Multimedia Delivery Over Deadline-Based Networks

A thesis presented
by

Jie Wu

to
The Department of Computer Science
in partial fulfillment of the requirements
for the degree of
Master of Science
in the subject of

Computer Science

The University of Manitoba
Winnipeg, Manitoba
December 2006

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****
COPYRIGHT PERMISSION

Multimedia Delivery Over Deadline-Based Networks

by

Jie Wu

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

of

Master of Science

Thesis advisor

Yanni Ellen Liu

Author

Jie Wu

Multimedia Delivery Over Deadline-Based Networks

Abstract

Increasing demand to transmit real-time data over packet-switched networks calls for quality-of-service (QoS) support from the underlying network. Deadline-based networks were developed for this purpose. A deadline-based network is a priority network in which packets with more urgent deadlines are delivered with a higher priority at network routers. Such deadline-based scheduling provides a better service to real-time data delivery than first-come-first-served (FCFS), which is the scheduling discipline used on the classical Internet. In the literature, effective and efficient support to multimedia delivery in deadline-based networks has not been studied. In this research, effective and efficient support to multimedia data delivery in deadline-based networks is investigated. Resource management schemes, especially end-system based schemes, are developed for this purpose. These include two admission control algorithms and two differential deadline assignment schemes. When network load is heavy, congestion may occur. Multimedia data may miss their delivery deadlines due to excessive queueing delays and high packet loss ratios. This would directly affect the playback quality at the application layer. To control the level of load and improve performance, two end-system based admission control algorithms are developed. In deadline-based networks, application deadline information is translated into packet deadlines, which are carried in packets and used by routers for channel scheduling. Based on the characteristics of video encoding schemes, a frame-type differential deadline assignment scheme is developed to provide better support to real-time video delivery. Finally, a differential deadline assignment scheme for multimedia is introduced to handle the scenarios when real-time multimedia traffic is sharing network channels with other real-time non-multimedia traffic. The performance of these schemes is evaluated using simulation. The proposed schemes are shown to improve the performance of multimedia delivery over deadline-based networks.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank all the people who have supported me along the way. First, I would like to express my sincere thanks to my supervisor Dr. Yanni Ellen Liu. She taught me that basic principles are the foundation of research work. During the time I was working with her, she provided a lot of advice and guidance to my thesis work. I really appreciate her inspirations.

I also would like to thank all professors and staff in Computer Science Department for their help and valuable comments and suggestions.

Last but not least, I would like to thank my parents and my significant other. They are the people who support me unconditionally throughout the time I was doing my thesis work. I would like to share my happiness with them.

This thesis is dedicated to my parents and my husband.

Chapter 1

Introduction

Multimedia delivery over packet-switched networks is becoming more and more popular. Typical multimedia content are audio and video. Examples of multimedia applications include video-on-demand, voice over IP, Internet radio broadcasting, video conferencing, multi-player interactive online games, remote medical surgery, and high definition TV. Some of these applications, for example, download-and-then-display movie trailer viewing, do not have stringent requirement on network delivery of multimedia data and are outside the scope of this study. Other multimedia applications, on the other hand, may have stringent quality of service (QoS) requirements on the underlying network, which are referred to as real-time multimedia applications and are of interest to this study. Among real-time multimedia applications, three types of applications can be categorized: streaming stored multimedia, streaming live multimedia, and real-time interactive multimedia [22].

In streaming stored multimedia applications, for example, movie-on-demand, audio and video data are usually pre-recorded, compressed, and stored on multimedia servers, and are requested and retrieved through the network by clients. The idea of streaming is that the playback at the client side can start long before the entire movie is received. This may reduce the time that a user needs to wait before starting to watch the movie. Users of streaming stored multimedia usually can handle up to 10 seconds initial delay [22]. But once the playout starts, it has the constraints that subsequent data must be received in time for the continuous playout.

The streaming idea also applies to streaming live multimedia applications. An example application in this category is a live soccer game broadcasting. Differ from streaming stored multimedia applications, in streaming live multimedia applications, audio and video are not pre-recorded or stored on any server, rather are generated on-the-fly; consequently some operations at the client side, such as fast-forward, pause and resume, which are common to streaming stored multimedia applications, may not be supported by streaming live multimedia applications. Streaming live multimedia applications can lag tens of seconds after the first data frame arrives at the receiver depending on the size of playback buffer. It also has the delay variation constraints that are common in all real-time multimedia applications.

In real-time interactive multimedia applications, people attempt to communicate over a packet-switched network like they do in the real world. Internet telephony and video conferencing are two examples of such applications. Differ from the other two types of applications, in order to maintain interactivity, end-to-end delay constraints in this case are much stricter than the first two types. For example, business class voice over IP (VoIP) calls require an end-to-end delay of less than 150 milliseconds for proper comprehension of voice [8]; a longer delay may result in poor and imperceptible voice quality [9]. For video-conferencing, in order for the users to interact naturally, the end-to-end delay around 100 ms is widely recognized as the desired one way delay requirement for interaction [2].

Other than delay, the service requirements of multimedia applications can also be specified in terms of bandwidth, delay variation (jitter), and loss ratio [22]. Bandwidth specifies the bit rate needed to deliver the multimedia data, it is normally application specific. Jitter is the variability of packets delays within the same packet stream [22]; it has implications in dimensioning the receiver buffer. In terms of loss ratio, most multimedia applications can tolerate loss to some extent. For example, Internet phone may tolerate packet loss ratio up to 10% [22].

In multimedia network applications, multimedia documents are encoded before being transmitted over a network. One of the current popular multimedia encoding technique is MPEG for video and audio compression. MPEG (Motion Picture Experts Group) is a state of the art data compression standardization group. Among MPEG standards, the emerging MPEG-4 [21] is becoming more popular. In this encoding scheme, multimedia data is encoded into audio and video frames. Different types of frames are defined to take advantage of spatial and temporal redundancy within multimedia data, so that compression is achieved yet still maintaining good multimedia quality at decoding. The MPEG-4 standard is open source [34], well-documented; MPEG-4 traces are publicly available. Each trace includes both an audio part and a video part. In this thesis research, as proof of concept, MPEG-4 video will be used.

After encoding, multimedia frames are sent to the network for transmission. Depends on the size of the maximum transfer unit (MTU) of the underlying network, each frame is subject to fragmentation at the sender before transmission. When packets carrying multimedia data arrive at the receiver after traversing intermediate routers along the path, they are reassembled into frames. These frames are played back at a constant frame rate [21]. To ensure a good playback quality, the fraction of frames that miss their playback times should be kept minimal. Thus the goal for the network is to deliver as many frames as possible to the receiver by their playback times.

To ensure the good performance in terms of the fraction of frames that arrive at the receiver in time for playback, certain level of delivery performance at the network layer is called upon. On the classical Internet, at the network layer, only best-effort service is provided. All data packets are serviced in a first-come-first-served (FCFS) manner inside routers. The delivery performance is good when the traffic is light. However, it does not scale well when load is heavy. Efforts have been made to provide quality-of-service (QoS) support to multimedia data delivery over the Internet. These efforts can be classified into application-layer schemes and network-layer schemes. At the application layer, schemes such as application adaptation and rate shaping have been used to adapt multimedia sending rate to the changing network conditions. At the network layer, frameworks including IntServ and Diffserv [38] can be used to provide QoS support to multimedia data delivery.

This thesis addresses effective and efficient support to multimedia network delivery. The research is based on a novel framework that was designed to support real-time document delivery applications over packet-switched networks, namely deadline-based network resource management [26, 28, 36]. In this framework, the notion of application data unit (ADU) is used. An ADU may correspond to a file. It may also correspond to an audio or video frame in audio or video transport. Each real-time ADU is associated with a delivery deadline, which is specified by the sending application and represents the time at which the ADU should be delivered at the receiver. The ADU deadline is mapped into packet deadlines, which are carried by packets and used by routers for channel scheduling. Deadline-based channel scheduling is employed in routers; packets with more urgent deadlines are serviced first. Such deadline-based scheduling has been shown to provide superior service to FCFS in delivering both discrete and continuous (multimedia) real-time data [36]. When the network is congested, some channels in the network are overloaded; queues at these channels may grow significantly, real-time delivery performance will be severely affected. In this case, deadline-based scheduling alone may not be sufficient to prevent degradation of delivery performance. Other resource management mechanisms such as admission control are needed.

Previous studies on deadline-based networks have mainly focused on real-time discrete documents. Example discrete real-time documents are text message, file transfer, or business transactions. In contrast, real-time multimedia data is characterized by a sequence of ADUs sent at a fixed rate. Effective and efficient delivery of multimedia documents has not been studied. In this thesis, the performance of multimedia delivery in deadline-based networks at various load conditions is first studied. For heavy and overload conditions, two application-layer admission control algorithms are developed to address the performance degradation issue. The admission control decision is at the multimedia document level. In particular, *end-system* *based* admission control schemes are introduced. With end-system based schemes, admission control functionality is solely implemented at both ends of the communication, no modification is incurred inside network routers, this is in contrast with a router-assistant approach to admission control in which every router explicitly participates in the admission control procedure. The latter approach has been adopted by IntServ and DiffServ frameworks. Because of the complexity in maintaining traffic flow states at routers it incurs, the former may be simpler and easier to deploy.

In this thesis, the two application-layer admission control algorithms developed are referred to as Algorithms I and II respectively. Algorithm I utilizes a network condition indicator "positive ACK rate" to make admission control decision; this indicator is used to infer the congestion level along a source/destination path. Algorithm II makes use of more information than the above indicator, namely bandwidth requirements of the arriving multimedia document as well as other outstanding documents. The performance of these two algorithms is evaluated by simulation. Experimental results show that both algorithms can be used to prevent throughput degradation at heavy and overload conditions. In between the two algorithms, Algorithm II achieves better performance than Algorithm I.

Besides the end-system based admission control, another technique investigated in this thesis is the deadline assignment scheme for packet delivery. In previous studies of deadline-based networks, document deadlines are assigned by sending applications and denote the absolute time at which the document should arrive at the receiver. A document deadline at the application level is mapped to the packet deadline at the network layer. Packet deadlines are carried by packets and are used for channel scheduling inside the network. A more urgent deadline gives a packet higher priority in terms of scheduling. Because of the encoding method of multimedia data, during the delivery of multimedia content, certain types of multimedia frame are considered more important than the others; it is desirable to achieve higher on-time performance to more important frames. In addition, when real-time multimedia traffic shares network resources with real-time non-multimedia traffic, the former may be given a bit higher priority, which may help improve its delivery performance. Both can be implemented using differential deadline assignment. The technique is to tell apart two different deadlines: playback deadline and scheduling deadline. The former denotes the time at which the frame should be received at the receiver for smooth playout; the latter is used by routers along the path for channel scheduling. More urgent scheduling deadlines are assigned to more important frames within different types of multimedia frames. More urgent deadlines can be assigned to real-time multimedia traffic when it shares the same channel with real-time non-multimedia traffic.

The rest of this thesis is organized as follows. Chapter 2 contains background information of this study and reviews the related literature to this research. In Chapter 3, the performance of multimedia delivery in deadline-based networks when without admission control or differential deadline assignment is studied. The results in this chapter motivates the research carried out in subsequent chapters. In Chapter 4, the two end-system based admission control algorithms developed for multimedia delivery in deadline-based networks are described. In Chapter 5, the differential deadline assignment schemes are presented and evaluated. Finally, Chapter 6 contains a summary of this thesis work and a discussion of possible future work.

Chapter 2

Background and Related Work

This thesis focuses on multimedia delivery over deadline-based networks. As mentioned in Chapter 1, as proof of concept, MPEG-4 video will be used. In this chapter, I start with some background information on the MPEG-4 video encoding scheme. In the literature, there has been much research on how to deliver multimedia on the current Internet. Some of these existing ideas may be useful to this research. Quality of Service (QoS) strategies for multimedia delivery over the Internet is reviewed. An important network resource management strategy for supporting multimedia delivery is admission control. After reviewing existing QoS strategies, literature on admission control is examined.

2.1 The MPEG-4 Video Encoding Scheme

MPEG-4 video encoding scheme is specified in [21]. An encoded MPEG-4 video consists of a sequence of frames, with a fixed number of video frames per second, for example, 25 frames per second. Three types of frames are defined: I-frame, P-frame, and B-frame. I-frames (Intra-coded frames) store encoded still images and have no dependence on any other frames; they are the points for VCR-like random access in MPEG-4 streams. P-frames (Predictive-coded frames) contain information about the part of the video information that changes as time elapses; they require information from the previous I-frame and/or previous P-frames for encoding and decoding. A B-frame (Bi-directionally predictive-coded frames) is defined as the difference between a prediction of the past image and the following P- or I-frame. It requires information of the previous and following I- and/or P-frames for encoding and decoding. It can attain the highest compression ratio among the three types of frames. MPEG-4 uses such different types of frames to serve the contradictory goals of a high compression ratio and the fast random access. For fast random access, coding the whole data stream as I-frames would be the best. On the other hand, the highest degree of compression would be achieved by using as many B-frames as possible. Thus different types of frames are devised. As a consequence of this encoding technique, I-frames are normally the largest in size, followed by the P-frames, and finally the B-frames. The I-, P-, and B-frames are arranged in a periodic pattern referred to as a Group of Pictures (GoP). Each GoP contains one I-frame and multiple P- and B-frames. These frames are arranged in the display order. An example GoP pattern is: "*IBBBPBBBP*". Each video stream consists of a sequence of GoPs. As the frame transmission rate is constant, given variable frame sizes, MPEG-4 video streams have variable bit rates. Besides variable bit rates, another characteristic of multimedia traffic is that errors can be propagated through the frames because of inter-frame dependency. In addition, a packet network may potentially delay or drop packets. All these place a challenge in multimedia networking.

