Resource Prediction and Admission Control
For Interactive Multimedia Presentations

Expert System
Prof. Glenn Shafer
Fall 2001
Ahmed Gomaa
I Introduction:
Digital libraries are concerned with the creation and management of information sources, the movement of information across networks and the effective use of this information by a wide range of users. A digital library is a vast collection of objects that are of multimedia nature, e.g., text, video, images, and audio. Users wishing to access the digital library objects may possess varying capabilities, preferences, domain expertise, and may use different information appliances. Facilitating access to those complex multimedia digital library objects that fit to the users' requirements is known as universal access [1].

Quality-of-Service (QoS) has been considered as one of the essential requirements for providing distributed multimedia services. One of the objectives of universal access objective is to render the Multimedia objects at low latency rates because the startup latency is significant and may negatively impact performance and QoS. Admission control mechanisms are used to limit the number of clients to be served in order to meet each client’s Quality of Service (QoS) requirements and to achieve high resource utilization. [2]. In this paper I will review the literature of mechanisms for predicting available resources and controlling admission for interactive multimedia presentations.

II Problem statement
In this paper I will present a review of literature on the resource prediction and admission control mechanisms for interactive multimedia presentations.

III Motivation
My research outside this course is focused on modeling customized multimedia presentations based on user profiles as a step toward universal access. In this paper I will try to understand what is being done in predicting resources and controlling admission mechanisms for multimedia presentation to enhance user experience and to increase the quality of service.

The further contents of the proposal are structured as follows: Section IV introduces the terminologies used and the related concepts. Section V presents summaries four admission control mechanism papers. And section VI discuss and categorize the papers reviewed.

IV Terminologies and concepts:

Multimedia:
Multimedia can be distinguished from traditional motion pictures or movies both by the scale of the production (multimedia is usually smaller and less expensive) and by the possibility of audience interactivity or involvement (in which case, it is usually called interactive multimedia). Interactive elements can include: voice commands, mouse manipulation, text entry, touch screen, video capture of the user, or live participation (in live presentations).
Digital Library:
There are many definitions of a "digital library." Terms such as "electronic library" and "virtual library" are often used synonymously. The elements that have been identified as common to these definitions [3] are:
The digital library is not a single entity;
The digital library requires technology to link the resources of many resources;
The linkages between the many digital libraries and information services are transparent to the end users;
Universal access to digital libraries and information services is a goal;
Digital library collections are not limited to document surrogates: they extend to digital artifacts that cannot be represented or distributed in printed formats.

Digital Library object:
The digital library object may be considered as a large multimedia presentation which include different multimedia components as videos, audios, texts and images
The digital library object is customized based on the user capabilities (i.e. culture and languages, discipline, technical expertise, physical capabilities – handicapped) and characteristics (Different information appliances, different preferences, mobility).

Universal access:
The Universal Access objective is to facilitate access to digital library multimedia objects according to the different capabilities and characteristics of the end-users

Bandwidth:
Bandwidth is the width of a band of electromagnetic frequencies. Generally speaking, bandwidth is directly proportional to the amount of data transmitted or received per unit time. For example, it takes more bandwidth to download a photograph in one second than it takes to download a page of text in one second. Large sound files, computer programs, and animated videos require still more bandwidth for acceptable system performance. Virtual reality (VR) and full-length three-dimensional audio/visual presentations require the most bandwidth of all.

QoS:
Short for Quality of Service, a networking term that specifies a guaranteed throughput level. The throughput is the amount of data transferred from one place to another or processed in a specified amount of time. Data transfer rates for disk drives and networks are measured in terms of throughput. Typically, throughputs are measured in kbps, Mbps and Gbps

Admission control:
Admission control is an essential element to achieve guaranteed services. For distributed multimedia communications systems, each resource along the path(s) between source(s) and sink(s) must monitor its availability. The admission control mechanisms are used to
limit the number of clients to be served in order to meet each client’s Quality of Service (QoS) requirements and to achieve high resource utilization. [2]

V Paper summaries:

V_I Stochastic Resource Prediction and Admission for Interactive Sessions on Multimedia Servers by Matthias Friedrich, Silvia Hollfelder and Karl Aberer

The authors describe a stochastic admission control strategy for multimedia presentation scenarios characterized by high variations in the data rates caused by interactive access. Their 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 can be flexibly adapted by choosing the length of the rounds, which will be statistically analyzed one by one.

