Overview of Research at the Distributed Multimedia Computing Laboratory

Harrick M. Vin

TR-96-29
November 1996

Overview of Research at the Distributed Multimedia Computing Laboratory

Harrick M. Vin

Distributed Multimedia Computing Laboratory
Department of Computer Sciences, University of Texas at Austin
Taylor Hall 2.124, Austin, Texas 78712-1188, USA
E-mail: vin@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885
URL: http://www.cs.utexas.edu/users/vin, http://www.cs.utexas.edu/users/dmcl

August 1996

1 Vision and Objectives

Recent advances in computing and communication technologies have made it not only technically feasible, but also economically viable to design multimedia information management systems in a variety of application domains (ranging from education to science and engineering, and from medicine to commerce and entertainment). Wide spread deployment of such applications, however, hinges upon the availability of *integrated multimedia computing environments* that support efficient mechanisms for storing, transmitting, and presenting multiple forms of information (e.g., video, audio, images, 3D graphics, text, and numeric data). The main objective of our research is to create such an environment by designing and implementing: (1) a multimedia file system for storage and retrieval of multi-resolution multimedia objects, (2) network algorithms and protocols for transmission of multimedia objects over integrated services networks, and (3) operating system mechanisms such as processor scheduling as well as transport and higher layer protocols for efficient processing and streaming of data at the end-station. This report summarizes some of our recent research in each of these three areas.

2 Multimedia File System

The main objective of this component of our research is to design and implement an integrated multimedia file system that can efficiently manage the storage and retrieval of heterogeneous multimedia objects. Since continuous media types, such as digital audio and video, differ fundamentally from textual and numeric data in their characteristics (e.g., real-time requirements and data transfer rate), much of our research to date has focused on the development of a continuous media storage server. We are currently in the process of integrating the results of our research with conventional file system design techniques. In this section, we will first summarize the results of our prior research, and then present an overview of the work in progress.

2.1 Efficient, Reliable Storage of Continuous Media

2.1.1 Efficient Placement on Disk Arrays

Problem

Digitization of audio yields a sequence of samples and that of video yields a sequence of frames. A continuously recorded sequence of audio samples or video frames is referred to as a *media stream.* Due to the immense sizes and data transfer rates of media streams, most multimedia servers are founded on disk arrays. To effectively utilize a disk array, multimedia servers interleave the storage of each media stream among disks in the array. The unit of interleaving, referred to as a *media block* or a *striping unit*, denotes the maximum amount of logically contiguous data stored on a

single disk. To maximize the throughput of an array, a multimedia server needs to determine an optimal block size and the number of disks over which blocks of a stream are interleaved (i.e., the degree of striping).

In conventional file systems, a block size is considered optimal if it minimizes the *average* response time while maximizing the throughput of the disk array. In contrast, to meet the continuous playback requirement of video and audio streams, an optimal block size for multimedia servers must minimize the *variance* in response time while maximizing the throughput. Whereas small blocks result in a uniform load distribution among disks in the array (thereby decreasing the variance in response times), they increase the overhead of disk seeks and rotational latencies (thereby decreasing throughput). Large blocks, on the other hand, increase the array throughput at the expense of increased load imbalance and variance in response times. Consequently, a multimedia server must select a block size that balances these tradeoffs.

Our Approach

We have formulated the problem of determining optimal block sizes for two different placement policies. As per these policies, each media block may contain either a fixed number of media units (e.g., video frames or audio samples) or a fixed number of storage units (e.g., bytes). If a media stream is compressed using a variable bit rate (VBR) compression algorithm, then the storage space requirement may vary from one media unit to another. Hence, a server that constructs a media block from a fixed number of media units will be required to store variable size blocks on the array (referred to as *variable-size* block placement). On the other hand, if media streams are stored in terms of fixed size blocks (referred to as *fixed-size* block placement), then each block will contain a variable number of media units. Thus, depending on the placement policy, accessing a fixed number of media units of a stream will require the server to retrieve either a fixed number of variable-size blocks, or a variable number of fixed-size blocks from disk. Due to the sequentiality of audio and video playback, variable-size block placement yields predictable access patterns for the disk array, and thereby simplifies disk resource management. This, however, is achieved at the expense of increased complexity of storage space management. The fixed-size block placement policy, on the other hand, simplifies storage space management at the expense of more complex resource management algorithms. Hence, the variable-size block placement policy is more suitable for predominantly read-only environments (e.g., video-on-demand (VOD) servers). On the other hand, environments which involve frequent creation, deletion, and modification of media objects (e.g., multimedia file systems) favor a fixed-size block placement policy.