Because each I-frame serves as a synchronization point for the subsequent B- and P-frames, among the three types of frames, I-frames affect the quality of playback the most. As an example, assume a video frame rate of 25 frames per second and a 12-frame GoP pattern, there are approximately two GoP's per second. I.e., there is one I-frame every half second. A single lost or corrupted I-frame can cause up to 0.5 second's pause in the video before the next I-frame re-establishes the video decoding order. In this case, some portion of the received data may become useless because a frame, in which the current frame depends on, is lost or undecodable [42]. This results in network resource wastage. In contrast, a single B- or P-frame can be lost with little visible effect on the quality of the video playback. The goal of multimedia network resource management is to minimize the waste and to maximize the throughput. Among the three types of frames, I-frames are more important than the other two.

2.2 QoS Strategies for Multimedia Delivery Over the Internet

As discussed, transmission of real-time multimedia typically has delay, jitter, and loss requirements. However, the classic Internet provides only the best-effort service; it does not offer any QoS support to real-time multimedia applications. There has been much research that aims at establishing QoS capabilities in the current Internet.

Approaches that have been proposed to providing QoS support can be divided into two categories: end-system based and network layer approaches. End-system based approaches are end-to-end schemes without explicit support from the underlying network, therefore, no changes need to be made in a classic IP router. Network layer approaches, on the other hand, are schemes with network layer assistance, thus some changes are needed in routers. I review these two categories in sequence.

2.2.1 End-System Based Strategies

End-system based strategies can be classified into two groups: rate adaptation schemes and Content Delivery Network (CDN). The main idea of rate adaptation schemes is to maximize the likelihood that the quality of the received video remains acceptable to the viewer by way of application adaptation. End-systems are designed to react to congestion and to keep network utilization high by adjusting their data transmission rate to the current available network bandwidth. These rate control schemes are employed at the source. Feedback mechanisms are usually used to detect changes in network condition.

In [13], Hou *et al.* proposed an end-to-end architecture to effectively deliver MPEG-4 video over the Internet. Their architecture includes an end-to-end feedback control algorithm and a source encoding rate control algorithm. Feedback control algorithm uses packet loss as congestion indication. This feedback control algorithm controls the MPEG-4 video encoder, which in turn performs adaptive encoding in respond to network congestion. In RAP (Rate Adaptation Protocol) [32], an end-to-end TCP-friendly congestion control mechanism is devised to dynamically adjust the transmission rate of a video flow when path condition fluctuates. RAP's rate adaptation part adopts TCP's additive increase/multiplicative decrease congestion control mechanism, and uses triple duplicate ACKs to indicate packet loss. Both TCP congestion control [22] and RAP [32] have in common the use of end-to-end acknowledgement to infer network congestion. In my research, I will adopt a similar means.

In the above schemes, the feedback comes from receiver acknowledgement. In comparison, one can actively probe the current network condition for rate adaptation. In [23], Lakshman *et al.* propose a "rate-matching" scheme for transmission of variable bit rate (VBR) video for interactive applications in an ATM network. In this scheme, a set of cooperating sources periodically probe the network for the appropriate transmission rate, video source rate is matched to the probed available bandwidth by modifying the quantization level that is used during compression.

Similar to the above adaptive encoding schemes, another approach to rate adaptation is rate shaping [37]. The objective of rate shaping is to match the rate of a pre-compressed video bit-stream to the current available bandwidth. A rate shaper (or filter) performs rate shaping. For example, a frame-dropping filter is used to reduce the data rate by dropping some frames according to their importance or priority [7, 14]. The difference between rate shaping and the rate adaptation schemes above is that rate shaping is not performed at source, but rather, within the network.

Besides rate adaptation, CDN (Content Delivery Network) [33] is another approach to QoS support for multimedia delivery. It improves data (including multimedia) delivery performance and overcome problems such as network congestion and server overload by doing content replication and load balancing at the application layer. A CDN consists of networked computers. These computers form an application-layer overlay network. It provides logical end-to-end service delivery infrastructure on top of the existing data transport network. The CDN schemes do not attempt to adapt either the encoding or the transmission rate of the video data. In [1], Apostolopoulos *etal.* investigated an approach that relies on the simultaneous transmission of several substreams of a video over different paths, where each substream encodes a portion of the video. The video can be correctly decoded with graceful quality degradation, even if some of the sub-streams are missing or incomplete. In [35], multiple paths between a given source and destination are assumed to be known. The proposed scheme can dynamically select the best path in order to optimize the quality of the received video. It should be noted that CDN is mainly designed for delivering stored multimedia rather than delivering live or interactive multimedia.

The approach taken in this thesis research is also end-system based. However, differ from end-system based multimedia rate adaptation and CDN, admission control is exercised. Albeit so, schemes presented in this section may be used at the same time when admission control is used so that the network multimedia delivery capability is further enhanced.

2.2.2 Network-Layer Strategies

At the network layer, two major frameworks - Integrated Services (IntServ) and Differentiated Services (DiffServ) - have been developed by IETF (Internet Engineering Task Force) to provide QoS support on the Internet. Both frameworks may be used to deliver multimedia content on the Internet. Two terms are defined and commonly used in these two frameworks. A "flow" is commonly defined as a stream of packets with the same source IP address, source port number, destination IP address, destination port number, and protocol ID [38]. A "class" represents a set of packets that requires specific delay, loss, and jitter characteristics. A class may contain packets from many flows. IntServ is a per-flow based QoS framework with dynamic resource reservation and can provide guaranteed QoS to individual flows. IntServ works with a signaling protocol RSVP which allows applications to reserve bandwidth in all routers along the flow path. In addition to a guaranteed service, IntServ also provides a controlled load service. Controlled load service is for applications requiring reliable and enhanced best effort service [38]. IntServ requires classification and policing at all routers, this raises a scalability concern as thousands of flows may pass by core routers simultaneously. In contrast to per-flow based IntServ, DiffServ is a per-class based QoS framework that provides service differentiation [39] among classes. Forwarding classes including expedited forwarding (EF) and assured forwarding (EF) are encoded in the packet header to indicate the need for low-delay, high throughput, or low-loss-rate service. Each customer negotiates a SLA (Service Level Agreement) with the ISP (Internet Service Provider). Compared with IntServ, DiffServ has better scalability because no per-flow based state information is stored at routers, rather only per-class state is kept. The down side of DiffServ is that end-to-end per-flow based guaranteed service may be difficult to attain.

These two Internet QoS frameworks are utilized by some multimedia delivery research. Under network congestion, the network may discard some packets of video streams. A small packet loss rate may translate into a much higher frame error rate. For example, a 3% packet loss percentage could translate into a 30% frame error rate [5]. Ziviani *et al.* [41] proposed a mechanism that combines DiffServ with RED (Random Early Detection) active queue management to improve the delivery quality of MPEG video streams. Because frames (packets) of encoded MPEG video have unequal importance, in their scheme, different levels of drop precedence are associated to packets that carry information of different frames. That is, packets transporting fragments of a B-frame are more likely to be discarded in a congested router than packets from I- or P-frames. This differentiation among packets from different types of frames are achieved by mapping those packets onto different drop precedences in a DiffServ AF (Assured Forwarding) class [12]. Along with a three-level RED mechanism for queue management, this scheme also provides the necessary drop discrimination using DiffServ mechanisms

In this thesis research, "class" is defined as all the traffic between a given source and destination pair. I do not devise network-layer mechanisms, rather I focus mainly on devising mechanisms within end-systems. At the network layer, the deadline-based channel scheduling is used.

2.3 Admission Control

Multimedia applications have special QoS requirements at the network layer. If there is no congestion within a network, QoS can be easily provided. However, within a packet network, because of statistical multiplexing (rather than deterministic multiplexing), congestion may occur. Network congestion causes degradation in network performance in the form of long queuing delays and packet losses. When network congestion occurs, user service may suffer high delay and low throughput. To deal with congestion, multiple resource management strategies can be used. Examples are selective packet dropping in routers, which is called buffer management, and admission control.

An admission control algorithm determines whether a new traffic stream can be admitted to the network for delivery without jeopardizing the QoS assurances granted to earlier traffic streams. As each traffic stream needs certain amount of network resources (e.g., link bandwidth and router buffer space) for transferring data from source to destination, admission control is used to control the network resource allocation. The goal is to correctly compute the admission region: an algorithm that unnecessarily denies access to flows that could have been successfully admitted will under utilize network resource; while an algorithm that incorrectly admits too many flows will induce QoS violations. There are many approaches to admission control and they can be classified into two categories: *measurement-based* and *parameter-based.* Admission control can be performed at both the network layer and the application layer. In the following two subsections, different admission control schemes are reviewed.

2.3.1 Measurement-Based Admission Control

A large stream of existing admission control schemes is based on on-line measurement at network routers. An admission control algorithm for a predictive service based on time-window measurement mechanism is introduced in [17]. Predictive service offers a fairly, but not absolutely, reliable bound on packet delivery times. The ability to occasionally incur delay violations gives admission control more flexibility. The measurement-based approach proposed in [17] is combined with the relaxed service commitment of predictive service, this keeps network utilization at a high level while still reliably meets the delay bound. Lee *et al.* [24] introduce a new framework for various measurement-based connection admission control (MBCAC) schemes in a multi-service network. "Available bandwidth" evaluation based on on-line measurements is used for admission control while an adaptive "ad hoc" feedback mechanism is used to protect the network when the CAC is too aggressive. Performance of different measurement-based admission control algorithms are compared in [18]. The key component of these algorithms is a measurement process that produces an estimate of the current network load, and then this load estimate is used to make the admission decisions.

Admission control can be used to reduce level of network congestion. But often, there is a tradeoff between providing QoS guarantees to multimedia delivery and maintaining high network utilization. Following the above measurement algorithm in [17], Bao and Sethi [3] proposed an adaptive admission control scheme. In this scheme, admission decision is based on adaptive measurement of network residue resources - the resources that are not utilized by the existing flows inside the network. The measurement result is between minimum of samples and average of samples. Thus, the network can choose different levels of conservativeness when estimating residue resources. As the result, admission control can be tighter or looser according to network conditions. This self-adjusting admission control scheme tries to achieve the highest network utilization yet still providing reliable QoS. Taking a similar approach, the real-time adaptive admission control (AAC) scheme proposed in [15] uses the following information: the available capacity from an adaptive bandwidth estimation scheme, a congestion indicator derived from a congestion controller, a peak cell rate estimate from new sources, along with the desired QoS metrics.

In contrast to above work based on measurements at routers, Elek *et al.* [10] proposed admission control procedure based on measurements at end-systems only. Such an end-to-end measurement was used to provide the controlled load service in IntServ. A sender sends probe packets periodically. The receiver sends back a measurement report to the sender which consists of the number of probe packets received. The measurement report is carried with high priority to ensure that it is transmitted with low loss. Admission decision is based on the calculated probe loss probability using information in the measurement report.

2.3.2 Parameter-Based Admission control

Parameter-based admission control can be divided into *deterministic* and *statistical* algorithms [40]. A deterministic algorithm [20] aims to provide guaranteed service, however it must assume the worst case for some of the system parameters and hence often under-utilizes available resources. Statistical admission control [19] approach generally allows higher utilization of resources, in exchange for a residual probability of missed request deadlines.

Admission control schemes for multimedia traffic are different from those for general data traffic because of the bursty nature of encoded multimedia data. It is often difficult to predict or characterize the bandwidth requirement of multimedia data. Earlier approaches to admission control of multimedia streams have used the peak rate or the average rate. Later approaches endeavors to devise more sophisticated rate estimation schemes. Pancha and Zarki [30] proposed a prediction method; it derives the bandwidth requirement of a single video source based on the information of the mean frame size, bandwidth of previous frames, and the standard deviation of the frame size. In [6], a neural network approach is adopted to predict the required bandwidth dynamically. Both deterministic and statistical admission control schemes discussed above are implemented at the network routers.

2.3.3 Admission Control in Deadline-Based Networks

In deadline-based networks [26, 36], congestion can happen under heavy load [28]. Two application-layer admission control algorithms - "Acceptance Probability" and "Estimated Bandwidt" - are developed in [29] to alleviate load and improve performance. These two admission control algorithms are targeted at *discrete* real-time data and the admission control is at the ADU level. "Estimated Bandwidth" has better performance because it utilizes more information when making admission decisions - it makes use of the bandwidth requirement of an arriving ADU as well as the bandwidth requirements of all the outstanding ADUs'. Differ from this study in which discrete real-time documents are f concern, the schemes in this thesis research target real-time multimedia data.

Chapter 3

Multimedia Delivery in Deadline-Based Networks

In the last chapter, approaches to supporting multimedia delivery over the current Internet are discussed. The goal of this thesis work is to develop network resource management strategies that support multimedia delivery in deadline-based networks. In this chapter, I study the performance of multimedia delivery in deadline-based networks when no special network resource management strategy is in use. The results from this chapter will server as the benchmark for evaluating the performance of the strategies that are developed in the later chapters. The method that I used in studying the performance is by way of discrete event simulation. I first describe the simulation model of deadline-based networks that is set up for this research, and then present the simulation experiments and results.

3.1 Performance Model

I simulate the scenarios in which video clips are transmitted over a shared deadline-based network among a number of users. The functionalities at each sender and receiver are first described. The network model, traffic model, performance metrics, the simulator, and simulation methodology that are used in the experiments are described in sequence afterwards.