Modelling of interactive client sessions

For session modelling, the authors assign to each client an application class \( c = 0 \ldots c-1 \) which is further differentiated into a finite number of states \( i = 0 \ldots Sc-1 \) representing the different presentation modes. Each client is allowed to switch between the states of his class. At each point in time, he 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 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 are either identical or similar.

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 memory-less. 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 \( p_{ij}^c \) no matter how he attained state \( i \). For the \( p_{ij}^c \), subsequently called time-independent transition probabilities, the following equations hold:

\[
\sum_{j=0, j \neq i}^{Sc-1} p_{ij}^c = 1.
\]

If a client moves to state of class \( c \), he stays there for a time interval being exponentially distributed with parameter \( v_j^c \) (subsequently call the leaving rate). Independent of how he reached state \( j \). The implementation of the ACSP requires the identification of the classes and their corresponding states, as well as the determination of the parameters \( p_{ij}^c \) respectively \( v_j^c \). The
latter could possibly be carried out by means of observation. In this paper, the authors assume that the time-independent transition probabilities as well as the leaving rates are known.

**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 time window \( W \) starting at time \( t_{\text{start}} \). The authors further divide \( W \) into rounds which all have the same length \( l_{\text{round}} \). The statistical analysis in the prediction is performed for each round separately.

![Time Window for resource prediction](image)

In this analysis, the random variable \( T_{\text{service}}^t \) is the time the server needs in the round starting at time \( t_{\text{start}} + t \) for serving the data requested by all admitted clients on average. Then an overload is the probability that \( T_{\text{service}}^t \) is greater than \( l_{\text{round}} \). 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:

\[
ub(t) \geq P(T_{\text{service}}^t > l_{\text{round}})
\]

The calculation of \( ub(t) \) is the essential part in the ACSP. It is performed in two separate steps. 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 \( ub(t) \).
The above figure 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 $t_{\text{start}}$, 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 $p_{\text{lim}}$ representing the maximal overload probability the system is willing to accept without rejecting the client. The value $p_{\text{lim}}$ is a configuration parameter showing how optimistic or pessimistic the admission control proceeds. High values for $p_{\text{lim}}$ lead to a larger number of admitted clients and therefore to more frequent overload situations. If $ub(t)$ is greater than $p_{\text{lim}}$, the client is rejected, otherwise the time parameter $t$ is increased by $l_{\text{round}}$, the matrix $M(t + l_{\text{round}})$ and the upper bound $ub(t + l_{\text{round}})$ 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 $p_{\text{lim}}$, the new client is accepted to service. Otherwise, if $ub(t)$ exceeds $p_{\text{lim}}$ for a single round, the client is rejected.

V. II Use of statistical methods to reduce delays for media playback buffering.

By Prathima Agrawal, Jyh-Cheng Chen, and Cormac J. Sreenan

This paper works with an alternative predictive approach, which records historical information and uses it to make short-term predictions about network delay, with the aim of not reacting too quickly to short-lived variations. This allows an application-controlled trade-off of packet lateness against buffering delay, suitable for applications which demand low delay but can tolerate or conceal a small amount of late packets. This paper builds on previous work by proposing aging techniques to improve delay predictions, evaluating performance using trace-driven simulation.

In the basic Algorithm, the quantity of interest is the total end-to-end delay ($ted$): the cumulative delay suffered by packets in the network and the playback buffer. In the predictive approach, a trade-off is performed of packet lateness against the total end-to-end delay suffered by packets, by taking application-supplied parameters and applying them to probabilistic data describing packet delay distributions.