For each of these placement policies, we have developed a model for determining the optimal media block size. This is the first model that precisely characterizes the effect of continuous media access on the performance of disk arrays. Through extensive simulations and experimentation, we have validated the model as well as characterized the dependence of optimal media block size on the number of disks in the array, the maximum number of clients accessing the server, and their data rate requirements. As for the degree of striping, we have shown that, in relatively small disk arrays, striping multimedia objects across all disks in the system (referred to as wide-striping) yields a balanced load and maximizes throughput. However, wide-striping is an inadequate load balancing mechanism for large disk arrays. Consequently, to maximize the throughput, the server is required to stripe multimedia objects across subsets of disks in the system, and replicate their storage so as to achieve load balancing. We are currently investigating the factors that govern degree of striping and replication policies.

Representative Publication

H.M. Vin, S.S. Rao, and P. Goyal, "Optimizing the Placement of Multimedia Objects on Disk Arrays", In *Proceedings of the Second IEEE International Conference on Multimedia Computing and Systems, Washington, D.C.*, Pages 158-165, May 1995

2.1.2 Failure Recovery

Problem

Fault-tolerance is an issue that arises with increase in number of disks in a multimedia server. Most of the conventional methods for disk failure recovery are based on the Redundant Array of Inexpensive Disks (RAID) architecture, which achieves fault tolerance either by *mirroring* or *parity encoding*. Mirroring (also referred to as RAID level 1) achieves fault tolerance by duplicating data on separate disks, and, consequently, incurs a 100% storage space overhead. Parity

encoding (variations referred to as RAID levels 3, 4, and 5) reduces this overhead by employing error correcting codes for failure recover. In such architectures, if one of the disks fails, the missing data is reconstructed by executing an exclusive-or operation on the data and the parity blocks stored on the surviving disks. In the event of a disk failure, such an approach significantly increases the load on the surviving disks. To prevent saturation (and thereby avoid discontinuities in the playback of media streams), a server employing the RAID architecture for failure recovery will need to operate at low utilization levels during the fault free state. To minimize such wasted bandwidth, a multimedia server must ensure that the recovery process has minimal impact on the system performance.

Our Approach

We have developed two techniques that utilize the inherent characteristics of video streams to minimize the overhead of on-the-fly disk failure recovery.

- In the first method, we exploit the sequential nature of video stream accesses to reduce the overhead of on-line recovery in a RAID array. Specifically, by requiring that parity blocks be computed over a sequence of blocks belonging to the same video stream, the method ensures that data blocks retrieved by a server for failure recovery would be requested by the client in the near future. By buffering such blocks and then servicing the requests for their access from the buffer, this method minimizes the overhead of on-line failure recovery. In fact, we have shown that, when this scheme is used for RAID level 5 array, the overhead of on-line recovery reduces from 100% to $1/(G-1)$, where G is the parity group size. Similarly, when this scheme is applied to a declustered parity array, the overhead reduces from $(G-1)/(C-1)$ to $1/(C-1)$, where C is the cluster size.

- Our second approach exploits the semantics of the video data itself for efficient failure recovery. Specifically, since human perception is tolerant to minor distortions in video playback, rather than perfectly recovering images stored on the failed disk, this method partitions each image in a video stream into several sub-images such that the information contained in each sub-image can be approximated from the other sub-images by using the *spatial* and *temporal* redundancies inherent in video streams.

  To illustrate the concept, consider the scenario where an image is partitioned into sub-images such that none of the immediate neighbors of a pixel in the image belong to the same sub-image. If these sub-images are stored on different disks, then even in the presence of a single disk failure, all the neighbors of the lost pixels will be available. In this case, the high degree of correlation between neighboring pixels will make it possible to reconstruct a reasonable approximation of the original image. Although conceptually elegant, such pre-compression image partitioning techniques significantly reduce the correlation between the pixels assigned to the same sub-image, and hence adversely affect image compression efficiency. Hence, a key challenge is to achieve good failure recovery without affecting compression efficiency.