3.1.1 Sender and Receiver Models

Consider sending a video clip from a sender to a receiver via a network. Each video clip can be considered the video portion (as opposed to the audio portion) of a multimedia clip. At a sender, an arriving video clip is characterized by four attributes: its source, destination, arrival time, and delivery deadline. Source and destination identify the sender and the receiver of this video clip respectively. The arrival time is the time at which the video clip transmission request arrives at the network. Since each video clip consists of a series of video frames, the first frame is assumed to be sent the moment the clip arrives at the network, i.e., sent at the video clip arrival time. All subsequent frames are sent at regular time intervals.

The delivery deadline of a video clip specifies the time at which the first frame in this clip that needs to be received by the receiver. This deadline is an application-layer deadline and is normally application dependent. This "video clip deadline" can be used to derive a frame deadline. Deadline for the rest of the frames of the video clip will be the video clip deadline plus their time offset from the first frame. For example, if a video clip's delivery deadline is D, and the time interval between two consecutive frames is t, then the ith frame's delivery deadline will be $D + (i - 1)t$.

Each frame is passed from the sending application to the transport layer, together with its size and deadline. At the transport layer, a maximum segment size M is defined; a frame is segmented into $m \geq 1$ transport segments. When each of these m segments is passed to the network layer, segmentation may again be performed since a maximum packet size P is defined at the network layer. As is normally the case on the Internet, a one-to-one mapping between transport segment and packet is assumed. The frame deadline is mapped onto packet deadlines. For simplicity, in this section, I assume that all packets of a frame carry the frame deadline.

Packets are routed through the network until they reach their destination node. At the receiver, all the packets that belong to the same frame are re-assembled and sent to the application layer. For simplicity, the processing times at the sender and the receiver are not modeled in my simulation. Once the frame is received by the receiving application, an ACK is generated and returned to the sender to indicate whether the frame is received on-time. A frame is on-time if all its packets are received on-time with respect to packet deadlines. If the frame is on-time, a positive ACK is returned; otherwise, an negative ACK is returned. It is possible that some packets of a frame are dropped inside the network due to buffer overflow, in this case, the frame can not be re-assembled; no ACK will be returned to the sender. Such a frame is assumed to be lost. Of particular interest is real-time interactive multimedia; it has the stringent delay requirement, thus it is assumed that the delay incurred by re-transmission is excessive; in my model, lost frames are not re-transmitted.

3.1.2 Network Model

The network model that is used in performance evaluation includes 12 nodes and 42 channels. Its topology is depicted in Figure 3.1. Each link in this graph consists of two channels, one per direction. The capacity of each channel is assumed to be 10 Mbit/sec and the propagation delay on each channel is assumed to be 10 millisecond. Given an estimated propagation speed of 10 microseconds/mile or 6 microseconds/km, this topology models a wide area network. The buffer size at each channel is assumed to be 50 packets. The maximum packet size at the underlying network layer is assumed to be 1000 bytes. Thus each buffer can hold up to 40 ms of data. Packets are routed through the network until they reach their destination node. Fixed routing is assumed. The maximum number of hops that a packet travels from a sender to

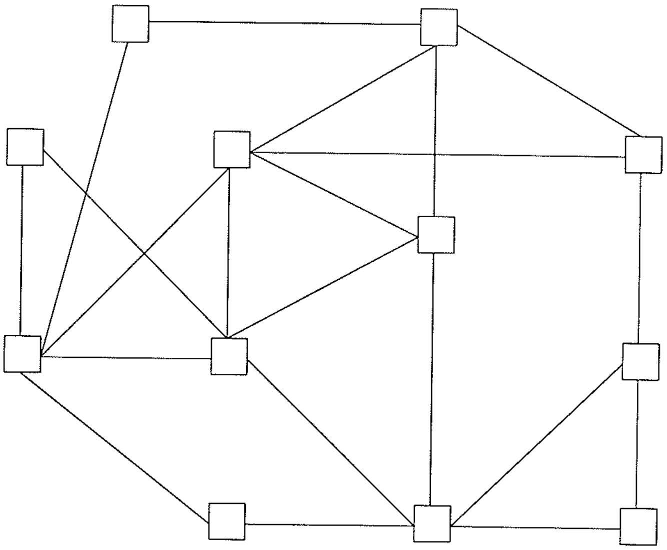


Figure 3.1: Network model

a receiver in this network model is 4. Recall a class denotes all traffic between a given source/destination pair. Given average arrival rate per class and fixed routing, the channels that carry most traffic can be identified. These channels are called the bottlenecks.

A deadline-based scheduling algorithm is implemented at network routers. The deadline-based channel scheduling scheme used in this study is the T/H algorithm [28]. This algorithm is based on the ratio T/H, where T is the time left (delivery deadline - current time) and H is the number of hops to destination from the current node. T/H is calculated when a packet arrives at a router, it can be viewed as the urgency of a packet; a packet with a smaller T/H means it is more urgent. There are two priority queues at each outgoing channel: the real-time queue and the best-effort queue. Packets in the real-time queue are serviced according to their T/H values. Packets in the best-effort queue are first-come-first-served. The real-time queue has higher priority than the best-effort queue. If a real-time packet is already late when it arrives at the channel ($T < 0$), then it is downgraded to the best-effort queue.

3.1.3 Traffic Model

The video clips used in this study are obtained from video traces that are available in the public domain [34]. Each video clip is derived from the data of a trace file. 34 trace files are used in this thesis work. They fall into a number of video content categories that include drama, action, cartoon, news, and sports. All the trace files are 20 seconds long. Each trace contains a sequence of frame sizes in bytes. The inter-frame time of these frames is 40 milliseconds, thus there are 500 frames in each trace. The GoP of these traces consists of 12 frames and the GoP pattern is "*IBBPBBPBBPBB*". The average bit rates of these traces ranges from 84.5 Kbps to 1.3415 Mbps, with an average of 0.55 Mbps. A histogram of average bit rates for the 34 traces is plotted in Figure 3.2. Four traces have lower than 0.3Mbps average bit rates, 9 traces have average bit rates between 0.3 and 0.5Mbps, and so on. Over all traces, frame size ranges from a few hundred bytes to over 15000 bytes, which may lead to the bursty nature of multimedia documents. The average frame size over all traces is 2884 bytes.

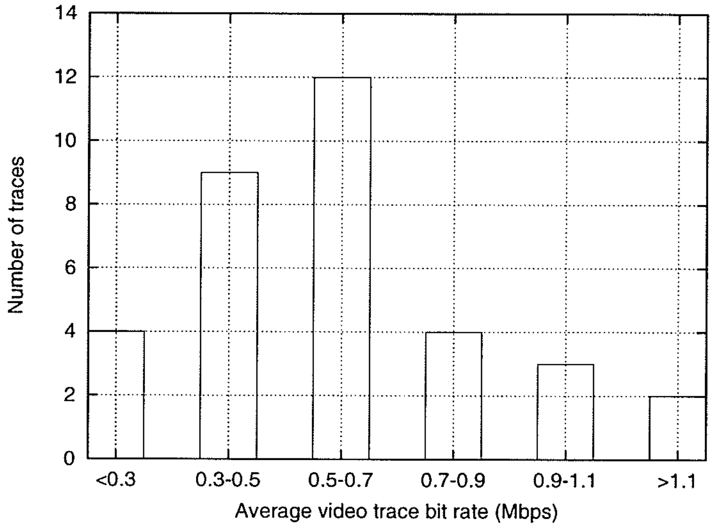


Figure 3.2: Average bit rates of video traces

It is assumed that video clip transmission requests arrive at each sender at a rate of λ clips per second. The video clip inter-arrival time is assumed to be exponentially distributed. At each sender, for each arriving video clip, the destination node is selected at random. Once the destination node is decided, a video clip is selected from the 34 trace files at random. After a trace file is selected, a sender starts to send frames (whose sizes are read from the trace file) at a constant rate - one frame every 40 milliseconds. Given this traffic model and the network model described in the last subsection, there is one bottleneck channel in the network. 13 classes are carried on this channel.

For each arriving video clip, the delivery deadline is modeled as follows. Let $x(y)$ be the end-to-end latency to transmit a frame of size y when there is no queueing and no segmentation. Let x_p be the end-to-end propagation delay, c_j the capacity of the j-th channel along the path based on shortest-path routing. Then $x(y)$ can be estimated by $x(y) = x_p + \sum_j y/c_j$. For each video clip i, I assume that the maximum frame size y_i is known at the video clip arrival time. For stored multimedia delivery, such as video on demand, y_i is known. For streaming live and real-time interactive multimedia data, I assume that y_i can be estimated. Given the maximum frame size y_i for video clip i, the allowable delay of the first frame in the video clip is assumed to be proportional to $x(y_i)$. Hence, the delivery deadline for the first frame, i.e., the video clip deadline, is given by $d = \mathit{arrival\ time} + k \times x(y_i)$, where k is referred to as a "deadline parameter". Deadlines for the rest of the frames of a video clip will be the video clip deadline plus their time offset from the first frame. If a video trace is coded m frames per second, then the n^{th} frame's deadline will be the video clip delivery deadline plus $(n-1)/m$ seconds.

In this thesis, the urgency level of a deadline is specified using parameter called *deadline parameter* k. In my experiments, k is modeled as 1.0 plus an exponentially distributed random variable ε, i.e., $k = 1 + \mathit{expo}(\varepsilon)$, therefore $1.0 \leq k \leq \infty$. k is greater than 1 would ensure that deadlines are reasonable; i.e., they are at least as large as the end-to-end latency. k can also be arbitrarily large. Including an exponentially distributed random variable component is because firstly, by varying ε, a wide variety of deadline urgency can be modeled; secondly, two other variable packet delays inside the network - queuing and processing delays - can be accommodated in the allowable end-to-end delay. In my simulation, for data packets, $k = 1 + expo(0.2)$, which represents a fairly urgent deadline. With respect to ACKs, at underlying network layer, ACK are treated the same as data packets. Routers do not identify whether an incoming packet is an ACK packet or not. ACKs are also associated with delivery deadlines. The difference with data packets is that ACK packets carry slightly more urgent deadlines. This is achieved by using a smaller deadline parameter k than data packets. The assumption is that ACKs are normally small in size, and are useful in network resource management, thus they are given a little higher priority in channel scheduling than data packets.

3.1.4 Performance Metrics

The first performance measure of interest is *bottleneck on-time throughput.* This is defined to be the number of frames that are delivered on-time per unit time for those video clips that pass through the bottleneck channel. I also collected the statistics for calculating the frame on-time rate of bottleneck traffic. The *frame on-time rate* is defined to be the number of frames that are delivered on-time over the total number of frames sent.

Given the above network model and traffic model, assuming all the frames are received on-time, the theoretical maximum throughput for bottleneck classes is given by (*bottleneck capacity*÷*average frame size*). In this network model, the theoretical maximum throughput is approximately 433 frames per second. This will be used in evaluating the performance when without and with resource management schemes in this and subsequent chapters respectively. Besides aggregated performance across all frame types, per frame-type on-time throughput, as well as the video clip blocking rate when with admission control, are also collected for traffic that goes through the bottleneck channel and for the entire network traffic respectively.

3.1.5 NS2 Simulator

The NS2 simulator [11] is used in this study. NS2 is a multi-protocol network simulator; it includes a number of Internet protocols such as IP, IPv6, TCP, UDP, RTP, HTTP, and FTP, etc. NS2 also supports several channel scheduling algorithms inside routers such as FCFS and WFQ (Weighted Fair Queuing) [31], and a number of buffer management schemes such as RED (Random Early Detection) [25]. In this study, no flow control or congestion control at the transport layer is used, thus UDP is used as the transport layer protocol in my experiments. For channel scheduling, the T/H algorithm is used. The simple drop-tail buffer management scheme is employed. For modeling multimedia delivery, NS2 provides a basic mechanism to build Constant Bit Rate (CBR) media streams. However, Variable Bit Rate (VBR) multimedia traffic model is not included in the default distribution.

In this thesis work, NS2 is augmented at both the application layer and the network layer. At the application layer, trace-based traffic model is implemented. Video frame sizes are retrieved from trace files and fed into the simulator. At the network layer, two changes are made. First, a new packet header is introduced to carry packet's deadline information. Second, NS2 is augmented with an implementation of the T/H algorithm at the routers.

3.1.6 Simulation Methodology

I first performed some pilot experiments to determine the simulation run length. For this, I used the λ value of 1.25 which represents a heavy load condition. It was determined that the length of the transient period is around 20 seconds. For obtaining steady-state results, the simulation run length was chosen to be 300 seconds. Each experiment was repeated six times. The sample mean and confidence intervals were calculated. Because the width of the confidence interval is very small compared to the sample mean, only sample mean results are reported.

3.2 Congestion In Deadline-Based Networks

In this section, I consider the case when no resource management strategy is used to transmit multimedia data within deadline-based networks. The objective is to study the effect of congestion on the on-time performance obtained in transmitting multimedia. To achieve so, I vary the level of load to the network and observe the network performance. Later in this thesis, schemes such as admission control will be developed. The effectiveness of the developed schemes will be evaluated by comparing their performance with that of the case in this section.

In my simulation, I vary the video clip arrival rate λ from 0.25 to 1.43 clips/sec. For the network model that is described above, the bottleneck channel is saturated when $\lambda \approx 0.77$ video clips/sec. Thus the arrival rates selected represent a wide range of load levels to the network; from light load, to medium load, to heavy load, and to overload conditions.