The scheme operates dynamically by updating its probability delay distributions and actual buffer sizes over time in response to the observed behavior of the network. A stream is characterized by the maximum acceptable delay (mad) it can suffer, and the maximum late packets (mlp) percentage it can tolerate. For each stream the authors construct a packet delay distribution (PDD): an estimate of the probable delays suffered by packets in the network over a time window. This PDD may draw on existing traffic conditions, history information or any negotiated service characteristics to derive initial estimates for delay bounds and distributions.

In other words, the authors consider how to calculate the end-to-end delay given a PDD. For any value for ($ted$) packets with a network delay above this value will be declared late. Hence $(1 – cdf(ted))$ packets will be declared late, where $cdf$ is the cumulative distribution function on the PDD. Now we can control the behavior by choosing a value
of ted so that either mlp or ted is minimized (best visualized by moving ted on a PDD curve to the right or left, respectively).

There is a need for a suitable statistical method, which can predict the delays by analyzing historical, and current information and monitor, maintain, update, and store statistical trends of network delays. The approach advocated here is to store and track network delay trends using a measured histogram to approximate the PDD curve. The histogram bins store frequency of observed delays, where bin width means the range of network delays grouped together to represent one value in the histogram. Naturally, for the same delay range, the number of bins increases proportionally as the width of the bins is decreased. Since network characteristics vary with time, these statistical trends vary, and current information is necessary for prediction to be accurate, that is, it requires updating each bin to diminish the effect of older samples. This is called aging.

The process of aging involves three parameters. The frequency determines how often the aging function should be invoked to update the histogram. Aging coefficient, defined as a number between 0 and 1. The manner in which this coefficient is used to perform scaling is the key difference between the three algorithms examined. In particular, for each algorithm, the authors define the relationship between coefficient and the actual aging factor used to scale each bin. In each case, newly arriving samples are added to the appropriate bin using a constant weight.

V_III Providing VCR capabilities in large-scale video servers By Jayanta K. Dey-Sircar, James D. Salehi, James F. Kurose, and Don Towsley

The basic service provided by a video server is the smooth playback of video streams. In digital libraries, a user would like to be able to browse quickly through a video clip. Or skip fact to the part that interests him. The bandwidth requirement of a stored video stream, known as priori, is used to construct a bandwidth allocation schedule. This allows resource reservation at a finer granularity than the peak allocation method, potentially admitting more users at a given QoS.

In order to accommodate FF/Rew capabilities, the paper explore 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. The paper goals are 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. Second, it defines several QoS metrics to characterize effective FF/Rew, and apply these metrics to evaluate the effective FF/Rew service mechanism.

Mechanisms supporting effective FF/Rew

Users spend most of their time in playback mode while watching a video sequence; and the duration of time spent in FWRew mode is relatively small.
The following mechanism is the basis for providing effective FF/Rew service. Since playback is the primary functionality in video-on-demand, so every user will 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,

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 FWRew 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 FWRew service. We call this the Release Bandwidth Delay Scheme (RD), since users release their playback bandwidth at the time they make a FWRew request. The potential advantage of RD over HD is reduction in FWRew 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.

The paper shows 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. 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.
The paper presents the design and implementation of a client-server distributed multimedia database environment that can be used to support large digital libraries. System architecture and design are described. Server functionalities, including client scheduling, data buffering and admission control, are investigated. A client request can only be admitted if both the quality-of-service (QoS) requirements from the client and the upper bound on total buffer consumption at the server are maintained.

In this paper, focus on the design of the entire system and the server functionalities. The design and implementation of a client-server distributed multimedia database environment that can be used to support large digital libraries will be presented. We will describe the details on the system architecture and design. Protocols to achieve server functionalities, including client scheduling, data buffering and admission control, are investigated. A client request can only be admitted if both the quality-of-service (QoS) requirements from the client and the upper bound on total buffer consumption at the server are reserved. I will present the admission control mechanism presented in the NetMedia system.

