



Enhanced Adaptive Mechanisms to Improve QoS in Distributed Multimedia Applications

Mirza Anwarullah Baig¹, Dr. Syed Abdul Sattar²

Ph.D. Scholar Dravidian University, Kuppam, A.P. India.
anwarntfs@gmail.com¹

Professor & Dean of Academics, Royal Institute of Technology & Science, Chevella RR Dist.
syedabdulsattar1965@gmail.com²

Abstract

Recent advances in computer communications, especially highly interactive distributed applications and those relying on the transfer of multimedia information and in particular, continuous media flows. It is essential that QoS is guaranteed in a system, considering distributed system area, the transport protocol and the multiservice network. Quality of Service (QoS) in distributed multimedia systems provides a unifying theme on which the functions and facilities of the new integrated standards can be constructed. The objective of this paper is to focus on several QoS parameters at different levels and Enhanced Adaptive Mechanisms proposed in order to improve the performance in distributed multimedia applications.

Keywords

Transmission Control Protocol (TCP); User Datagram Protocol (UDP); Quality of Service (QoS) MPEG.



Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 12, No. 10

editor@cirworld.com

www.cirworld.com, www.ijctonline.com



I. INTRODUCTION

Nowadays most networks, operating systems, and databases, many resources are shared, the activities of other users can lead to large and sudden variations in resource accessibility. Due to these variations, QoS must be adapted during a presentation as smoothly and quickly as possible. In simple best-effort approaches every stage of a video pipeline, for instance simply drops data when there are insufficient resources to process it (not enough network bandwidth, buffer overflow, CPU too busy, and so on). This approach is very inefficient because all resources used for processing data that is dropped by a later pipeline stage are wasted. A more sophisticated form of adaptation is needed. Supporting complex content presentations and providing the user with control overview parameters requires a sophisticated system design.

The application needs to run several video and audio pipelines at a time keeping them synchronized. Starting up and stopping as well as concatenating streams needs to be scheduled on real time basis. Connections to the servers of the respective frames need to be managed. In a single presentation, the quality of component streams can differ considerably, when some data is retrieved from a local server and some over a congested network transmitting continuous media data in real-time over the Internet does not allow the use of a reliable protocol such as TCP. Because packets may be loosed, TCP tries to resend them. The resent packets, again, can be lost, and so on. Hence, transmitting a particular piece of data can take arbitrarily long. For continuous media, this delay is not acceptable. Not only is late data itself useless, but with a reliable protocol it also delays the delivery of subsequent data, potentially stalling the entire pipeline. Because of these problems, an unreliable UDP is preferred, and the application must deal with packet loss and out-of-order delivery.

Limited network bandwidth and disk space require compression at least of the video data. Most compression algorithms are lossy, making QoS management more complicated. Moreover, peculiarities of the technique used have a major impact on the design of the multimedia application. The MPEG standard of video compression, for instance uses inter-frame decoding, such that few frames can only be decoded if surrounding reference frames are already available. If reference frame is dropped, the frames belongs to it are also useless too. Any stage of the video pipeline should take these dependencies into account when making dropping decisions.

Because it is necessary for the user to control the trade-offs between several QoS dimensions, the adjustment of several parameters such as frame rate and image resolution must be supported. The goal is to have the configuration to provide the good quality using the resources that are presently available as the notion of best being use and task specific. Coming close to this goal requires good heuristics. Moreover, even assessing quality based on a QoS specification is a non-trivial problem, which can be addressed using a formal error model.

II. QOS CONTROL

QoS control is a useful technique for allowing efficient resource utilization. It achieves this goal by specifying appropriate priorities among different ways of allocating them. Moreover, these systems should be able to operate in an environment with shared resources. To control resource management it is necessary to measure and assess presentation quality. Quality of Service (QoS) here denotes performance characteristics of multimedia systems. Such characteristics include low level parameters such as throughput, transmission delay, and reliability as well as higher-level dimensions such as spatial resolution, frame rate (temporal resolution), and color depth. For continuous media streams, delay jitter and synchronization are important too. Other elements of QoS include the cost of a presentation and priorities of different streams.

The following sections give an overview of the notion of QoS on several levels and how they are related to each other. After that, the main aspect of resource management is discussed. Implementation mechanisms such as admission testing, resource reservation, scheduling, policing, are debated and then Enhanced Adaptive Mechanism have been proposed in order to improve the performance in multimedia applications.