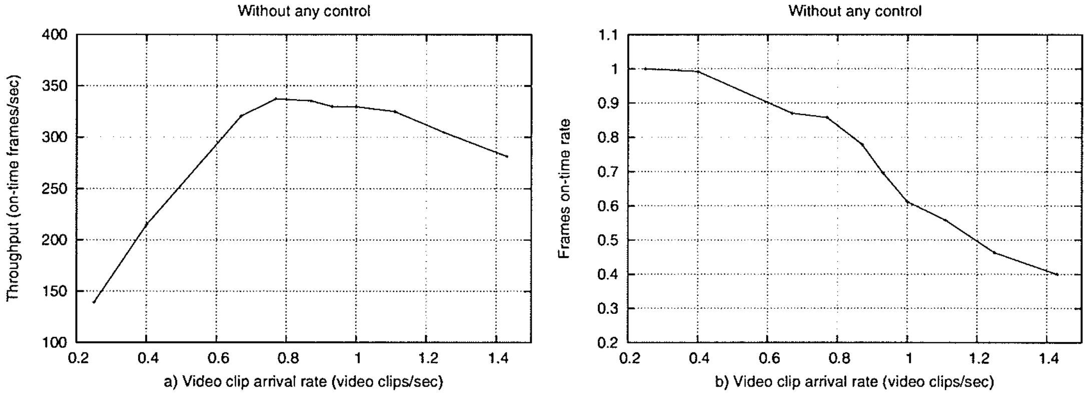


Figure 3.3: Performance when without any QoS schemes

The bottleneck on-time throughput results when λ is varied are shown in Figure 3.3 a). It can be observed that as the load increases, the bottleneck throughput continues to increase until λ reaches 0.77. When $\lambda > 0.77$, throughput starts to degrade. Throughput degradation can be explained as follows. When the system is at heavy load, the queueing delay at the routers becomes excessive, resulting in more late frames. At the same time, buffers at network channels start to become full, resulting in dropped packets. This would cause more frames not to be delivered on-time. The objective of admission control is to prevent throughput degradation in the presence of increased video clip arrival rates.

I also plot the performance in terms of the bottleneck traffic frame on-time rate. As shown in Figure 3.3 b), frame on-time rate drops dramatically after the bottleneck channel is saturated. Because of the similarity between on-time throughput and on-time rate in characterizing performance, in what follows, I will report on on-time throughput performance mainly.

Chapter 4

Admission Control for Multimedia Delivery

At the end of last chapter, the initial experiments of multimedia delivery in deadline-based networks exhibit throughput degradation when at heavy and overload conditions. The frame on-time rate also deteriorates severely when the load is beyond certain level. To prevent throughput degradation and to provide acceptable video delivery performance, especially when network is congested, the approach that is taken in this research is end-system based admission control. Two admission control algorithms are developed and evaluated in this chapter. Both algorithms perform admission control based on end-to-end feedbacks. A "positive ACK rate" is utilized as the feedback. It reflects the on-time performance of multimedia frames at the receiver side. Algorithm I makes admission control decisions based solely on positive ACK rate; Algorithm II also utilizes the estimated bandwidth of an incoming multimedia clip. I first describe the feedback, namely the positive ACK rate that is used by both algorithms. In Section 4.2, I present the two algorithms. After that, I evaluate the performance of these two schemes in Section 4.3.

4.1 Weighted Moving Average of Positive ACK Rate

In my proposed schemes, admission control is assumed to be performed by an admission control entity residing at the application layer of a sending host. Admission control is performed based on feedbacks. The feedback information is derived from ACKs returned from receivers: positive ACKs indicate good network condition, while negative ACKs indicate that the path is congested and fewer video clips should be admitted.

In my schemes, admission control decisions are made on a per source/destination pair basis. This is because for a network with fixed routing, packets to the same destination will traverse the same path inside the network. The ACK information returned from previous transmissions therefore reflects the congestion condition along the source/destination path, and may be used as indication for path conditions during upcoming transmissions.

To estimate the load inside the network for each source/destination pair, each sender maintains a rate R for every receiver. This rate R is called "Positive ACK Rate". R can be the ratio between the number of positive ACKs received and the total number of ACKs received. Another way to define R is to use the ratio between the number of positive ACKs received and the total number of frames sent. Because of possible packet drops, the first ratio may not be a good indicator of path conditions. For example, assuming in the last 2 seconds, a few dozen frames are sent. The network is congested so only two ACK packets are returned to the sender. Both ACK packets happen to be positive, then using the first ratio, the "Positive ACK Rate" is 1, but the network condition is far from ideal. In such cases, using the first ratio might be misleading. Therefore, the second ratio is used in this research. A higher R indicates better network condition, and $R \approx 1$ represents the ideal condition.

R can be calculated cumulatively since the start of an admission control agent, or it can be calculated only for last N seconds, or using a moving average. To more accurately estimate the load inside the network for each source/destination pair, I adopt the scheme used to estimate round trip time in TCP, which is "exponential weighted moving average [22]". In this scheme, a sample of R is collected every Q seconds. Each Q seconds is referred to as an update interval. At the end of update interval i, the moving average R_his is computed as a weighted average between the value of R_his at the end of the last update interval and R that is collected in the current update interval (See Equation 4.1). The exponential averaging

$$R\_his = (1 - \beta) * R\_his + \beta * R \tag{4.1}$$

coefficient β is used to smooth out the possible fluctuation in ACK packets transmission. The value of the moving average R_his is used by both admission control algorithms that are described in this chapter.

In all experiments, β is fixed at 0.25. The update interval Q is a tunable parameter. It decides how often each sender collects statistics to update R_his. Intuitively, the more often we collect the data, the more accurately can R_his reflect the channel congestion condition, the down side is the higher algorithm overhead. On the other hand, because *R_his* is utilized to help making admission decision on arriving video clips, Q should match video clips arrival rate, e.g., if video clips arrive every second, choosing Q value of 1 second may be reasonable. The effect of Q will be discussed in Section 4.3.3.

4.2 Admission Control Algorithms

In this section, two admission control algorithms for multimedia delivery over deadline-based networks are developed.

4.2.1 Algorithm I - Positive ACK Rate

The first algorithm only makes use of the moving average of the positive ACK rate *R_his*. This algorithm is shown at the end of this subsection and consists of two parts. The first part is the admission test. Upon an incoming video clip, the current *R_his* is compared with a threshold *R_threshold*. If *R_his* $\geq$ *R_threshold*, the video clip is accepted; otherwise, it is rejected. *R_threshold* acts as a doorsill of the network. If the doorsill is higher, then less video clips can be admitted, and vice versa.

The second part of this algorithm is the adjustment of *R_threshold*. *R_his* is refreshed periodically and then is used to update *R_threshold*. *R_his* denotes the moving average positive ACK rate after an update and *R_his_old* denotes the moving average positive ACK rate before an update. If *R_his* is larger than *R_his_old*, that means more positive ACKs are coming back, and network condition is becoming better. Thus, the "doorsill" *R_threshold* can be adjusted to a lower value and more video clips can be accepted. If *R_his* is less than *R_his_old*, that indicates network condition is becoming worse. In this case, *R_threshold* should be raised in order to stop more video clips from entering the network. If *R_his* is equal to *R_his_old*, *R_threshold* remains the same. An upper bound 0.9 and a lower bound 0.2 are set for *R_threshold* in order to keep it in a reasonable range. Through adjusting *R_threshold*, network load is controlled. The details of Algorithm I are shown below.

```
/* initialization */
R_threshold = R_threshold_init;
/* admission test */
if ( R_his $\geq$ R_threshold)
      accept;
else
      reject;
/* adjustment of R_threshold periodically */
if ( R_his > R_his_old )
      R_threshold -= $\Delta$D;
      if ( R_threshold < 0.2) R_threshold = 0.2;
else if ( R_his < R_his_ old )
      R_threshold += $\Delta$D;
      if ( R_threshold > 0.9) R_threshold = 0.9;
```

4.2.2 Algorithm II - Bandwidth Left

The second algorithm also consists of two components. The first component is an admission test. At each sender, a real-valued variable Sum is maintained for every destination d. Sum denotes the total bandwidth of all the outstanding video clips that are destined to d. Outstanding video clips are those that have been accepted and are being delivered.

Upon the arrival of a video clip, its destination d is retrieved. The bandwidth requirement (denoted by X) of the arriving video clip is estimated. Define two algorithm parameter C_{min} and C_{max}. If X is smaller than or equal to C_{min}, the video clip is accepted for transmission without further consideration, and its estimated bandwidth is entered into a committed bandwidth table. The idea is that video clips with a very small bandwidth requirement are always admitted. If $X > C_{min}$, the sum of X and the total bandwidth requirement of all outstanding video clips of destination d is calculated. If this sum is less than or equal to C_{max}, the arriving video clip is admitted, otherwise it is rejected. C_{max} serves as an upper bound on committed bandwidth for this destination.

The second component is concerned with the update of C_{max} and Sum. The update of C_{max} utilizes the weighted moving average R_his discussed in the last section. Initially, C_{max} is set to a pre-specified upper bound C_{max0}. If R_his is relatively high, network condition is considered to be good. Thus, more video clips can be accepted to the network. Otherwise, fewer video clips should be accepted. When the value of R_his changes, the parameter C_{max} is adjusted. Specifically, C_{max} is increased by ΔD if $R\_his \geq 0.9$, and decreased by ΔD if $R\_his < 0.9$. At no time can C_{max} be larger than the pre-specified upper bound C_{max0}. Also, it can never be smaller than C_{min}.

The update of *Sum* is as follows. *Sum* indicates the amount of bandwidth that is being used by all outstanding video clips along a source/destination path. When a video clip is fully transmitted, the bandwidth that was used by the video clip should be made available to the upcoming video clips. In this algorithm, once the last frame of a video clip is transmitted, its bandwidth is deducted from the sum of all the outstanding video clips that are going to the same destination. The details of Algorithm II are described below.

```
/* initialization */
Cmax = Cmax0;
/* admission test */
if ( X ≤ Cmin)
    accept;
    Sum += X;
else if ( Sum + X ≤ Cmax )
    accept;
    Sum += X;
else
    reject;
/* adjustment of Sum */
if ( last frame of a video clip is sent)
    Sum -= X;
```

```
/* adjustment of Cmax periodically */
if ( R _ his $\geq$ 0.9 )
    Cmax += $\Delta$D;
    if ( Cmax > Cmax0) Cmax = Cmax0;
else
    Cmax -= $\Delta$D;
    if ( Cmax < Cmin ) Cmax = Cmin;
```

It can be seen that Algorithm II needs to maintain the set of outstanding video clips together with their bandwidth requirement. In contrast, Algorithm I does not need these information.

4.3 Performance Evaluation of Admission Control Algorithms

In this section, the performance of the two proposed admission control algorithms is evaluated by simulation. The first step in my performance evaluation is to identify those factors that affect the performance and to quantify their impact on performance. After that, more experiments are carried out to further study the effect of these factors on performance. The performance when there is no admission control serves as the benchmark.

4.3.1 Performance Evaluation of Algorithm I

For admission control Algorithm I, the following algorithm parameters may affect its performance: *R_threshold_init* and ΔD. *R_threshold_init* is the initial value of *R_threshold*. As discussed in the last subsection, *R_threshold* plays the role of a doorsill for network traffic: a high doorsill stops more traffic from entering the network; on the other hand, a low doorsill allows in more traffic. Therefore, a large value of *R_threshold_init* indicates that the admission control is initially conservative. Conversely, a small value of *R_threshold_init* represents an initially aggressive admission control entity. ΔD is the amount to change when periodically adjusting the value of *R_threshold*. When network condition changes, the weighted moving average *R_his* changes. If it becomes higher, *R_threshold* is decremented by ΔD; if it drops, *R_threshold* is incremented by ΔD. The extent by which *R_threshold* is adjusted may trigger smaller or larger responses to congestion feedback. The effect of these two parameters on performance is first investigated through a 2^2 factorial experimental design [16].

In my model, besides the above two algorithm parameters, other factors that affect the performance include the video clip arrival rate λ and the video deadline urgency parameter k. In the 2^2 factorial design, "typical" levels are chosen for the other two factors as follows: λ equals to 1.25, it corresponds to a heavy load condition; k equals 1.2, it represents fairly urgent deadlines. For each of the two algorithm parameters, lower- and upper-bound levels are chosen. These are:

- A: *R_threshold_init* = 0.2 and 0.9
- B: ΔD = 0.01 to 0.1

In my 2^2 factorial design experiments, the same simulation methodology as in Section 3.1.6 is used. Based on the simulation results from the four experiments, the effect of the two algorithm parameters is calculated and is listed in Table 4.1. It shows that varying the level of ΔD accounts for around 96% variation in performance, while the other parameter and the interaction of the two parameters account for insignificant portion of variation in performance. Thus ΔD has much larger impact on performance than *R_threshold_init*.