**Admission Control**
Guaranteeing continuous playback for each client request requires that the retrieval time of data in a round does not exceed the playback time of the data retrieved in that round. This necessitates the server to employ admission control to decide whether a new client request can be admitted without violating the deadline requirements of the current client requests. The formulation of an admission control strategy would depend on the QoS requirements of each client and the shared resources at the server site. Different clients would have varying QoS parameters. The multimedia server should exploit this variation while servicing the client requests.

Let $\alpha$ embody the QoS associated with each client request. A strict servicing policy would set $\alpha$ to 1, thereby requiring the server to meet all the deadline requests of the client. A flexible request would have a value in the range $[0,1]$ suggesting that the server can drop some segments if it cannot meet a deadline requirement. In such a case, the server can also delay the transmission of segments so long as the new deadline requirements are within the QoS specifications set by the client. Thus, the fraction of data, such as frames in a video clip, that can be dropped or time through which the server can delay the transmission of segments is set by $\alpha$.

Another important parameter in admission control is the buffer limitation. Servicing a new request would mean increase in the buffer requirements. Admitting a new request should not violate the deadline requirements of the current requests being serviced. A simple strategy would be to make sure that the total buffer consumption in next $m$ successive intervals does not exceed the maximum buffer size $B_{\text{max}}$. 

The admission control policy is a greedy policy that calculates the buffer requirement for each new request and then admits the request if the total buffer consumption in next $m$ successive intervals does not exceed the maximum buffer size $B_{max}$ and the QoS specifications of all the current clients requests are not violated. The goal of the admission control algorithm, under such conditions, would be to maximize the number of client requests admitted. The greedy admission control algorithm takes in as input a request ({$R_1$, $R_2$, $R_3$}, $R_{n+1}$) that consists of a set of current requests being serviced and the new request $R_{n+1}$ that needs to be admitted, along with the QoS parameter for each client request, and produces as output the decision whether to admit the new request or not.

![Greedy admission algorithm]

Using buffer sharing does increase the number of client requests admitted over non-shared buffer schemes. This is because more clients can be admitted without overflowing the buffer.
VI Discussion:

VI-I Usage of application semantics.

A multimedia session consists of the presentation of multiple multimedia objects that have to be synchronized in temporal order. The temporal order of the presentation is known at admission time. The goal of the approach is to determine a starting point for the presentation for which all required resources (i.e., network and end system) are available. The basic reservation model does not consider user interactions. Friedrich et al. present an admission control scheme that targets at the highly varying resource requirements of multimedia sessions. Their approach achieves continuous presentations and reduces startup latency. They model application classes using Continuous Time Markov Chains (CTMCs) and stochastically predict the resource usage within a future time interval. Simulation results show that high server utilization as well as a good Quality of Service is achieved.

VI-II Inspection of the past system behavior.

In a Digital Library application, which consists of a sequence of web server accesses. The goal is to achieve a fair completion guarantee for any accepted session. This approach does not adequately reflect time-constraints of multimedia sessions. Agrawal et al. propose to stochastically predict network delays by recording historical information in order to reduce buffering delays [5].

VI-III A priori reservation.

To guarantee a given QoS, worst-case assumptions about the required data rate can be made. Obviously, in case of reservation of this high data rate, server resources are wasted, and the number of clients that can be served in parallel decreases. Dey-Sircar et al. [6] give stochastic guarantees by means of reserving separate server bandwidth for VCR-interactions. The drawback of their work is that they assume interactions to occur rarely.

VI-IV Re-admission at interaction points.

A straightforward way to support interactive client sessions is to initiate for each user interaction a new admission request at the server, as described in Gollapudi and Zhang [7]. This leads to high startup latency in case of high system load.
References:


