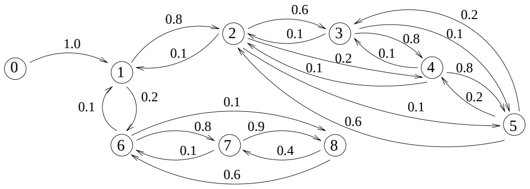

Figure 4: Class 1: States and Transitions

The second class represents a multimedia presentation consisting of 4 different states with extremely high variations in the requested data rates. The mean data rate in state 2, for example, is more than 80 times larger than that of states 0 and 3. The simulation parameters within the states of class 2 are summarized in Table 3. The time-independent transition probabilities are shown in Figure 5.

Finally the simulation parameters of the third class are summarized in Figure 5 and in Table 4, respectively. In this class which consists of 4 different states, the variations in the requested data rates are not as high as those in class 2. By default, an equal combination of these three classes is employed for simulation purpose.

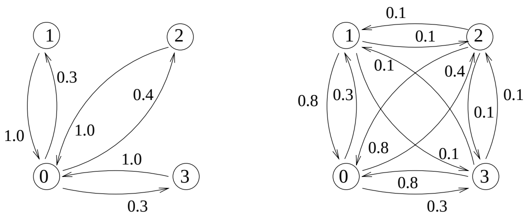

Figure 5: Class 2 (left side) and Class 3 (right side)

4.2 Experimental testbed

In our simulations, the time interval between the arrival of two successive clients in a given class is exponentially distributed. For each class, the parameter of the distribution is 0.05, i.e., every 20 seconds a new client asks for admission, on average. The presentation times of the clients are also exponentially distributed with a parameter of 0.003, i.e., the average presentation time is 300 seconds for each class. The server resources are set to 100000, that means, the server is able to deliver 100 Mbit/s. In all experiments, 3000 rounds are simulated. The single requests of the clients are scheduled according to the strategy Earliest Deadline First (EDF). Table 5 summarizes the default simulation parameters.

Parameter	Value
server resources	100000
number of rounds	3000
duration of a round	1 time unit
average arrival of new clients	20 time units
distribution of clients arrival	exponential
average duration of a presentation	300 time units
distribution of presentation duration	exponential

Table 5: Default Simulation Parameters

4.3 Experimental results

In this section, we present the simulation results we obtained by using the ACSP. We measure the server utilization and the ratio of requests served within their deadlines (*in-time-ratio*). Keeping the deadlines corresponds to *all* temporal QoS parameters, i.e., startup delays as well as intra- and intermedia constraints. This means that a startup request has the same priority as consecutive requests. For the simulations, we do not allow spatial QoS adaptations. Another goal of the simulations is to investigate if the ACSP leads to a stable system behavior, i.e., if the proposed model is able to recover from underload or overload periods. For the implementation of our simulations, we used the CSIM tool [17].

4.3.1 Experiment 1

First, we study the system behavior under variations of the time window W and the parameter p_{lim} (see Figures 6 and 7). For time windows of length 10, 20, 30 and 40 the system behavior becomes instable for small values of p_{lim}, i.e., the growing number of client requests cannot be handled by the server.

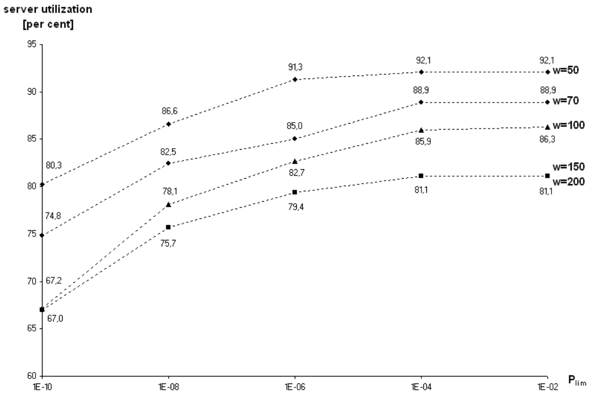

Figure 6: Server Util. for Variations of W and p_{lim}

For $W \geq 50$ the system becomes stable and it can be seen that, for a constant p_{lim}, a larger time window leads to a lower server utilization and to a better QoS level. This result can be explained as follows: For growing time windows W

State	Media Combinations	Mean Data Rates	Mean Holding Times
0	1 MP3 and 2 MPEG-1	3.13 Mbit/s	40
1	1 MP3	130 Kbit/s	30
2	1 MP3 and 1 MPEG-1	1.63 Mbit/s	20
3	1 MP3 and 2 MPEG-1	3.13 Mbit/s	100
4	2 MPEG-1	3 Mbit/s	80
5	1 MP3 and 1 MPEG-2	4.630 Mbit/s	100
6	1 MP3	130 Kbit/s	20
7	1 MP3	130 Kbit/s	100
8	1 MPEG-1	1.5 Mbit/s	100