Table 4.1: 2^2 factorial design results for Algorithm I

Factor	A	B	AB
Effect (%)	1.28	96.23	2.49

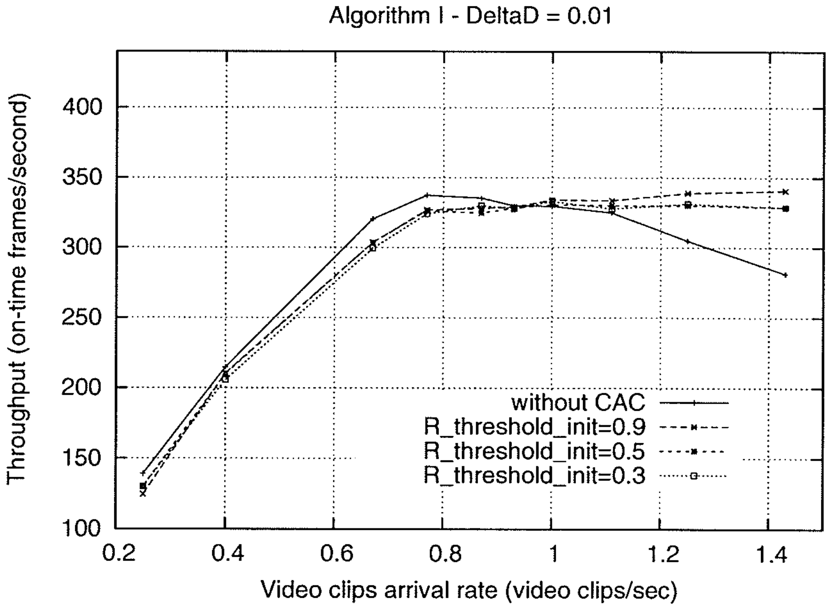


Figure 4.1: Algorithm I: effect of *R_threshold_init*

I next conduct more experiments to study the algorithm performance when

R_threshold_init and ΔD are varied. Figures 4.1 and 4.2 plot the experimental results when *R_threshold_init* and ΔD are varied respectively, the corresponding results for the case of no admission control are included. In both experiments, the deadline parameter k of 1.2 is used. The video arrival rate λ ranges from 0.25 to 1.43. The bottleneck on-time throughput performance is collected. In Figure 4.1, three values of *R_threshold_init* are experimented, namely 0.3, 0.5, and 0.9. ΔD value is 0.01. It can be observed that (i) compared with the case of no admission control, all three values of *R_threshold_init* result in throughput improvement when load is heavy ($\lambda > 0.9$); at low to medium load ($\lambda \leq 0.9$), there is a small degree of performance degradation. This is a side effect for a network that employs admission control [4]. (ii) Consistent with the 2^2 factorial design results, changes in the value of *R_threshold_init* do not incur significant changes in performance. This holds for other values of ΔD as well.

The effect of ΔD is presented in Figure 4.2. In this experiment, *R_threshold_init* equals 0.9. k equals 1.2. The value of ΔD is varied from 0.01 to 0.05, then to 0.1. It can be observed that (i) Algorithm I achieves better performance than the case of no admission control when load is heavy ($\lambda > 0.9$), below that load level, throughput is lower than the case of no admission control, (ii) a smaller ΔD results in better performance; this is especially true when load is light or medium ($\lambda < 0.9$). Thus a low ΔD should be used in Algorithm I.

I also collected the result for video clip blocking rate as the result of admission control. This is the blocking rate for bottleneck traffic classes only. In Figure 4.3, the blocking rate for Algorithm I is plotted, in which *R_threshold_init* equals 0.9, average k equals 1.2, ΔD is varied between 0.01 and 0.1. It is observed that blocking rate

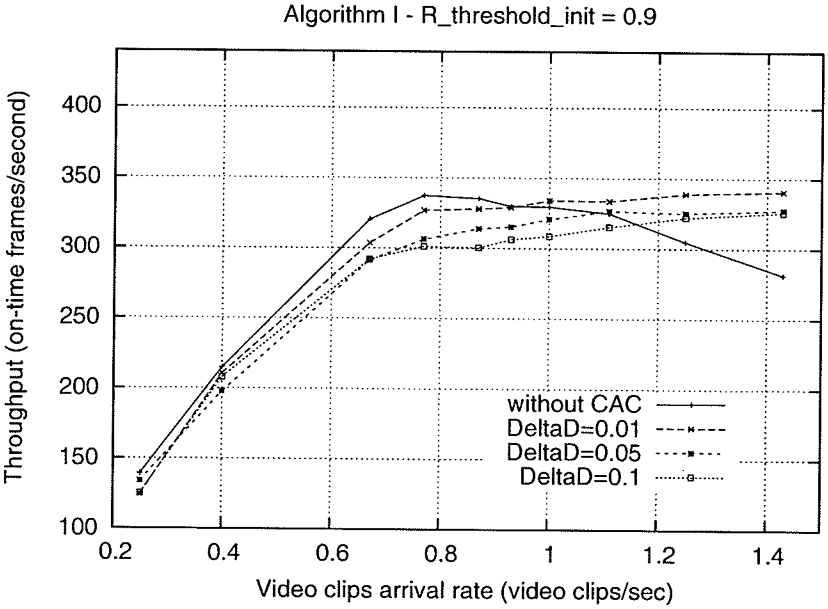


Figure 4.2: Algorithm I: effect of ΔD

keeps increasing as load is increased, at $\lambda = 1.43$, around half video clips are dropped. This corresponds to traffic intensity of one on the bottleneck. We will compare this result with that of Algorithm II, which is presented in the next subsection.

We conclude that Algorithm I can be used to prevent throughput degradation at heavy load. At medium and light load, it results in inferior performance to the case of no admission control. Between the two algorithm parameters, ΔD should be kept low while *R_threshold_init* has little effect on performance.

4.3.2 Performance Evaluation of Algorithm II

The second admission control algorithm, "Bandwidth Left", has three tunable parameters: C_{min}, C_{max0}, and ΔD. All are in terms of bandwidth, measured in bits

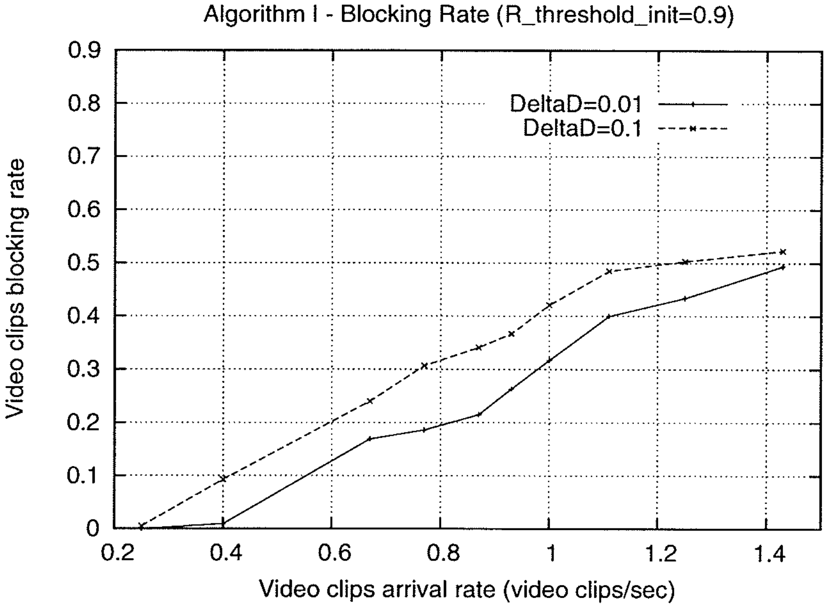


Figure 4.3: Algorithm I: blocking rate results

per second. To quantify the effect of each factor on performance, I first performed a 2^3 factorial design similar to the case of Algorithm I. The levels used for the three factors are:

- A: C_{min} = 0.3 and 0.45 Mbps
- B: C_{max0} = 2.5 and 6 Mbps
- C: ΔD = 0.025 to 0.2 Mbps

The levels for C_{min} represent insignificant video clip bandwidth requirements compared to the 10Mbps channel capacity. Because C_{min} is a threshold where video clips with estimated bandwidth lower than C_{min} is always accepted, it should be kept small. The levels chosen for C_{max0} correspond to one fourth and more than half of the bottleneck channel capacity respectively. These denote the total bandwidth of all outstanding video clips to a destination. ΔD is the adjustment amount for C_{max0}, and is kept small. Based on the simulation results from the eight experiments, the amount of variation in results explained by the factors and factor interactions is shown in Table 4.2. Among the three factors, both C_{min} and C_{max0} account for significant portion of variation in results, thus will be further studied. In contrast, ΔD accounts for little variation in results.

Table 4.2: 2^3 factorial design results for Algorithm II

Factor	A	B	C	AB	AC	BC	ABC
Effect (%)	25.87	59.24	0.81	1.85	1.08	2.58	8.55

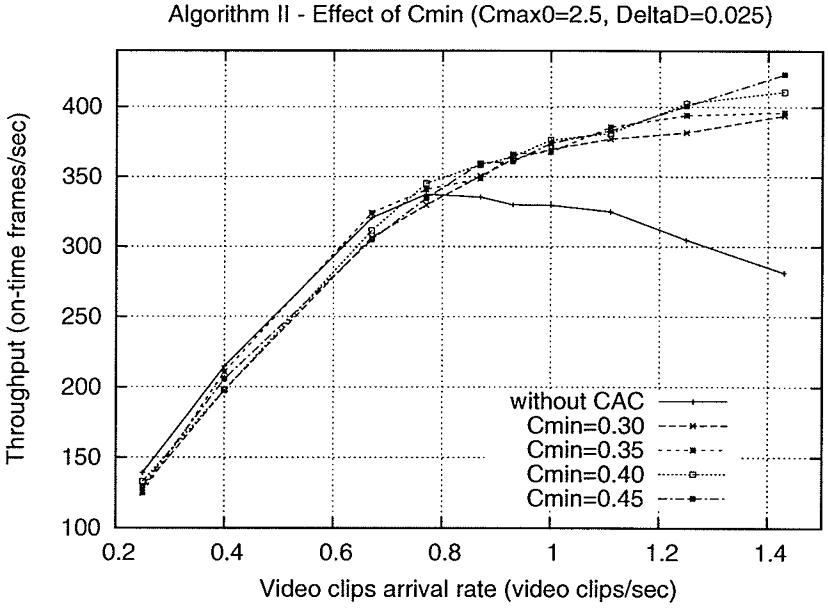


Figure 4.4: Algorithm II: effect of C_{min} ($C_{max0} = 2.5$)

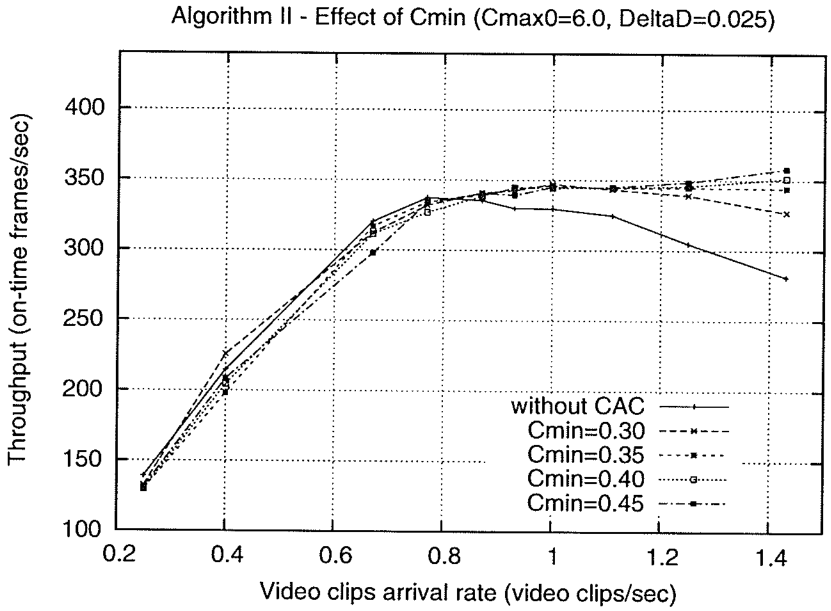


Figure 4.5: Algorithm II: effect of C_{min} ($C_{max0} = 6.0$)

To study the performance of Algorithm II, I first vary the value of C_{min}. Four values of C_{min}, namely 0.3, 0.35, 0.4, and 0.45 Mbps, are investigated. The values of other parameters are: $C_{max0} = 2.5$ and 6.0 Mbps, $\Delta D = 0.025$ Mbps, $k = 1.2$. In Figure 4.4 and 4.5, the bottleneck on-time throughput is plotted against the video clip arrival rate, for $C_{max0} = 2.5$ and $C_{max0} = 6.0$ Mbps respectively. The corresponding results for the case of no admission control are also shown. It can be observed that in both figures, all values of C_{min} result in significant improvement in performance over no admission control when $\lambda > 0.8$. The degradation in performance at medium and light load ($\lambda < 0.8$) is minor. Among the four values of C_{min}, 0.45 results in the best performance.

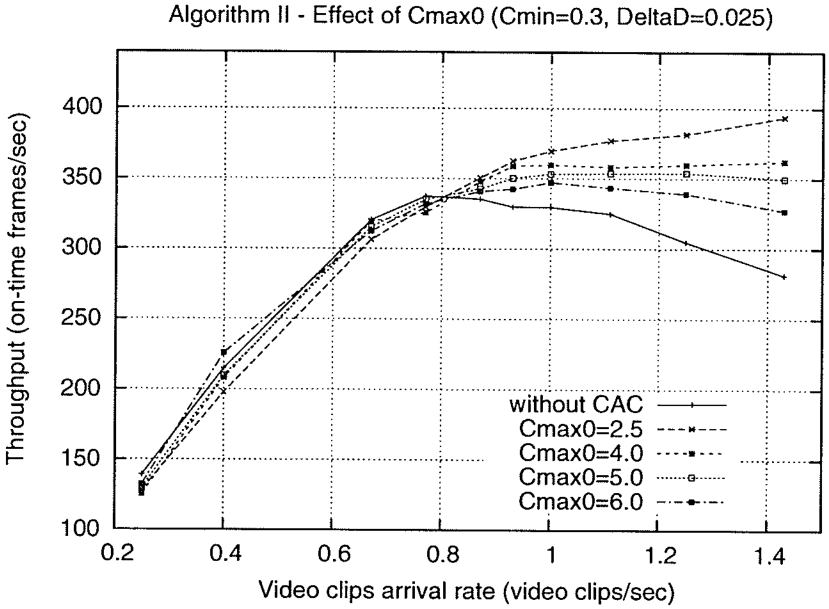


Figure 4.6: Algorithm II: effect of C_{max0} (C_{min} = 0.3)

I next study the performance of Algorithm II by varying C_{max0}. Four values of C_{max0} are experimented (2.5, 4, 5, and 6 Mbps). The values of other parameters are: C_{min} = 0.3 and 0.45 Mbps, ΔD = 0.025 Mbps, k = 1.2. The results are shown in Figure 4.6 and 4.7 for C_{min} = 0.3 and 0.45 Mbps respectively.