The degree to which performance guarantees can be provided depends on the system support for these resource management tasks. Controlling several QoS parameters rather than having them hardcoded in multimedia applications helps to exploit resources more efficiently by allowing finer adjustment to the needs of the user. To support compositions of several continuous media streams, presentation descriptions and synchronization mechanisms are needed.

A. Quality of Service Levels

The notion of QoS is an important issue. Vogel et al. [1] and Hutchison et al. [2] Give overviews of QoS in distributed multimedia systems. QoS has been discussed on several levels. The following sections categorize them as *user-perception* and *resource-level* QoS.

User-Level QoS

The ultimate goal of a quality notion is to represent the value of a presentation to the user. *User-level* QoS is an assessment based on several issues:

User Perception: Can errors in the presentation be perceived by humans? For example, it is likely that nobody can tell the difference between a video displayed with 60 frames per second and one with 80 frames per second. A single missing frame may not be realized either and need not imply a reduction of quality. Apteker et al. [3] and Steinmetz and Engler [4] have studied user assessment of quality.

User Preferences: Does the user care about certain errors? A user may see the difference between a 60 frames per second video and 40 frames per second, but may be perfectly happy with 20 frames per second, particularly when some



cost is involved with a higher frame rate. Another user may feel bothered by a low frame rate, but may not care about low image resolution. Hence, perceived quality depends on subjective user preferences.

Task Dependency: What effect do errors have on the usability of a presentation for particular tasks? Reading figures and tables in an educational video requires a high spatial resolution but not a high frame rate, whereas it is the other way around for action videos such as sports.

B. Presentation-Level QoS

Presentation-level QoS is independent of the user as well as of the underlying system and devices. It describes the output, not the required resources. A set of QoS dimensions can be defined on the presentation level [6, 7]. For videos, there is for instance frame rate, spatial resolution, and color depth. Delay jitter is critical for continuous media streams. If a presentation consists of several streams, for instance a video and an audio stream, the accuracy of synchronization is another important characteristic. In contrast to the actual user perception, QoS can be described easily on this level. A technically literate user can use these parameters to specify his preferences and requirements [1].

C. Resource-Level QoS

Resource-level QoS parameters define quality in terms resources for particular devices, for instance CPU cycles; network throughput, bit error rate, and delay jitter; disk bandwidth; resolution and colors of the display. In communicating with devices (for measuring or requesting quality) and admission testers a multimedia system needs to use these dimensions [8, 9, 10, 11, 12]. They are generally pointless to end users.

D. Mapping between QoS Levels

Any sophisticated QoS management (with or without performance guarantees) needs a way to assess the quality a resource-level QoS configuration provides. This assessment is necessary to compare and choose among several possible configurations. The mapping between resource and presentation-level parameters is often not a problem. For instance, for uncompressed video, the required bandwidth can be calculated by frame rate times pixel per image (resolution) times color depth. There may be, however, several ways of providing a particular presentation quality. The use of compression, for instance, trades CPU capacity for network bandwidth [6]. Assessing a presentation-level QoS specification, that is, mapping the presentation level QoS parameters to a quality value, needs to somehow take user perception, user preferences, and task requirements into account.

Staeli proposes a general QoS model consisting of *content*, *view*, and *quality* specifications [6, 7]. Content describes the output related to logical space and time as edited by the author of a presentation. The user-defined view is the ideal mapping of content to output devices and real time. Quality denotes the difference between actual and ideal presentation. An *error model* is used to calculate this difference. The error in each dimension can be interpreted as a combination of several error components like shift rate, jitter, and synchronization error etc. Weights for the components of all errors can be integrated in the interpretation, allowing calibration of the quality measurement for particular users and tasks.

A multimedia application implementing such an error model does not depend on particular knowledge of user perception, because new information can be easily included in the error interpretation as it becomes known. This QoS model provides an overall quality measure that includes the relative importance of various QoS dimensions and allows the system to assess trade-offs between several configurations.

III. RESOURCE MANAGEMENT

Conventional resource allocation strategies mainly aim at providing fairness among tasks. Continuous media, however, introduce real-time constraints. In an overload situation, a fair policy will allocate an equal amount of resources to all tasks, but this share may be too little for each task and leads to poor quality. Because of these problems, new approaches to resource management have been proposed. They can be categorized by the level of performance guarantees they provide.

A. Performance Guarantee

Best Effort

Today's common operating systems and the Internet provide shared resources with little support for resource management techniques suitable for multimedia requirements. Hence, most applications have taken a best-effort approach in providing QoS. They adapt the quality of their presentations to the amount of resources that is currently available [13, 14, 15, 16]. Commonly, best-effort applications have a simple, one-dimensional quality model or no notion of quality at all.

Hard Guarantees

In a guaranteed system, the user gives the required QoS. The system guarantees this quality by reserving resources such as CPU cycles, memory, network bandwidth, and storage system access and dedicates them to the application. If there are insufficient resources for meeting the specification, the system rejects the request. This approach requires the ability of network and operating system to reserve resources. A variety of protocols have been proposed for implementing this reservation. Some are listed in Section 2.2.3.



Soft Guarantees

Guarantees are not necessarily static. Allowing some degree of quality variation becomes useful if the resource needs of a presentation are not constant. This situation occurs when the QoS requirements change during a presentation or the effort for providing a presentation changes. In some cases of QoS changes, depending on how well the content of a stream can be compressed the bandwidth and CPU cycles requirements change. In this case, hard guarantees require dedication of sufficient resources to meet the maximum need of an application. Often such a strong commitment is not needed and wasteful [17]. Other approaches use statistical guarantees or specify a range of acceptable quality parameters [18, 19]. This flexibility enables the system to reduce quality temporarily in transient overload situations resulting in more tasks being admitted and in better resource utilization.

B. Resource Management Tasks

Several categories for resource management tasks and QoS processing steps have been identified in the literature [1, 20]. Many combinations of these steps may be useful. The effects range from statistical performance improvements to hard performance guarantees.

QoS Specification and Mapping

Approaches that monitor quality or that predetermine resource consumption, need a way of obtaining a QoS specification. The interface may simply consist of several sliders or it may let the user choose from several quality examples. The requirements may be input on any of the QoS levels described in Section 2.1. If user- or presentation-level are used, the specification needs to be mapped to the resource-level for communication with devices.

QoS Negotiation

If guarantees are to be provided, a feasible configuration must be negotiated between all system components involved. Each of them must check if it has the resources required to provide the requested quality.

Resource Reservation and Admission Testing

If all system layers have sufficient resources to provide the requested QoS, these resources can be reserved for the application. If there are insufficient resources, the new task could cause an overload situation. Hence, it will not be admitted, protecting the already admitted tasks from unacceptable performance deterioration. The user (or application) may choose a lower quality and try again. Because performance guarantees require dedicated resources, admission testing is necessary to provide them. However, it is also possible to use weaker admission tests that allow overload situations to a limited extent or for a limited time, resulting in softer guarantees. Moreover, policing may be necessary to protect the admitted tasks from each other as explained below.

Scheduling

To provide guarantees, resources not only need to be reserved but also their use needs to be scheduled according to real-time constraints. Tasks handling continuous media streams need to be processed periodically by the CPU. Conventionally, operating systems do not support such requirements and implement some fair scheduling policy such as round robin. These scheduling policies can cause multimedia tasks to miss their deadlines causing low quality; even if there actually are sufficient CPU cycles to process all streams. This problem can be addressed by using rate-monotonic or earliest-deadline-first scheduling [10]. Split-level scheduling is an advanced technique using lightweight processes [21].

Policing

If it is possible for an application to consume more resources than requested at the admission test, it can try to obtain resources reserved for other tasks. Policing mechanisms can be used to detect these violations and react, for instance, by restricting the task's resource consumption or terminating it.

Monitoring

Most systems that allow QoS to vary implement some kind of monitoring of the currently provided quality. Based on this information, the user can be notified of quality degradation or system components can be adjusted to changed resource availability, for instance. Enhanced adaptation techniques usually require monitoring.

C. Existing Solutions

Protocols

A variety of network protocols for real-time transmission and bandwidth reservation have been developed. Guarantees can be provided only if all routers on a path through the network support a protocol, however. The session reservation protocol SRP in the DASH system [8, 9] processes resources in two phases. In the first phase, all nodes in the pipeline reserve the resources involved. If a maximum end-to-end delay is exceeded, the reservation fails. On the other hand, if the delay is smaller than a target delay, the reservations are relaxed in a second phase. The Internet Stream Protocol ST2+ [22] was proposed as an adjunct to IP and can be accessed by higher-level end-to-end real-time protocols. It supports data streams to single or multiple destinations. Prior to genuine transmission, real-time channels are established. During this phase, a resource manager at each host or router reserves CPU, main memory, and network bandwidth according to QoS specifications. The Tenet Group developed several protocols. The Continuous Media Transport Protocol (CMTP) [11] is



based on the Real-Time Channel Administration Protocol (RCAP), a connection administration service, and the Real-Time Internetwork Protocol (RTIP), a network service providing real-time guarantees in ATM and FDDI networks. As for QoS, CMTP supports maximum limits of stream delay, delay jitter, granularity of data loss, and probability of a data loss. The resource Reservations Protocol (RSVP) [12] is designed to support not only point-to-point, but also multipoint-to-multipoint applications. Hence, it allows sharing of data streams. The application that reserves bandwidth is the only one to control the packets but may allow others to read the stream, too. This approach saves bandwidth in multicast scenarios. The receiver initiates the reservation. For defining timing, multicast, and payload information, the Real-Time Transport Protocol (RTP) [23] is becoming more and more common.

Other Issues

In addition to the CPU and network, management of other resources such as storage access and physical memory also needs to have real-time capabilities [10, 24, 25]. Efficient IPC mechanisms, such as memory-mapped streams [21], are critical to process large amounts of multimedia data in time. Several integrated systems for providing end-to-end QoS guarantees such as the Meta-Scheduler [8], the Heidelberg Transport System (HeiTS) [26, 27], and the Lancaster Quality of Service Architecture (QoS-A) [17] have been developed. A QoS-Broker has been proposed as a way of encapsulating resource management [28].

IV. ENHANCED ADAPTIVE MECHANISMS

The Enhanced adaptive mechanism for resource management is a greedy best-effort approach. Multimedia Application tools try to get as close to optimal quality as possible consuming as many resources as they can get to achieve this goal. This approach does not require any support by the operating system nor by the network and there is no need to deal with QoS specifications.

This approach aggressively consumes resources in order to provide a good presentation [7]. In order to save resources for other applications or to be polite to others, the user may want to limit his resource consumption and be satisfied with a suboptimal, but sufficient quality. A specification of a maximum quality can achieve this goal. Still, no support by the underlying system is needed to do so. However, the application must have a QoS specification and monitor the quality currently provided.

Simple best-effort applications adapt their resource consumption in an uncontrolled way. In a video pipeline, for instance, buffers overflow, network packets are lost, or a software decoder cannot handle the data volume, since the CPU is too busy. For information that is dropped at a late stage of the pipeline, all resources used for processing it in earlier stages are wasted. An overloaded pipeline performs worse than a fully loaded one. Feedback mechanisms can be used to adapt the rate at which the remote server sends frames to the rate at which they are eventually displayed [13, 14, 16]. Ideally, no more frames are fed into the video pipeline than can be processed with the available resources. Controlling this dynamic adaptation is not a simple task. If adaptation is too slow, variations in resource availability are not well compensated for. If adaptation is too fast, the system is susceptible to measuring errors and may overreact or oscillate. Many multimedia applications use some kind of feedback. Often, however, their structure is specific to the application and their behavior is not well known. Cen et al. [13] have taken a more systematic approach by using a feedback toolkit for adaptation in their multimedia application. The toolkit contains several filters and control algorithms [29, 30]. Using well known components allows the composition of predictable feedback mechanisms based on control theory.

V. CONTROLLING SEVERAL QoS DIMENSIONS

For multimedia application, the simplest parameter to vary in order to change resource consumption is the frame rate. Having control over several QoS dimensions provides a more exact way to specify quality (for guaranteed performance approaches) or to adapt quality (for best-effort approaches) [25, 31, 31]. Such dimensions include spatial resolution, color depth, or the lossiness of compression algorithms. If a multimedia application is capable of varying several QoS parameters, it needs a way of assessing the trade-offs between adjusting in one QoS dimension or another. An error model [6, 7] as described in Section 2.1 can solve this problem by providing an overall quality measure based on user perception and preferences. A quality value can be calculated for QoS configurations, creating a partial order among them. The system can assess which of two possible configurations is better. Another approach has been proposed by Thimm and Klas [33]. They describe possible ways of adaptation as ∂ -sets and propose a heuristic scheme for selecting the most appropriate one, based on resource availability and user preferences.

VI. CONCLUSION

To control and manage the resources in distributed multimedia applications, a QoS notion is necessary. QoS can be specified in terms of device parameters (resource level), characteristics of the output (presentation level), or an assessment modeling the value of a presentation to the user (user level). Throughout this paper, most quality notions are expressed at the presentation level. An error model can be used to interpret a presentation level description and map it to user level QoS. Providing good support for the resource management to multimedia applications requires extensions of operating systems and networks by new resource management features. With support of all stages of a continuous media pipeline performance guarantee is possible. However, with Enhanced adaptive mechanism for resource management approach needs to be taken. Feedback mechanisms allow doing this adaptation in systematic and controlled way improving resource utilization. Applications that control several QoS dimensions are capable of adapting their resource consumption in different ways approximating the needs of the user more exactly. Supporting presentations that are composed of several streams from possibly different servers requires coordination of timing and resource management



among the streams. Such complex presentations introduce the need of additional synchronization and prefetching mechanisms.

VII. REFERENCES

- [1] A. Vogel, B. Keherve, G. v. Bochmann, and J. Gescei. Distributed multimedia and QOS: A survey. *IEEE Multimedia*, Vol. 2, No. 2, summer 1995, pp.10-19.
- [2] D. Hutchison, G. Coulson, A. Campbell, and G. S. Blair. Quality of Service management in distributed systems. In *Network and Distributed Systems*, 1994, Addison-Wesley, Editor M. Sloman, pp. 213-302.
- [3] R. T. Apteker, J. A. Fisher, V. S. Kisimov, and H. Neishlos. Video Acceptability and Frame Rate. *IEEE Multimedia*, Vol. 2, No. 3, Fall 1995, pp.32-40.
- [4] R. Steinmetz and C. Engler. Human perception of media synchronization. Tech. Report 43.9310, IBM European Networking Center, Heidelberg, 1993.
- [5] R. Staehli. Quality of Service Specification for Resource Management in Multimedia Systems. Ph.D. thesis, Oregon Graduate Institute of Science & Technology, January 1996.
- [6] R. Staehli, J. Walpole, and D. Maier. Quality of Service Specification for Multimedia Presentations. *Multimedia Systems*, November 1995, Vol. 3, No. 5/6, pp. 251-263.
- [7] D. P. Anderson. Meta-scheduling for distributed continuous media. Technical report CB/CSD 90/599, University of California, Berkeley, October 1990.
- [8] D. P. Anderson, S.-Y. Tzou, R. Wahbe, R. Govindan, and M. Andrews. Support for continuous media in the DASH system. In *Proc. of the 10th International Conference on Distributed Computing Systems (ICDCS10)*, Paris, May 1990, pp. 54-61.
- [9] R. Steinmetz. Analyzing the Multimedia Operating System. *IEEE Multimedia*, Vol. 2, No. 1, spring 1995, pp. 68-84.
- [10] B. Wolfinger and M. Moran. A continuous media data transport service and protocol for real-time communication in high speed networks. In *Proc. of the 2nd International Workshop on Network and Operating System Support for Digital Audio and Video*, Heidelberg, Germany, November 1991. LNCS 614, Springer Verlag, Berlin, pp. 171-182.
- [11] L. Zhang S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A new resource ReSerVation Protocol. *IEEE Network*, Vol. 7, No. 5, Sept. 1993, pp. 8-18.
- [12] S. Cen, C. Pu, R. Staehli, and J. Walpole. A distributed real-time MPEG videoaudio player. In *Proc. of the 5th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Durham, New Hampshire, April 1995. LNCS 1018, Springer Verlag, Berlin, pp. 142-153.
- [13] Z. Chen, S.-M. Tan, R. H. Campbell, Y. Li. Real time video and audio in the world Wide web. In *World Wide Web Journal*, Vol. 1, January 1996.
- [14] K. Jeffay, D. L. Stone, T. Talley, and F. D. Smith. Adaptive, best-effort delivery of digital audio and video across packet-switched networks. In *Proc. of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, California, November 1992. LNCS 712, Springer Verlag, Berlin, pp. 3-12. [25] K. Nahrstedt and J. M. Smith. The QOS Broker. *IEEE Multimedia*, Vol. 2, No. 1, spring 1995, pp. 53-67.
- [15] R. G. Herrtwich. The role of performance, scheduling, and resource reservation in multimedia systems. In *Operating Systems of the 90s and Beyond*, A. Karshmer and J. Nehmer, editors, LNCS 563, Springer Verlag, Berlin, 1991, pp. 279-284.
- [16] S. T.-C. Chou and H. Tokuda. System support for dynamic QoS control of continuous media communication. In *Proc. of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, California, November 1992. LNCS 712, Springer Verlag, Berlin, pp. 363-368.
- [17] D. Ferrari, J. Ramaekers, and G. Ventre. Client-network interactions in quality of service communication environments. In *Proc. of the 4th IFIP Conference on High Performance Networking*, Laige, Belgium, Dec. 1992, pp. E1-1 - E1-14.
- [18] D. Hutchison, G. Coulson, A. Campbell, and G. S. Blair. Quality of Service management In distributed systems. In *Network and Distributed Systems*, 1994, Addison-Wesley, editor M. Sloman, pp. 213-302.
- [19] R. Govindan and D. P. Anderson. Scheduling and IPC mechanisms for continuous media. In *Proc. of the Thirteenth ACM Symposium on Operating Systems Principles*, Pacific Grove, CA, October 1991, pp. 68-80.
- [20] D. Maier, J. Walpole, and R. Staehli. Storage System Architectures for Continuous Media Data. In *proceedings of Foundations of Data Organization and Algorithms*, FODO '93, LNCS 730, 1993, Springer-Verlag, pp. 1-18.