Table 2: Class 1

State	Media Combinations	Mean Data Rates	Mean Holding Times
0	1 MP3	130 Kbit/s	30
1	1 MP3 and 1 MPEG-1	1.63 Mbit/s	200
2	1 MP3 and 2 MPEG-2 and 1 MPEG-1	10.63 Mbit/s	300
3	1 MP3	130 Kbit/s	100

Table 3: Class 2

the ACSP estimates the overload probability for a larger number of rounds as our algorithm looks further into the future. Thus, fewer clients are admitted to the service, the server utilization decreases and the rate of requests served within their deadlines increases.

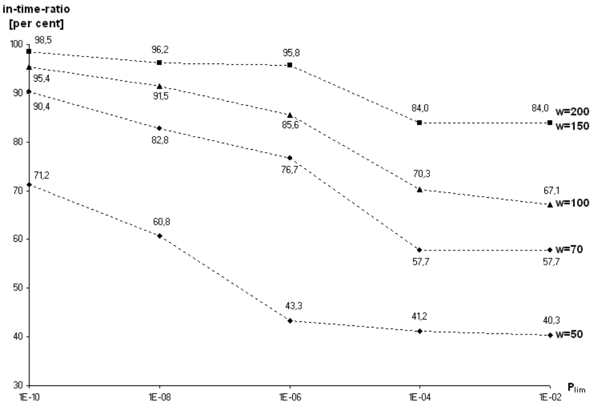

Figure 7: In-time-ratio for Variations of W and p_{lim}

Likewise we observe that, for a constant W, a higher value for p_{lim} leads to a lower server utilization and to a better QoS level. This correlation can be explained as follows: A low value for p_{lim} means that, for each round within the time window W, the estimated overload probability will be compared with this low value. In this case, it is more likely that the estimated overload probability exceeds p_{lim} and thus a newly arriving client is rejected than for a large p_{lim}. Therefore, a smaller value for p_{lim} means that less clients are admitted to the service, the server utilization decreases and the rate of requests served within their deadlines increases.

Furthermore, it can be seen that for $W = 200$ and $W = 150$ the simulations lead to the same values for the regarded variables, i.e., our application scenario converges into an equilibrium state for the given parameter values.

For $p_{lim} = 10^-8$ $(10^-8 = 1E - 08)$ and $W = 150$, more than 96 per cent of all requests are served within their deadlines and the average utilization has an acceptable value about 75 per cent. Thus, this parameter combination is a good choice in the given scenario. If an even higher in-time-ratio is needed $p_{lim} = 10^{-10}$ can be employed leading to about 98.5 per cent of requests served within their deadlines. However, this would most probably lead to an 8 per cent reduction of the server utilization. Accordingly, if the in-time-ratio is of low importance, lower values for W and/or higher values for p_{lim} could also be preferred.

4.3.2 Experiment 2

Now, we evaluate how the system reacts if the data rate variations in the simulations and the data rate variation assumed in the ACSP model differ. We separately consider the situations in which the maximum respectively minimum values are between 20 and 100 per cent above respectively below the average data rate. That means that we study higher data rate variations in the states of all classes than assumed in the ACSP.

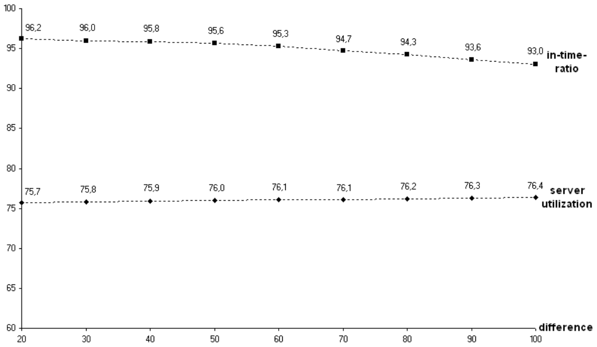

Figure 8: Behavior for differing Consumpt. Rates

The corresponding results for $p_{lim} = 10^{-8}$ and $W = 150$ in Figure 8 show that a higher variation leads to nearly the same result with respect to the server utilization and to a decreased value for the observed QoS level. These correlations can be explained as follows: A high variation in the requested data rate does not affect the average value in the long run. Therefore, the average server utilization does not vary, ei-

State	Media Combinations	Mean Data Rates	Mean Holding Times
0	1 MP3	130 Kbit/s	30
1	1 MP3 and 1 MPEG-1	1.63 Mbit/s	200
2	1 MP3 and 1 MPEG-1	1.63 Mbit/s	300
3	1 MP3 and 1 MPEG-1	1.63 Mbit/s	100

Table 4: Class 3

ther. On the other hand, a high variation in the requested data rate leads to a high amount of resource requests within single rounds. These, in turn, lead to violations of deadlines not only within the rounds they occur but also in subsequent rounds. Thus, the in-time-ratio decreases. Nevertheless, as the results show, the system behavior remains stable since the number of delayed requests are still strongly limited.

4.3.3 *Experiment 3*

Now, we assume that only clients of the second class are asking for admission, i.e., the server has to deal with excessively high variations in the requested data rates. The simulation results for $W = 100$, $W = 150$ and different values for p_{lim} are summarized in Figures 9 and 10.

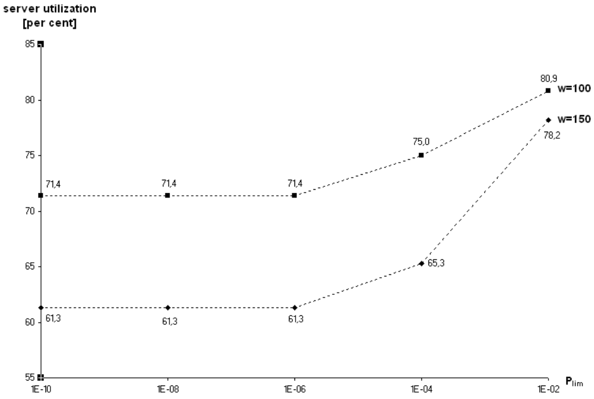

Figure 9: Server Utilization (only Class 2)

It can be observed that, in comparison with experiment 1, the server utilization as well as the in-time-ratio decrease. These results are plausible because of the following considerations: First, in the long run, the clients of class 2 reside in states with low mean data rates. In these periods the overall amount of data the server has to deliver is very low. Thus, the average server utilization is low, too, since our approach does not look for an optimal point in time to admit a client but only test the consequences for an immediate admission. Such an approach would also have drawbacks, such as to find another session that enables a better system utilization exactly until the resources of the new one will be needed. On the other hand, as shown in experiment 2, high data rate variations typically lead to violations of deadlines in subsequent rounds. As a result, the in-time-ratio decreases.

4.3.4 *Experiment 4*

Next, we evaluate how the system reacts with respect to changes of the parameter l_r (see Figure 11). In these simulations, p_{lim} was set to 10^{-8}, and we regarded a future time interval with a length of 150 time units. It can be seen that the average server utilization becomes better with growing values for l_r whereas the in-time-ratio decreases dramatically.

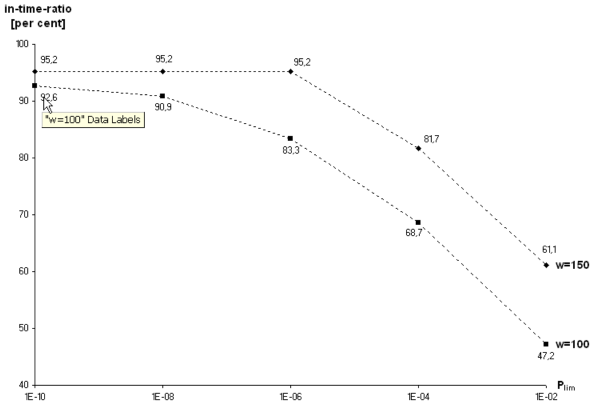

Figure 10: In-time-Ratio (only Class 2)

For $l_r = 10$, only about two third of all requests could be served in time. For $l_r \geq 11$, the system even became instable, i.e., the server could not deal with the growing resource claims. These results show that the granularity of the overload prediction is a critical parameter. Large values for l_r mean that the ACSP-predictions are based on the average client behavior and that peaks in the resource demands of the clients are not taken into account. Thus, large values for l_r correspond with a naive admission control mechanism regarding only the average behavior of the clients. This experiment clearly indicates that a less precise prediction in terms of the round length is not suitable for scenarios with high data rate variations.

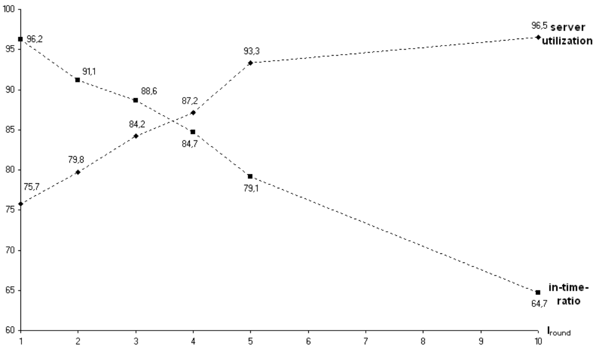

Figure 11: Behavior for Changes of l_r

4.3.5 *Experiment 5*

Finally, we study the system behavior under the assumption that the real data rate distributions are normal, exponential, erlang, and hyperexponential (Hyper). The system behaves well for normally distributed client requests. Thus, by employing the Central Limit theorem [7], even bursty client behavior is covered by our model. As described in Section 3, the ACSP assumes that the amounts of data the clients request are rectangularly distributed. The simulation results shown in Table 6 are very similar to those obtained in experiment 1 and show that our model is very robust against differences between the assumed and the real data rate distributions.

We summarize our simulation results as follows: For applications with highly varying resource requirements a precise prediction in terms of studying fine-grained rounds is required. Instable system behavior could only be observed for very small values for the parameter W and for large values of l_r. In the given environment, the time window should be set to a value of $W = 150$ and the parameter l_r should be set to a value of 1 to obtain a good system behavior. With p_{lim} set to a value of 10^{-8}, we achieved a very good QoS and a high server utilization, in most cases. We showed that the ACSP mechanism is even applicable if the real requested data rate differs from the assumed one in terms of mean variation and generating distribution function. In scenarios with unexpected behavior, the parameter p_{lim} should be set lower than in scenarios with more predictable behavior.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented an admission control scheme that targets at the highly varying resource requirements of multimedia sessions. With our session-based approach, we are able to achieve continuous presentations and to reduce startup latency. This is of special importance for interactive applications with frequent media switches.

We model application classes using Continous Time Markov Chains (CTMCs) and stochastically predict the resource usage within a future time interval. Simulation results show that a high server utilization as well as a good Quality of Service are achieved.

Future work will be focused on the following aspects: First, we will consider specific user profiles. More precise information on user behavior, like their preferences for a specific content, enable a more precise parameter setting within the ACSP. Another aspect is the integration of discrete data requests (mixed workload). Especially VRML scenes seem to be interesting in our context since, on the one hand, no 'real' time-constraints are given, but, on the other hand, in case of high system load, the delivery of such discrete data may be very slow. This leads to high delays especially at the start of a VRML presentation because at this point in time a high amount of data is typically requested. A further topic is to use other mathematical theories as a basis for our model. In this context new stochastical bounds and distributions will be studied.

6. ACKNOWLEDGEMENTS

We would like to thank Aris Ouksel for his many helpful comments.

7. REFERENCES

[1] Audio compression MP3. http://www.iocon.com/das/.

[2] K. Aberer and S. Hollfelder. Resource prediction and admission control for interactive video browsing scenarios using application semantics. In *Proc. of Int. Conf. on Data Semantics - 8 (DS-8), Semantic Issues in Multimedia Systems, IFIP TC-2 Working Conference*, pages 27–46, January 1999.

[3] E. L. Abram-Profeta and K. G. Shin. Providing unrestricted VCR functions in multicast video-on-demand servers. In *Proc. of Int. Conf. on Multimedia Computing and Systems (ICMCS)*, pages 66–75, June/July 1998.

[4] N. H. Balkir and G. Ozsoyoglu. Delivering presentations from multimedia servers. *VLDB Journal. Special Issue on Multimedia Databases*, pages 297–307, December 1998.

[5] H.-J. Chen, A. Krishnamurthy, T. D. C. Little, and D. Venkatesch. A scalable video-on-demand service for the provision of VCR-like functions. In *Proc. of Int. Conference on Multimedia Computing and Systems*, pages 65–72, May 1995.

[6] M.-S. Chen, D. D. Kandlur, and P. S. Yu. Support for fully interactive playout in a disk-array-based video server. In *ACM Multimedia*, 1994.

[7] H. Cramer and M. Leadbetter. *Stationary and Related Stochastic Processes*. Wiley, 1967.

[8] J. K. Dey, S. Subhabrata, J. F. Kurose, and J. D. Salehi. Playback restart in interactive streaming video applications. In *IEEE Conference on Multimedia Computing and Systems*, pages 458–465, June 1997.

[9] J. K. Dey-Sircar, J. D. Salehi, J. F. Kurose, and D. Towsley. Providing VCR capabilities in large-scale video servers. In *ACM Multimedia*, pages 25–32, 1994.

[10] M. L. Escobar-Molano and S. Ghandeharizadeh. On coordinated diplay of structured video. *IEEE Multimedia Systems*, 4(3):62–75, July-September 1997.