It can be observed that all experiments achieve superior performance to the case of no admission control. Throughput degradation is essentially prevented when C_{min} = 0.45 Mbps. In addition, C_{max0} has large impact on performance, a lower value of C_{max0} results in better performance when network load is heavy ($\lambda > 0.8$). This suggests a conservative upper bound C_{max0} should be used at each sender.

The effect of ΔD on performance is plotted in Figures 4.8 and 4.9, for C_{min} = 0.3 and C_{min} = 0.45 Mbps respectively. In these experiments, C_{max0} = 2.5 Mbps, k =

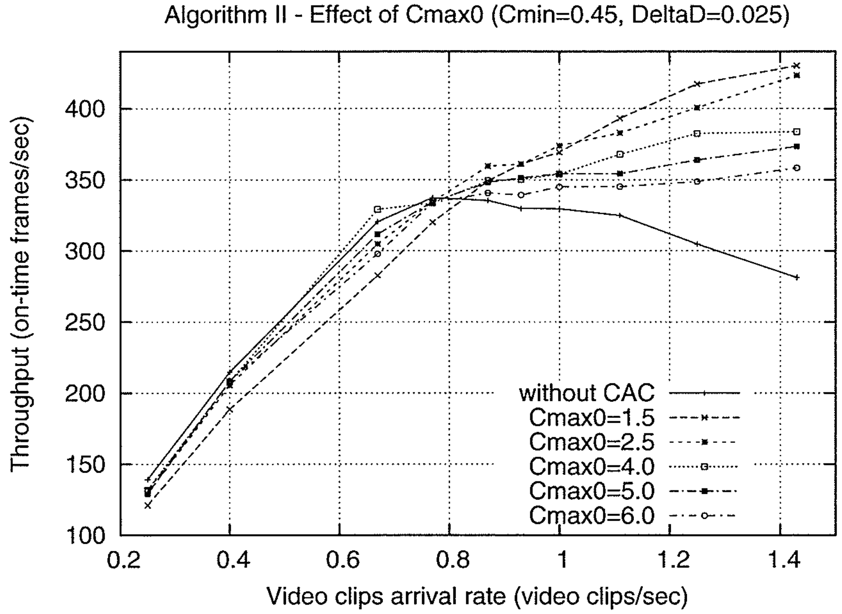


Figure 4.7: Algorithm II: effect of C_{max0} ($C_{min} = 0.45$)

1.2. It is observed that a smaller value of ΔD performs better than larger values.

Similar to the case of Algorithm I, I also collected the results on video clip blocking rate for bottleneck traffic. In Figure 4.10, the results for Algorithm II are plotted. It can be observed that comparing to the case of Algorithm I (see Figure 4.3), regardless of the load, the blocking rate is significantly lower. Comparing the on-time throughput results in this subsection with that for Algorithm I, a higher throughput can be achieved.

We conclude that admission control using Algorithm II can effectively prevent throughput degradation and significantly improve the video clip on-time performance. Among the algorithm parameters, both C_{min} and C_{max0} have large impact on performance. The values of C_{min} and ΔD should be small compared to bottleneck channel capacity, while C_{max0} can be at the same order as the bottleneck capacity, the value that is one fourth of the channel capacity resulted in the best performance.

Comparing the results of both admission control algorithms, it can be seen that in almost all cases, Algorithm II achieves better performance than algorithm I. This is because Algorithm II utilizes more information when making admission decisions - it makes use of the bandwidth requirement of an arriving video clip as well as the bandwidth requirements of all the outstanding video clips. In contrast, Algorithm I does not use any information of the incoming video clips or accepted clips; the admission control agent is less informed than that of Algorithm II when making admission decisions. The advantage of algorithm I is that it does not store the bandwidth requirement of all accepted video clips, thus is simpler to implement.

4.3.3 Effect of Update Interval Q

Besides the tuning parameters of the two algorithms discussed, I also studied the effect of the update interval Q on performance. Q is utilized by both algorithms, it determines how often the moving average "Positive ACK Rate" R is calculated. As mentioned in Section 4.1, a smaller Q will lead to a more accurate indication of current network condition, while a larger Q incurs less overhead. To evaluate the effect of update interval Q, I select the best case of Algorithm II in my experiments. The parameters are: $C_{min} = 0.45$, $C_{max0} = 2.5$, $\Delta\text{D} = 0.025$, and $k = 1.2$. Q is varied among 1 second, 3 seconds, and 5 seconds. The simulation is run for 300 seconds. The results are shown in Figure 4.11.

It can be observed that Q does not have great impact on performance. A smaller value of Q leads to somewhat better performance when load is heavy. When load is light, a larger Q is slightly better. Consider both, a smaller Q is recommended if affordable in terms of implementation.

4.3.4 Summary

In this chapter, two application-layer admission control algorithms are developed and evaluated. Both algorithms are found to result in better performance than the case of no admission control when network is heavily loaded. The performance degradation at medium and light load is small. Between the two algorithms, the one that utilizes the bandwidth requirements of both arriving transmission request as well as previous accepted outstanding requests achieves better performance. The one that relies solely on network congestion condition feedback does not perform as well. This indicates that with bandwidth requirement information, the admission control entity can make more informed admission decisions and achieve better performance.

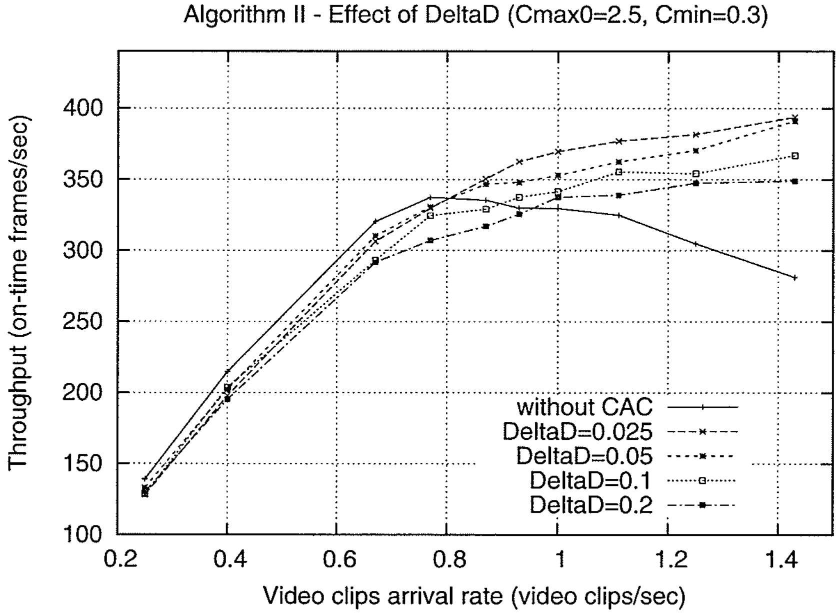


Figure 4.8: Algorithm II: effect of ΔD ($C_{min} = 0.3$)

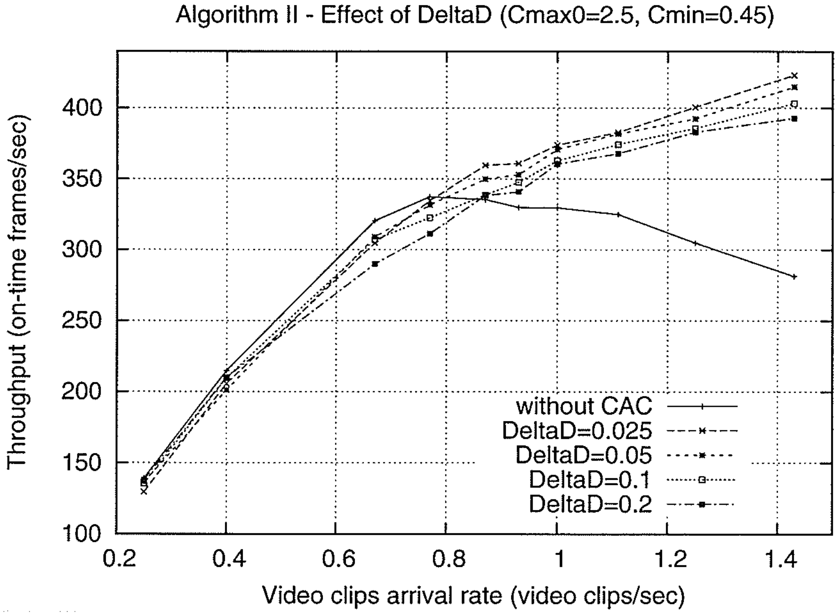


Figure 4.9: Algorithm II: effect of ΔD ($C_{min} = 0.45$)

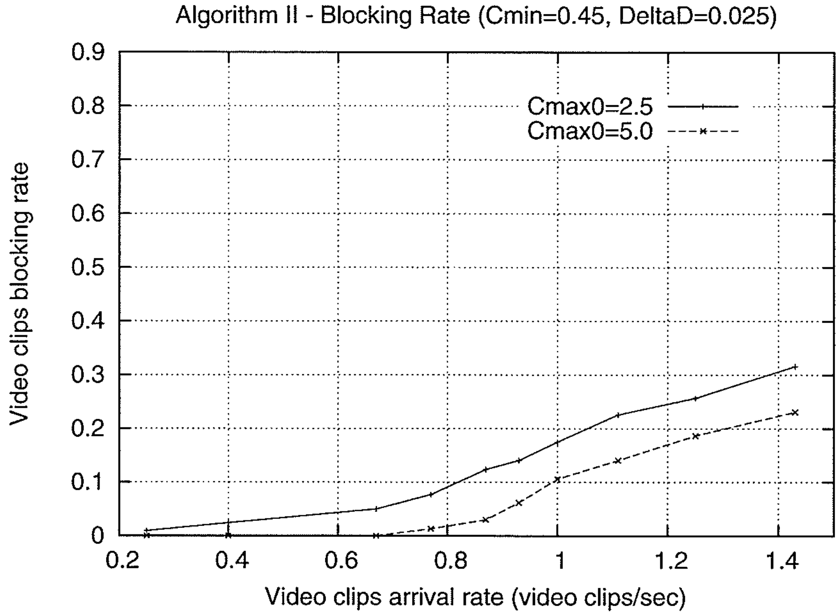


Figure 4.10: Algorithm II: blocking rate results

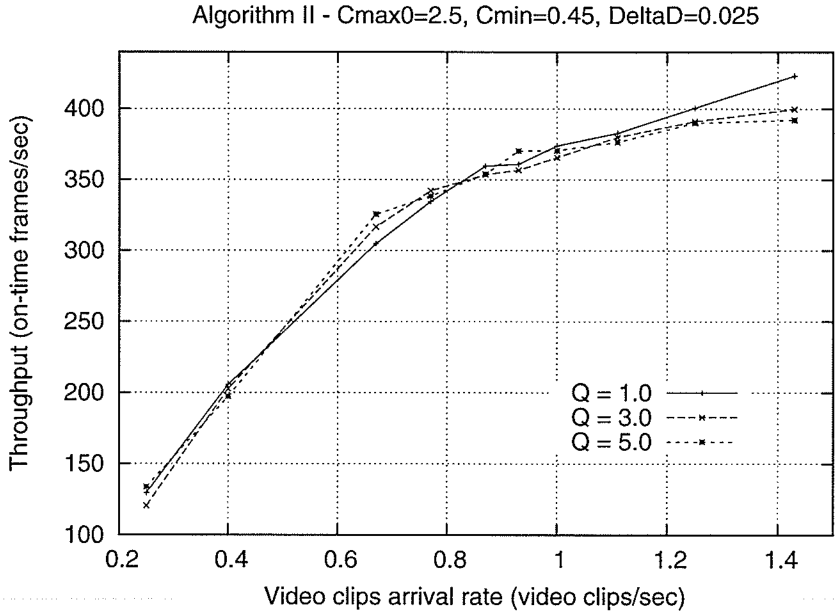


Figure 4.11: Effect of update interval Q

Chapter 5

Deadline Assignment for Multimedia Delivery

The admission control algorithms presented in the last chapter can be used to provide good performance in multimedia delivery when network is congested, especially when network is overloaded. Besides admission control, another resource management strategy investigated in this thesis is differential deadline assignment for multimedia traffic. Two schemes are studied. In the first scheme, "frame-type differential deadline" assignment is introduced and studied. The scheme relies on the fact that in MPEG video, different types of frames have different level of importance. In this scheme, different deadline urgency levels are assigned to different types of frame such that during deadline-based channel scheduling, the more important types of frames are given higher priority. This may improve the on-time performance of the more important frames.

In the second scheme, the case when both multimedia traffic and non-multimedia real-time traffic share the same bottleneck is considered. In this case, the multimedia traffic is continuous in nature; frames arrive at the network at a constant rate, for example, one frame every 40 ms. In contrast, the non-multimedia real-time traffic does not have the continuous nature, therefore, the former may be more demanding on network resources than the latter. The idea of the second scheme is to give real-time multimedia traffic more urgent deadlines than non-multimedia real-time traffic. This way during deadline-based channel scheduling, multimedia traffic is given preferential treatment, which may result in better performance than without differential deadlines. We present and evaluate these two schemes in this chapter.

Before describing these two schemes, I first introduce two types of deadlines: scheduling deadlines and playback deadlines. The distinction between these two is important to understand the two differential deadline assignment schemes.

5.1 Scheduling Deadlines and Playback Deadlines

In a deadline-based network, each real-time document or ADU is associated with an ADU deadline, which represents the absolute time at which the ADU should be delivered at the receiving application. How to assign ADU deadlines is application-dependent. For example, in real-time multi-player online games such as first-person-shooter (FPS) games, state update messages are exchanged between game clients and the game server. To maintain interactivity, the end-to-end deadline, i.e., the time duration between the ADU arrival time and the ADU deadline, should be maintained around 150 to 200ms [27]. In comparison, in a video on demand multimedia application, the end-to-end deadline can be 10 seconds.

When an ADU is passed down to the network layer, an ADU may be segmented into more than one packet. The ADU deadline is translated into a packet deadline; this deadline is carried by packets and used by routers for channel scheduling. It is during this translation where differential deadline assignment can be enforced. The idea is to tell apart two types of deadlines: "deadline for scheduling", which is a network-layer deadline carried by packets, and "deadline for playback", which is also carried in packets and is the ADU deadline at the application layer. The scheduling deadline is what routers recognize and use for channel scheduling. The playback deadline is specified by users and determines the on-time ADU delivery performance. One can keep the playback deadline intact, and have it conveyed to the receiver, but change the urgency level of the scheduling deadline, so as to improve the on-time performance.

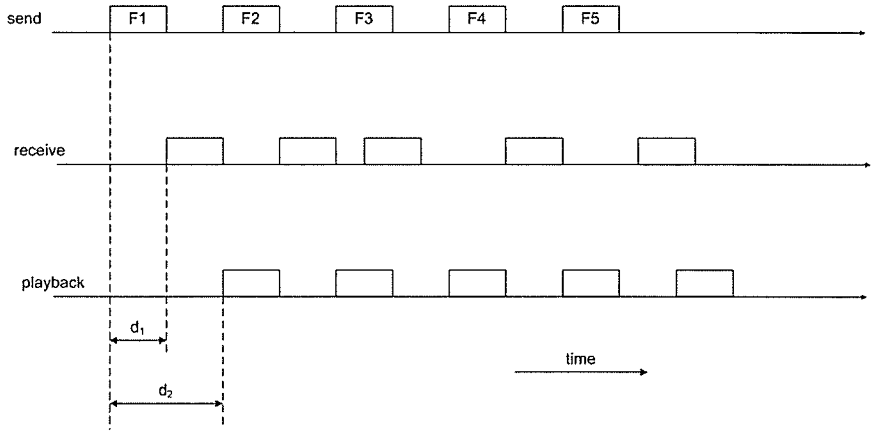


Figure 5.1: Scheduling deadlines vs. playback deadlines

In the case of multimedia, the scheduling deadline can represent the time at which the frame packets "should" arrive at the receiver's network layer, while the playback deadline is used to determine frame on-time rate at the application layer. These two types of deadlines for multimedia is illustrated in Figure 5.1.

In this graph, three time lines are drawn. The top one occurs at the sender, the other two occur at the receiver. A sequence of frames is sent from the sender at regular time intervals. For simplicity, in this graph, it is assumed that every frame can be accommodated in a packet. For the first frame, assume the sending time is zero, then d_1, the time duration between when the frame is sent and when the frame packet should be received is the value of its scheduling deadline. This frame is due to be played out some time later, in the graph, d_2 indicates the value of the frame's end-to-end playback deadline. For multimedia, because of the isochronous nature of frame arrival and playback processes, frames in a video clip normally have a fixed end-to-end playback deadline.

Having defined the two types of deadlines, I next present the two differential deadline assignment schemes that are developed to better support multimedia delivery over deadline-based networks. In my discussion of deadline assignment schemes, the "deadline" will be the deadline for scheduling.

5.2 Frame-Type Differential Deadline Assignment

Because of the encoding scheme used by MPEG video, different encoded frames in a MPEG video have different level of importance. For MPEG-4, I-frames are considered more important than P- or B-frames. Giving I-frames higher priority than P- and B-frames may reduce the video quality degradation caused by packet loss. This can be implemented by assigning more urgent deadlines to I-frames. In this section, on-time performance when different types of frames are given different level of deadline urgency is investigated using simulation. The performance model is first presented. Experiments and results are described afterwards.

In my performance model, document deadlines are modeled with respect to the end-to-end latency. A document deadline is given by $d = \textit{arrival time} + kx$, where arrival time is the document arrival time at the network, x is the end-to-end latency when with no fragmentation and no queueing. By varying the deadline parameter k, different deadline urgency can be modeled.

In the deadline assignment schemes discussed so far, different types of frames are given the same urgency level, i.e., the same value of k,. I refer to this deadline assignment scheme "frame-type indifference" deadline assignment. In my "frame-type differential" deadline assignment scheme, I determine and assign different frame deadline urgency levels depending on the type of a frame. To make these two deadline assignment schemes comparable, I assume that they have the same "deadline demand". For a video clip, "deadline demand" is defined as the sum of the end-to-end deadlines of all the frames. Let $D(i)$ be the end-to-end deadline assigned to frame i, $i \geq 1$, then the "deadline demand" of a video clip will be $\sum_{i=1}^{n} D(i)$, where n is the number of frames in the video clip. Let k_I , k_P , and k_B represent the deadline parameters for I-, P- and B-frames respectively in frame-type differential deadline assignment. Let n_I , n_P , and n_B represent number of I-, P-, and B-frames in a video clip respectively. Thus, $n = n_I + n_P + n_B$. Let N denote the number of hops along the video clip path and y_I , y_P , and y_B represent the average sizes of I-, P-, and B-frames respectively. With "frame-type indifference" scheme, a video clip's "deadline demand" can be calculated by $k * \sum_{i=1}^{n} x_i$, where x_i is the end-to-end latency of frame i. x_i consists of the end-to-end store-and-forward transmission delay of this frame and the end-to-end propagation delay. Based on this information, differentiated deadline assignment scheme can be deducted from Equation 5.1.

$$\begin{aligned} k * \sum_{i=1}^{n}\{\sum_{j=1}^{N}\frac{y_i}{C_j} + \sum_{j=1}^{N} x_{p_j}\} &= k_I * \sum_{i=1}^{n_I}\{\sum_{j=1}^{N}\frac{y_{I_i}}{C_j} + \sum_{i=1}^{N} x_{p_j}\} \\ &\quad + k_P * \sum_{i=1}^{n_P}\{\sum_{j=1}^{N}\frac{y_{P_i}}{C_j} + \sum_{i=1}^{N} x_{p_j}\} \\ &\quad + k_B * \sum_{i=1}^{n_B}\{\sum_{j=1}^{N}\frac{y_{B_i}}{C_j} + \sum_{i=1}^{N} x_{p_j}\} \end{aligned} \tag{5.1}$$

The left-hand side of Equation 5.1 denotes the deadline demand for the frame-type indifference case, while the right-hand side denotes the deadline demand for the frame-type differential case. With just this equation of three unknowns, the values of k_I , k_P, and k_B are not fixed. Rather different sets of values can be taken for different differential deadline assignments. For example, one can choose these k values such that different priorities among the frame types are enforced. More specifically, because I-frames are more important than P-frames, and P-frames are more important than B-frames, the three deadline parameters can follow $k_I < k_P < k_B$; this way I-frames have the highest priority in channel scheduling, while P- and B- frames are given lower priority. In what follows, I use simulation to evaluate the performance of frame-type differential deadline assignment schemes. I will compare the results with that of the frame-type indifference case.

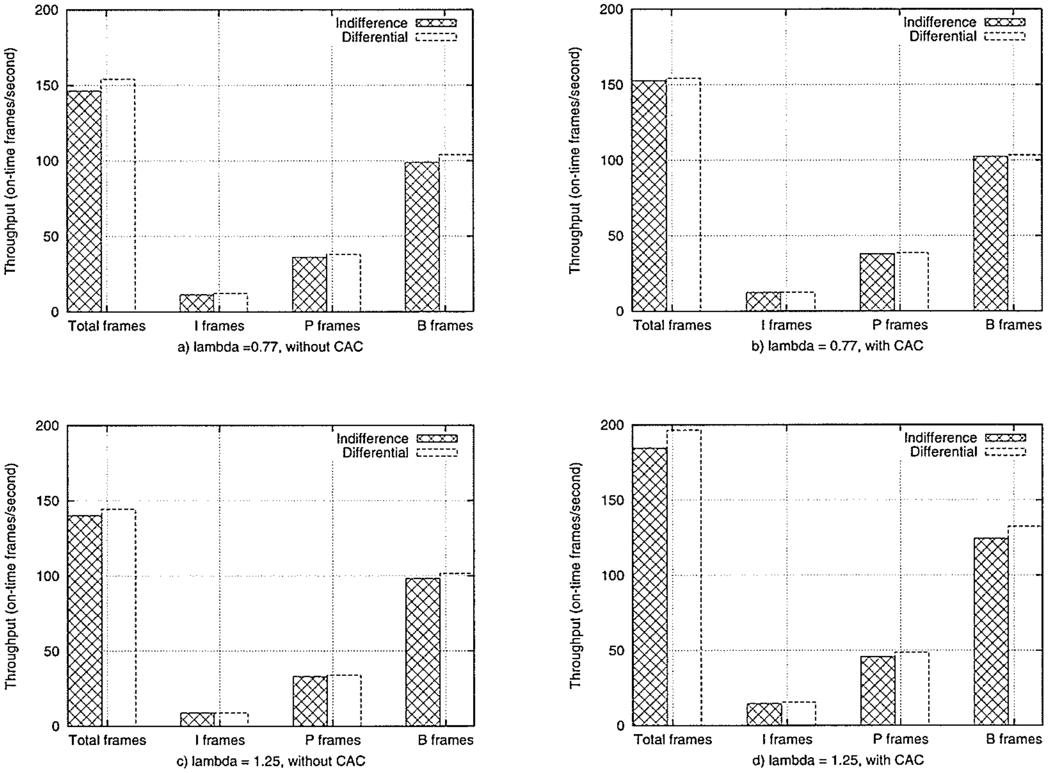
The same network and traffic models as in Section 3.1 are used. There are 500 frames in each movie trace. Within each movie trace, there are 42 I-frames, 125 P-frames, and 333 B-frames. From the statistics of the trace files used in my simulation, the overall average frame size is 2884 bytes. Average I-, P-, and B-frame sizes are 5409, 3277, and 2418 bytes respectively. Bring all these numbers to Equation 5.1, assuming $k = 1.2$ in the indifference scheme, a set of k values for the differential scheme are chosen. It fulfills the conditions: $k_I < k_P < k_B$, and $k > 1$. In this set, $k_I = 1.05$, $k_P = 1.17$, , and $k_B = 1.23$. They are shown in Table 5.1. Note that selecting optimal k values so that the performance of video delivery can be optimized is a challenging problem and is considered a future work of this study.

Table 5.1: k in frame-type differential deadline assignment

Scheme	k_i	k_p	k_b
Indifference	1.2	1.2	1.2
Differential	1.05	1.17	1.23

These k values are used to compare the on-time performance when using these two different deadline assignment schemes.

I experimented with close to half of the traffic classes that go through the bottleneck link. With frame-type indifference and frame-type differential schemes respectively. The bottleneck on-time throughput for this half amount of traffic is used as the performance measure. In addition to aggregated on-time throughput, I also collected the on-time throughput per frame type. Two levels of video clip arrival rates are experimented, these are $\lambda = 0.77$ and $\lambda = 1.25$ respectively. They represent heavy and overload conditions respectively. For both levels of λ, experiments are performed with or without admission control. Results from these four combinations are shown by a), b), c) and d) in Figure 5.2 respectively.



a) lambda =0.77, without CAC

b) lambda = 0.77, with CAC

c) lambda = 1.25, without CAC

d) lambda = 1.25, with CAC

Figure 5.2: Frame-type differential deadline assignment performance

In this Figure, "Indifference" identifies the case when frame-type indifference deadline assignment is used. "Differential" identifies the case when frame-type differential deadline assignment is in place. It can be observed that the frame-type differential deadline assignment scheme performs better than using uniformed k. Not only is the total throughput improved, throughputs of all of I-, P-, and B-frames are improved as well. Among the three frame types, the extent at which on-time throughput is increased for I-frames is a bit higher than or about the same as those for P- and B-frames. Thus by giving I-frames more urgent scheduling deadlines, the throughput of all three types can receive some gain. This suggests that frame-type differential deadline assignment scheme can be used to gain better performance in delivering multimedia traffic over a deadline-based network.

5.3 Sharing with Real-Time Non-Multimedia Traffic

The second differential deadline assignment scheme investigated in this thesis is not concerned with differential treatment of different types of frames within a multimedia stream itself, rather studies differential treatment between multimedia and non-multimedia real-time traffic. Example non-multimedia real-time traffic includes bids in an on-line auction, stock-trades in on-line trading, stock quote updates, and time-sensitive business documents in electronic commerce applications. Differ from multimedia data where frames are generated in a continuous manner, e.g., one frame every 25 ms, and end-to-end playback deadlines are similar to each other, non-multimedia real-time documents are not continuous in nature and may have vastly distinct deadlines.