  We have achieved this objective by developing a novel *post-compression* partitioning technique in which an image is partitioned into several sub-images *after* compression. We have illustrated our method by developing *Loss-Resilient JPEG (LRJ)* and *Loss-Resilient MPEG (LRM)* compression algorithms. We have also developed a *Distributed Recovery in an Array of Disks (DRAD)* architecture that combines these loss-resilient compression algorithms with techniques for efficient placement of video streams on disk arrays to ensure that on-the-fly recovery does not impose any additional load on the disk array. Together, they enhance the scalability of multimedia servers by: (1) integrating the recovery process with the decompression of video streams, and thereby distributing the reconstruction process across the clients, and (2) supporting graceful degradation in the quality of recovered images with increase in the number of disk failures. Observe that, since such a scheme decouples the process of on-line, imperfect recovery from off-line, perfect rebuild of the failed disk, it represents a fundamental departure from conventional failure recovery techniques. Moreover, this recovery technique is equally effective in masking packet losses resulting from network congestion, and hence, achieves end-to-end failure recovery.

Representative Publication

H. M. Vin, P. J. Shenoy, and S. Rao, "Efficient Failure Recovery in Multi-Disk Multimedia Servers", In *Proceedings of the 25th International Fault Tolerant Computing Symposium (FTCS), Pasadena, CA*, Pages 12-21, June 1995

2.2 Efficient Retrieval of Media Streams

2.2.1 Admission Control Algorithms

Problem

Due to the sequential and periodic nature of media playback, a multimedia server services multiple streams by proceeding in *rounds*. During each round, the server retrieves a fixed number of media units (e.g., video frames) for each stream. To ensure continuous playback, the number of media units accessed for each stream must be sufficient to sustain its playback rate, and the service time (i.e., the total time spent in retrieving media units during a round) should not exceed the duration of a round. Hence, before admitting a new client, a multimedia server must employ admission control algorithms to decide whether a new client can be admitted without violating the continuous playback requirements of the clients already being serviced.

Our Approach

To achieve this objective, an admission control algorithm needs to estimate the resource requirements (i.e., bit rate, disk access times, etc.) of the new client as well as all the clients already being serviced. We have developed three categories of admission control algorithms that differ significantly in their estimation methods:

- *Deterministic* admission control algorithms make worst-case estimates of the bit rate and disk access times, and are used when clients cannot tolerate any losses.
- *Statistical* admission control algorithms use pre-computed probability distributions of the bit rate and disk access times to guarantee that deadlines will be met with a certain probability. Such algorithms achieve much higher utilization than deterministic algorithms, and are used when clients can tolerate infrequent losses.
- *Measurement-based* admission control algorithms use measurements of bit rate and disk utilization from recent past as an indicator of future resource requirements. They achieve the highest disk utilization at the expense of providing the weakest guarantees.

These admission control algorithms span an entire spectrum and achieve varying server utilization while providing different levels of guarantees. In environments with varying user requests, a multimedia server must support some or all of the above admission control algorithms to service its heterogeneous clientele.

Representative Publications

1. H. M. Vin, A. Goyal, and P. Goyal, "Algorithms for Designing Multimedia Storage Servers", In *Computer Communications*, Vol. 18, N0. 3, Pages 192-203, March 1995
2. H. M. Vin, P. Goyal, A. Goyal, and A. Goyal, "A Statistical Admission Control Algorithm for Multimedia Servers", In *Proceedings of the ACM Multimedia'94, San Francisco, CA*, Pages 33-40, October 1994

2.2.2 Multi-resolution Multimedia Objects

Problem

Most of the existing work on multimedia servers assumes homogeneous computing and communications infrastructure (i.e., each client is assumed to be equally capable of accessing and consuming multimedia information). However, in reality, the computing infrastructure at client sites may range from hand-held devices to powerful workstations, and the networks connecting storage servers to client sites may range from the huge installed base of ethernets, token rings, telephone lines, etc. to high speed (e.g., FDDI, ATM, etc.) and wireless networks. In such heterogeneous environments, retrieving a multimedia object from a file system without considering the capabilities of the client site and the interconnection network may be highly wasteful of resources, and, in general, may not be feasible. Development of techniques for efficiently servicing clients over such heterogeneous environments is a nascent research area.

Our Approach

To efficiently service user requests originating from heterogeneous computing and communication infrastructures (as well as request with different quality of service requirements), we are designing a file system that provides systematic support for storing and retrieving multimedia information at various levels of detail or resolution (in chroma, spatial, and temporal dimensions). Since most multi-resolution encoding techniques generate multiple streams of data, a subset of which may be combined to obtain the desired level of resolution, a server must organize their storage on disk such that: (1) the overhead in accessing the highest resolution level is minimized, and (2) any subset of streams can be accessed without accessing any additional information.

We have carried out some preliminary experiments on efficient placement of multi-resolution MPEG-2 video streams on disk array. We have demonstrated that, by integrating the placement policy with multi-resolution encoding, a server can completely eliminate the overhead of interactive scan operations (e.g., fast forward and rewind) by dynamically reducing the resolution of the streams being delivered to clients. We have generalized our method to a variety of multi-resolution compression algorithms, and demonstrated that, by tailoring the retrieval of media streams from disk to the desired resolution level, such a file system will transfer only as much data as needed, and thereby improve the utilization of both the server and network resources.

Representative Publication

P.J. Shenoy and H.M. Vin, "Efficient Support for Scan Operations in Video Servers", In *Proceedings of the ACM Multimedia'95, San Francisco, CA*, Pages 131-140, November 1995

2.3 Work In Progress

We are currently extending our work on continuous media server design along two directions:

- *Design and implementation of an integrated multimedia file system:* The main objective of this research is to investigate various issues involved in the integrated management of heterogeneous objects. These include the development of: (1) data type independent as well as data type dependent file system data structures (e.g., i-nodes), (2) techniques for enabling the co-existence of data type specific striping and failure recovery methods (e.g., small stripe unit and perfect recovery requirement of textual data vs. large stripe unit and approximate recovery requirements of video), (3) resource reservation and admission control algorithms, and (4) techniques for minimizing response times for non real-time requests while meeting the performance requirements of real-time requests. We are also evaluating the suitability of client-pull (employed in conventional file systems) and server-push (utilized in most video servers) architectures for such integrated multimedia file systems. Promising algorithms developed for addressing all of these requirements are being implemented in our prototype multimedia file system.

- *Scalability:* To meet the scalability requirements (in terms of storage space, bandwidth, and reliability) of large multimedia servers (e.g., web servers), we are investigating various issues involved in designing *clustered* multimedia server. A clustered architecture consists of a set of nodes (consisting of a processing node as well as a hierarchy of storage devices) interconnected by a network (e.g., fibre channel, SSA, ATM, IBM SP switch, etc.). The specific issues being investigated include: (1) models for characterizing the effect of network bandwidth and latency on the stripe unit size and degree of striping, (2) techniques for tolerating disk, node, and network link failures, (3) admission control algorithms, as well as (4) methods for replication, distribution, and caching to ensure reliable and responsive access to information.

3 Network Support for Multimedia

The main objective of this component of our research is to develop a framework that will enable integrated services networks to provide Quality of Service (QoS) guarantees to a variety of applications.

3.1 Packet Scheduling Algorithms

Integrated services networks are required to support a variety of applications (e.g., audio and video conferencing, multimedia information retrieval, ftp, telnet, WWW, etc.) with a wide range of Quality of Service (QoS) requirements. Whereas continuous media applications such as audio and video conferencing require the network to provide QoS guarantees with respect to bandwidth, packet delay, and loss; applications such as telnet and WWW require low packet delay and loss. Throughput intensive applications like ftp, on the other hand, require network resources to be allocated such that the throughput is maximized. A network can meet these requirements by appropriately *scheduling* its resources. In this section, we describe our work on developing a framework for providing end-to-end guarantees as well as a scheduling algorithm for integrated services packet networks.

3.1.1 Guaranteed Rate Scheduling Algorithms

Problem

Recently, several research groups have investigated network architectures and algorithms that employ open-loop control strategies for providing QoS guarantees to applications. In these architectures, a source is required to negotiate a contract with the network by specifying its traffic characteristics. The network, in turn, employs packet scheduling and admission control algorithms, and guarantees that as long as the source conforms to its traffic specification, the QoS guarantees provided to the channel will be met. Although a large number of scheduling algorithms have been proposed in literature, most of them have been analyzed for only a limited set of traffic specifications, and in homogeneous networking environments. However, to be useful in integrated services networks, the analysis must address:

- *Heterogeneity in source traffic characteristics*: The traffic characteristics of multimedia sources differ significantly. For example, while many audio applications require constant bit rate service, resource requirement of applications transmitting Variable Bit Rate (VBR) compressed video sequences varies significantly over time.
- *Heterogeneity in the network characteristics*: Current networks are, and future networks will remain, heterogeneous along several dimensions. For example, in a large network consisting of several autonomous domains, switches may employ different scheduling algorithms (e.g., work conserving, non-work conserving, ones that separate delay and rate allocation, and the ones that only allocate rate). Furthermore, due to the variation in the size of data transmission unit in internetwork environments (e.g., an internetwork consisting of ATM, FDDI, ethernet, and token ring), packet fragmentation and/or reassembly may also occur in the network.

Thus, in such heterogeneous environments, the techniques for providing QoS guarantees should be flexible enough to accommodate: (1) a wide range of traffic specifications, (2) variable rate allocations for a source, (3) a variety of scheduling algorithms at the switches, and (4) internetworking environments (in which fragmentation and reassembly may occur).

Our Approach

We have developed a comprehensive framework for providing QoS guarantees by: (1) defining a class of generalized Guaranteed Rate (GR) scheduling algorithms, and (2) developing a general method for providing QoS guarantees in a heterogeneous networking environment.

The class of GR scheduling algorithms guarantee a deadline (referred to as *delay guarantee*) to a packet based on its expected arrival time. The delay guarantee of these algorithms is independent of a traffic specification and the behavior of other flows at the server. This enables a single server employing a scheduling algorithm in GR to isolate flows as well as provide service guarantees to flows conforming to any specification. We have demonstrated that the class of GR scheduling algorithms is broad, and includes work conserving and non-work conserving scheduling algorithms as well as algorithms that allocate only rate and those that separate rate and delay allocation (e.g. Virtual Clock, Packet-by-Packet Generalized Processor Sharing (PGPS), Self Clocked Fair Queuing (SCFQ), Delay Earliest Due Date (EDD), and Rate Controlled Static Priority Queuing). We have defined work conserving generalized Virtual Clock, PGPS, and SCFQ scheduling algorithms that can allocate variable rate to the packets of a flow, and have shown that they also belong to GR. To prove that a scheduling algorithm belong to GR, we have developed a proof methodology in which we first show that a preemptive equivalent of the algorithm belongs to GR, and then utilize a relationship

between preemptive and non-preemptive scheduling algorithms to show that the non-preemptive algorithm belongs to GR. This methodology not only simplifies the proofs but also leads to the definition of several scheduling algorithms in GR that are suitable for servers at which packet fragmentation may occur. The algorithms that we have defined for such servers reduce computational complexity as well as delay incurred by packets. Finally, we have demonstrated that if a rate control element is employed in conjunction with any scheduling algorithm in GR, the resulting non-work conserving algorithm also belongs to GR (this leads to the definition of several scheduling algorithms). Furthermore, such rate control elements do not change the delay guarantee of the scheduling algorithm.

The delay guarantee of the GR class enables a single server to provide service guarantees to flows conforming to any specification. To enable a sequence of servers to provide similar service guarantees, we have developed a method for deriving the delay guarantee of a network of servers each of which employs a scheduling algorithm in the GR class. This method enables the derivation of delay guarantee for a network of servers even when: (1) different rates are allocated to packets of a flow at different servers along the path and the bottleneck server for each packet may be different, and (2) packet fragmentation and/or reassembly may occur in the network. We utilize the delay guarantee of a network of servers to obtain an upper bound on end-to-end delay. We have demonstrated that our method of deriving delay guarantee for a network of servers reduces the problem of determining end-to-end delay bound to a single server problem. We have illustrated the end-to-end delay bound computation for flows conforming to Leaky Bucket, Exponentially Bounded Burstiness and Flow Specification; and have demonstrated that our method for determining these bounds is not only simple, but also leads to tighter results (e.g., it improves upon the delay bound presented in for flows conforming to Leaky Bucket in PGPS networks).

Representative Publication

1. P. Goyal, S. S. Lam, and H. M. Vin, "Determining End-to-End Delay Bounds In Heterogeneous Networks", In *ACM/Springer-Verlag Multimedia Systems Journal (to appear)*, 1996 (An earlier version of this paper appeared in the Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95), April 1995).

2. P. Goyal and H. M. Vin, "Generalized Guaranteed Rate Scheduling Algorithms: A Framework", Technical Report TR-95-30, Department of Computer Sciences, The University of Texas at Austin, July 1995 (submitted for publication).