[11] M. Friedrich, S. Hollfelder, and K. Aberer. Stochastic resource prediction and admission for interactive sessions on multimedia servers. GMD Technical Report 50, GMD, Sankt Augustin, Germany, March 1999.

[12] C. Gopal and J. F. Buford. Delivering hypermedia sessions from a continuous media server. In S. M. Chung, editor, *Multimedia Information Storage and Management*, pages pp. 209–235. Kluwer Academic Publishers, 1996.

[13] S. Hollfelder and K. Aberer. An admission control framework for applications with variable consumption rates in client-pull architectures. In A. D. Sushil Jajodia, M. Tamer Özsu, editor, *Proc. of Int. Workshop on Multimedia Information Systems MIS'98*, pages 82–97. Springer LNCS, September 1998.

[14] S. Hollfelder, V. Oria, and T. Özsu. Mining user behavior for resource prediction in interactive electronic malls. In *Int. Conference on Multimedia and Expo (ICME)*, July/August 2000. will appear.

[15] L. Kleinrock. *Queueing Systems, Volume I: Theory*. Wiley, 1975.

[16] C. Ludmila and P. Phaal. Session based admission control: A mechanism for improving the performance of an overloaded web server. HP Labs Technical Reports, External HPL-98-119, 980612, Hewlett Packard, June 1998.

[17] Mesquite Software, Inc. *CSIM17 Users' Guide*, 1994.

[18] G. Nerjes, P. Muth, and G. Weikum. Stochastic performance guarantees for mixed workloads in a multimedia information system. In *Proc. of the IEEE International Workshop on Research Issues in Data Engineering (RIDE'97)*, April 1997.

[19] G. Nerjes, P. Muth, and G. Weikum. Stochastic service guarantees for continuous data on multi-zone disks. In

Distr.	**Rect.**	**Exp.**	**Normal**	**Hyper**	**Erlang**
serv. utiliz.	75.7%	78.9%	78.0%	76.2%	76.4%
in-time-ratio	96.2%	91.4%	93.4%	94.9%	95.6%

Table 6: Results for diff. Data Rate Distributions

Proc. of the Symposium on Principles of Database Systems (PODS'97), pages 154–160, May 1997.

[20] B. Özden, R. Rastogi, A. Silberschatz, and P. S. Narayanan. The Fellini multimedia storage server. In S. M. Chung, editor, *Multimedia Information Storage and Management*. Kluwer Academic Publishers, 1996.

[21] T. Plagemann and V. Goebel. Analysis of quality-of-service in a wide-area interactive distance learning system. *Telecommunication Systems Journal,* 11(1-2):139–160, 1999.

[22] N. Reddy. Improving latency in interactive video server. In *Proc. of SPIE Multimedia Computing and Networking Conference*, pages 108–112, February 1997.

[23] S. S. Roa, H. M. Vin, and A. Tarafdar. Comparative evaluation of server-push and client-pull architectures for multimedia servers. In *Proc. of Nossdav 96*, pages 45–48, 1996.

[24] P. J. Shenoy and H. M. Vin. Efficient support for interactive operation in multi-resolution video servers. *Multimedia Systems*, 7(3):241–253, July 1999.

[25] H. C. Tijms. *Stochastic Models. An Algorithmic Approach*. Wiley series in probability and mathematical statistics. Wiley, 1994.

[26] H. M. Vin, A. Goyal, A. Goyal, and P. Goyal. An observation-based admission control algorithm for multimedia servers. In *Proc. of the First IEEE International Conference on Multimedia Computing and Systems (ICMCS),* pages 234–243, May 1994.

[27] H. M. Vin, A. Goyal, and P. Goyal. Algorithms for designing large-scale multimedia servers. *Computer Communications,* March 1995.

[28] H. M. Vin, P. Goyal, A. Goyal, and A. Goyal. A statistical admission control algorithm for multimedia servers. In *Proc. of the ACM Multimedia,* pages 33–40, October 1994.

[29] W3C. Synchronized media integration language (SMIL), Boston Specification. http://www.w3.org/TR/smil-boston, February 2000.

[30] M. Y.Y.Leung, J. C. Lui, and L. Golubchik. Buffer and I/O resource pre-alocation for implementing batching and buffering techniques for video-on-demand systems. In *Proc. of Int. Conf. on Data Engineering (ICDE),* pages 344–353, 1997.

[31] W. Zhao and S. K. Tripathi. A resource reservation scheme for synchronized distributed multimedia sessions. *Multimedia Tools and Applications*, 7(1/2):133–146, July 1998.