When these two types of real-time data share the same limiting network resource, e.g., a bottleneck channel, it might be possible to give multimedia data preferential treatment while still providing good support to non-multimedia real-time traffic. The preferential treatment in deadline-based networks refers to more urgent deadlines, thus higher priority in deadline-based channel scheduling. In this section, experiments are carried out to study the performance improvement out of one such scheme. I first describe the simulation model and experiments. Simulation results are presented afterwards.

The same network model as that in Section 3.1 is used. The traffic model is slightly changed from an all-video-clip one to a mixture of video clips and discrete documents. In particular, among the 13 traffic classes that pass through the bottleneck, 6 of them are dedicated to movie transmission, and are referred to as the "foreground traffic". The other 7 classes carry discrete real-time documents and are referred to as "background traffic". All other non-bottleneck traffic classes contain discrete real-time data as well. The traffic model for discrete real-time documents is similar to that of the movie ones, except that the frames are now individual ADUs, and the ADU inter-arrival time is assumed to be exponentially distributed, while in movie clips, the frame inter-arrival times are constant. For foreground traffic, the video clip (scheduling) deadline is determined based on deadline parameter k and the largest frame size in a clip, the playback deadline is 150 ms after when a frame is sent into the network. For background traffic, an ADU's scheduling deadline and playback deadline are the same, both are calculated using the deadline parameter k and the ADU size.

Thus Foreground traffic represents the multimedia real-time traffic, and the background traffic represents the non-multimedia real-time traffic. The differential deadline assignment between foreground and background traffic is implemented as follows. In three experiments, foreground traffic is given three different urgency levels: the average values of k are 1.5, 1.2 and 1.05 respectively. Background traffic, on the other hand, always has the same urgency: the average k is fixed at 1.5. The movie arrival rate and ADU arrival rate are selected such as the offered load on the bottleneck channel is 91%. The same simulation method is used as in Section 3.1.6. Bottleneck on-time throughputs for both foreground and background traffic are used as the performance metric. Results from the simulation are reported in Table 5.2.

Table 5.2: Differential deadline assignment for multimedia traffic

Traffic	1.5/1.5	1.2/1.5	1.05/1.5
Foreground	147.15	149.59	153.00
Background	130.17	128.89	137.72

In this table, 1.5/1.5 denotes the case before differential deadline assignment is in place. 1.2/1.5 denotes the case when foreground traffic is assigned more urgent scheduling deadlines than in the 1.5/1.5 case, while the background traffic keeps having the same deadline urgency. 1.05/1.5 case treats foreground multimedia traffic with even higher deadline urgency. It can be observed that as the level of deadline urgency increases, foreground multimedia receives higher and higher on-time throughput. For background traffic, the on-time throughput results are not as clear-cut; it first dropped slightly, and then increased significantly. Similar trends are observed for more experiments with different arrival rates.

We conclude that when foreground multimedia traffic is competing for the channel bandwidth with non-multimedia background traffic, assigning more urgent scheduling deadlines to foreground traffic would lead to increased on-time throughput of foreground traffic. At the same time, the on-time performance of the background non-multimedia real-time traffic is not significantly affected; in some cases, its on-time throughput even increases. Thus differential deadline assignment scheme may be used to better support multimedia data when it shares bottleneck channel with other non-multimedia real-time traffic.

Chapter 6

Conclusion and Future Work

In this thesis research, how to effectively and efficiently deliver multimedia documents over deadline-based networks is studied. To tackle this problem, two complementary approaches are investigated.

First, to ease the possible congestion within deadline-based networks, and to provide good multimedia delivery performance when network load is high, two end-system based admission control algorithms are developed. Both algorithms infer the network congestion condition through end-to-end acknowledgements. Simulation results show that both algorithms lead to performance improvement at heavy load and overload conditions. Between the two algorithms, the one that makes use of the bandwidth requirements of the arriving video clip and currently outstanding video clips achieved better performance.

Second, differential deadline assignment schemes are devised to better support multimedia delivery over deadline-based networks. For each real-time frame, a scheduling deadline and a playback deadline is told apart. Based on the observance that with current multimedia encoding techniques, different types of frames have different levels of importance, different types of frames are assigned different scheduling deadlines such that the more important frames receive higher priority in deadline scheduling. This frame-type differential deadline assignment scheme is evaluated against the frame-type indifference deadline assignment scheme, and is shown to receive noticeable gain in performance.

The other differential deadline assignment scheme investigated in this thesis deals with scenarios when multimedia real-time traffic is competing network resources with other non-multimedia real-time traffic. By giving former more urgent scheduling deadlines, the isochronous nature of multimedia data is better preserved, while at the same time, the performance of non-multimedia traffic is not significantly affected.

The contributions of this research include the development of two end-system admission control algorithms to deliver multimedia over deadline-based networks, as well as differential deadline assignment schemes that can be used to further enhance the performance of multimedia delivery.

6.1 Future Work

In Section 5.2, the frame-type differential deadline assignment scheme appears promising in improving performance. However the method used in choosing the set of differential deadlines was rather arbitrary. It will be useful and interesting to study how to choose the set of differential deadlines so that the performance gain over frame-type indifference case can be maximized. In the experiments to evaluate the performance of the two developed admission control algorithms, Algorithm II takes into bandwidth requirement in making admission control decisions. In current experiments, only average bit rates are used, a possible future research is to come up with better estimates than the first moment on video clip bit rate so that the on-time performance can be maximized.

Bibliography

[1] J. G. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaing with content delivery networks. In *Proceedings of IEEE INFOCOM 2002*, volume 3, pages 1736 – 1745, June 2002.

[2] M. Baldi. Interactive multimedia networking. In *Proceedings of the 8th International Conference on Telecommunications, 2005. ConTEL 2005*, volume 2, pages 683 – 684, June 15 – 17 2005.

[3] Yin Bao and Adarshpal S. Sethi. Performance-driven adaptive admission control for multimedia applications. In *ICC '99: Proceedings of IEEE International Conference on Communications*, volume 1, pages 199–203. IEEE Press, October 1999.

[4] Dimitri Bertsekas and Robert Gallager. *Data networks.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2 edition, 1992.

[5] Jill M. Boyce and Robert D. Gaglianello. Packet loss effects on mpeg video sent over the public Internet. In *Proceeding of the ACM Multimedia 98*, pages 181–190, September 1998.

[6] C.J. Chang, L.F. Lin, S.Y. Lin, and R.G. Cheng. Power-spectrum-based neural-net connection admission control for multimedia networks. In *IEEE Proceedings - Communication*, volume 149 of *2*, pages 70–76. IEEE Press, April 2002.

[7] Kameswari Chebrolu and Ramesh R. Rao. Selective frame discard for interactive video. *Proceeding of IEEE International Conference on Communications*, 27(1):4097–4102, June 2004.

[8] Jonathan Davidson, James Peters, Manoj Bhatia, Satish Kalidindi, and Sudipto Mukherjee. *Voice over IP Fundamentals.* Cisco Press, second edition, July 2006.

[9] Mohaned A. El-Gendy, Abhijit Bose, and Kang G. Shin. Evolution of the Internet QoS and support for soft real-time applications. In *Procedings of the IEEE*, volume 91 of *7*, pages 1086–1104, July 2003.

[10] V. Elek, G. Karlsson, and R. Ronngren. Admission control based on end-to-end measurements. In *Proceedings of IEEE Infocom 2000*, pages 623–630, March 2000.

[11] Kevin Fall and Kannan Varadhan. *The ns Manual*, April 2002. http://www.isi.edu/nsnam/ns.

[12] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. *RFC2597 : Assured Forwarding PHB Group.* The Internet Engineering Task Force, Jun 1999.

[13] Y. Thomas Hou, Dapeng Wu, Wenwu Zhu, Hung-Ju Lee, Tihao Chiang, and Ya-Qin Zhang. An end-to-end architecture for MPEG-4 video streaming over

the Internet. In *Proceeding of International Conference on Image Processing*, volume 1, pages 254–257, October 1999.

[14] Jie Huang, Charles Krasic, Jonathan Walpole, and Wu chi Feng. Adaptive live video streaming by priority drop. In *Proceeding of IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 342–347, July 2003.

[15] S. Jagannathan, A. Tohmaz, A. Chronopoulos, and H.G. Cheung. Adaptive admission control of multimedia traffic in high-speed networks. In *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, pages 728–733. IEEE Press, October 2002.

[16] R. K. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley and Sons, Upper Saddle River, NJ, USA, 1991.

[17] Sugih Jamin, Peter B. Danzig, Scott Shenker, and Lixia Zhang. A measurement-based admission control algorithm for integrated services packet networks. In *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 2–13. ACM Press, 1995.

[18] Sugih Jamin, Scott J. Shenker, and Peter B. Danzig. Comparison of measurement-based admission control algorithms for controlled-load service. In *INFOCOM'97: Proceedings of IEEE International Conference on Computer Communications*, volume 1, pages 147–152. IEEE Press, June 1997.

[19] Sooyong Kang and Heon Y. Yeom. Statistical admission control for soft real-time

vod servers. In *SAC '00: Proceedings of the 2000 ACM symposium on Applied computing*, pages 579–584. ACM Press, 2000.

[20] Edward W. Knightly, Dallas E. Wrege, J&#246;rg Liebeherr, and Hui Zhang. Fundamental limits and tradeoffs of providing deterministic guarantees to vbr video traffic. In *SIGMETRICS '95/PERFORMANCE '95: Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 98–107. ACM Press, 1995.

[21] Rob Koenen. *Overview of the MPEG-4 Standard.* Internationl Organisation for Standardisation, v.21 edition, March 2002. http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.

[22] James F. Kurose and Keith W. Ross. *Computer Networking - A Top-Down Approach Featuring the Internet.* Addison Wesley, Second edition, 2003.

[23] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan. Transporting compressed video over atm networks with explicit-rate feedback control. *IEEE/ACM Transaction of Networking*, 7(5):710–723, 1999.

[24] Teck Kiong Lee, M. Zukerman, and R.G. Addie. Admission control schemes for bursty multimedia traffic. In *INFOCOM 2001:Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 478–487. IEEE Press, April 2001.

[25] D. Lin and R. Morris. Dynamics of random early detection. In *Proceedings of ACM SIGCOMM'97*, pages 127 – 137, October 1997.

[26] Yanni Ellen Liu. *Deadline-based network resource management.* PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2003.

[27] Yanni Ellen Liu, Jing Wang, M. Kwok, J. Diamond, and M. Toulouse. Fps game performance in wi-fi networks. In *Proceedings of the 4th International Game Design and Technology Workshop and Conference (GDTW 2006)*, November 2006.

[28] Yanni Ellen Liu and Johnny W. Wong. Deadline based channel scheduling. In *Proceedings of the IEEE Global Telecommunications Conference (Globecom'01)*, pages 2358–2362, San Antonio, USA, November 2001. IEEE Press.

[29] Yanni Ellen Liu and Johnny W. Wong. Admission control in deadline-based network resource management. In *Proceedings of the 23rd IEEE International Performance, Computing, and Communications Conference (IPCCC'2004)*, pages 95–102, Phoenix, Arizona, April 2004. IEEE Press.

[30] P. Pancha and M. El Zarki. Bandwidth-allocation schemes for variable-bit-rate mpeg sources in atm net-works. *IEEE Transaction on Circuits and Systems for Video Tech.*, 3(3):190–198, 1993.

[31] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks. *IEEE/ACM Transactions on Networking*, 2(2):137 – 150, April 1994.

[32] Reza Rejaie, Mark Handley, and Deborah Estrin. Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In

Proceeding of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies – INFOCOM '99, volume 3, pages 1337–1345. IEEE Press, March 1999.

[33] Stefan Saroiu, Krishna P. Gummadi, Richard J. Dunn, Steven D. Gribble, and Henry M. Levy. An analysis of Internet content delivery systems. *SIGOPS Operating System Review*, 36(SI):315–327, 2002.

[34] Patrick Seeling, Martin Reisslein, and Beshan Kulapala. Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial. *IEEE Communications Surveys & Tutorials*, 6(3):58–78, 2004. http://trace.eas.asu.edu/tracemain.html.

[35] Shu Tao and Roch Guérin. Application-specific path switching: a case study for streaming video. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 136–143. ACM Press, 2004.

[36] Johnny W. Wong and Yanni Ellen Liu. Deadline based network resource management. In *Proceedings of the Ninth International Conference on Computer Communications and Networks (ICCCN'00)*, pages 264–268, Las Vegas, Nevada, October 2000. IEEE Press.

[37] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha. Streaming video over the Internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):282–300, March 2001.

[38] Xipeng Xiao and Lionel M. NI. Internet QoS: A big picture. *IEEE Network*, pages 8–18, 1999.

[39] Weibin Zhao, David Olshefski, and Henning Schulzrinne. Internet quality of service: an overview. Technical Report CUCS-003-00, Columbia University, 2000. http://www.cs.columbia.edu/ hgs/netbib/.

[40] Roger Zimmermann and Kun Fu. Comprehensive statistical admission control for streaming media servers. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 75–85. ACM Press, 2003.

[41] Artur Ziviani, José Ferreira de Rezende, Otto Carlos Muniz Bandeira Duarte, and Serge Fdida. Improving the delivery quality of MPEG video streams by using differentiated services. In *Proceeding of the 2nd European Conference on Multiservice Networks – ECUMN'2002*, pages 107–115, Colmar, France, April 2002.

[42] Artur Ziviani, Bernd E. Wolfinger, José Ferreira de Rezende, Otto Carlos Muniz Bandeira Duarte, and Serge Fdida. Joint adoption of QoS schemes for MPEG streams. *Multimedia Tools and Applications*, 2003.