- [21] R. Staehli and J. Walpole. Constrained Latency Storage Access. *Computer*, March 1993, pp. 44-53.
- [22] L. Delgrossi and L. Berger, Editors. Internet Stream Protocol version 2 (ST2), protocol specification - version ST2+. Internet request for comments: 1819, RFC-1819, August 1995.
- [23] L. Delgrossi, C. Halstrick, D. Hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt. Media scaling for audiovisual communication with the Heidelberg Transport System. In *Proc. ACM Multimedia 93*, ACM Press, New York, 1993, pp.99-104.
- [24] D. Hehmann, R. G. Herrtwich, W. Schulz, Th. Schuett, and R. Steinmetz. Implementing HeITS: Architecture and implementation strategy of the Heidelberg High-Speed Transport System. In *Proc. of the 2nd International Workshop on Network and Operating System Support for Digital Audio and Video*, Heidelberg, Germany, November 1991. LNCS 614, Springer Verlag, Berlin, pp. 33-44.
- [25] K. Nahrstedt and J. M. Smith. The QOS Broker. *IEEE Multimedia*, Vol. 2, No. 1, spring 1995, pp. 53-67.
- [26] L. A. Rowe and B. C. Smith. A continuous media player. In *Proc. of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, California, November 1992. LNCS 712, Springer Verlag, Berlin, pp. 376-386.
- [27] H. Massalin and C. Pu. Fine-grain adaptive scheduling using feedback. *Computing Systems*, Vol. 3, No. 1, winter 1990, pp. 139-173.
- [28] C. Pu and R. M. Fuhrer. Feedback-based scheduling: a toolbox approach. *Proc. Of the Fourth Workshop on Workstation Operating Systems*, Napa Valley, California, October 1993, pp.124-128.
- [29] T. Talley and K. Jeffay. Two-dimensional scaling techniques for adaptive, rate based transmission control of live audio and video streams. In *Proc. of the Second ACM International Conference on Multimedia*, San Francisco, Oct. 1994, ACM Press, pp. 247-254.
- [30] H. Tokuda, Y. Tobe, S. T.-C. Chou and J. M. F. Moura. Continuous media communication with dynamic QOS control using ARTS with an FDDI network. *ACM SIGCOMM 92*, Baltimore. *Computer Communications Review*, Vol. 22, No. 4, October 92, pp. 88-98.
- [31] H. Thimm and W. Klas. \square -Sets for optimized reactive adaptive playout management in distributed multimedia database systems. In *Proc. of the 12th International Conference on Data Engineering*, New Orleans, Louisiana, Feb. 1996, pp. 584-592.
- [32] M. E. Hodges, R. M. Sasnett, and M. S. Ackerman. A construction set for multimedia applications. *IEEE Software*, Vol. 6, No. 1, January 1989, pp 37-43.
- [33] Th.D.C.Little and A. Ghafoor. Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Communication*, Vol 8. No. 3, April 1990, pp. 413-427.