3.1.2 Scheduling Support for Integrated Services Packet Switching Networks

Problem

To determine the characteristics of a suitable scheduling algorithm for integrated packet switching networks, consider the requirements of some of the principal applications envisioned for integrated services networks:

- *Audio applications*: To maintain adequate interactivity for such applications, scheduling algorithms must provide low average and maximum delay.
- *Video applications*: Variable bit rate (VBR) video sources, which are expected to impose significant requirements on network resources, have unpredictable as well as highly variable bit rate requirement at multiple time-scales. These features impose two key requirements on network resource management:
  - Due to the difficulty in predicting the bit rate requirement of VBR video sources, video channels may utilize more than the reserved bandwidth. As long as the additional bandwidth used is not at the expense of other channels (i.e., the channel utilizes idle bandwidth), it should not be penalized in the future by reducing its bandwidth allocation.
  - Due to multiple time-scale variation in the bit rate requirement of video sources, to achieve efficient utilization of resources, a network will have to overbook available bandwidth. Since such overbooking may yield persistent congestion, a network should provide some QoS guarantees even in the presence of congestion.

Unfair scheduling algorithms, such as Virtual Clock, Delay EDD, etc., penalize channels for the use of idle bandwidth and do not provide any QoS guarantee in the presence of congestion. Fair scheduling algorithms, on the other hand, guarantee that, regardless of prior usage or congestion, bandwidth would be allocated fairly. Hence, fair scheduling algorithms are desirable for video applications.

- *Data applications*: To support low-throughput, interactive data applications (e.g., telnet), scheduling algorithms must provide low average delay. On the other hand, to support throughput-intensive, flow-controlled applications in heterogeneous networks, scheduling algorithms must allocate bandwidth fairly. Due to the coexistence of VBR video sources and data sources in integrated services networks, the bandwidth available to data applications may vary significantly over time. Consequently, the fairness property of the scheduling algorithm must hold regardless of variation in server capacity.

In summary, a suitable scheduling algorithm for integrated services networks should: (1) achieve low average as well as maximum delay for low throughput applications (e.g., interactive audio, telnet, etc.); (2) provide fairness for VBR video; and (3) provide fairness, regardless of variation in server capacity, for throughput-intensive, flow-controlled data applications. Moreover, to facilitate its implementation in high-speed networks, it should be computationally efficient. None of the scheduling algorithms proposed in the literature meet all of these requirements.

Our Approach

We have developed Start-time Fair Queuing (SFQ) algorithm that is computationally efficient and allocates bandwidth fairly regardless of variation in a server rate. We have shown that its fairness measure is at most a factor of two away from the lower bound and is at least as good as the fairness measure of all known fair scheduling algorithms. We have analyzed the throughput, single server delay, and end-to-end delay guarantee of SFQ. To accommodate links whose capacity fluctuates over time (for example, flow-controlled and broadcast medium links), this analysis is carried out for servers which can be modeled as either Fluctuation Constrained (FC) or Exponentially Bounded Fluctuation (EBF) servers. To the best of our knowledge, this is the first analysis of a fair or a real-time scheduling algorithm for such servers.

We have shown that SFQ is suitable for integrated services networks since it: (1) achieves low average as well as maximum delay for low-throughput applications (e.g., interactive audio, telnet, etc.); (2) provides fairness which is desirable for VBR video; (3) provides fairness, regardless of variation in server capacity, for throughput-intensive, flow-controlled data applications; (4) enables hierarchical link sharing which is desirable for managing heterogeneity; and (5) is computationally efficient.

We have also demonstrated that: (1) SFQ is better suited than WFQ for integrated services networks since it efficiently achieves fairness over variable rate servers and provides significantly smaller average and maximum packet delay to low-throughput applications; (2) SFQ is strictly better than SCFQ since maximum packet delay in SFQ is considerably smaller than in SCFQ and both have same the fairness measure and implementation complexity; (3) SFQ is strictly better than FQS since it has lower complexity and achieves fairness over variable rate servers without increasing the maximum packet delay; and (4) SFQ provides considerably better fairness properties and smaller maximum delay than DRR.

Representative Publication

P. Goyal, H. M. Vin, and H. Cheng, "Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks", In *Proceedings of SIGCOMM'96, San Francisco*, Pages 157-168, August 1996

3.2 Network Protocols for Video Transport

Digital video is expected to constitute significant portion of the aggregate traffic on an integrated services network. Moreover, since the bandwidth requirement of uncompressed digital video is of the order of few hundred megabits/second, for the foreseeable future, digital video will be transmitted in a compressed form. Consequently, in this component of our research, we focus on network protocols that are specifically designed for transporting of compressed video.

The fundamental difficulty in developing such network protocols is that the bit rate requirement of digital video compressed using Variable Bit Rate (VBR) encoding algorithms (e.g., MPEG) has multiple time-scale variations. Whereas the short term variations (i.e., frame-to-frame variations) occur due to the use of inter-frame compression techniques, the longer term variations (i.e., at the time-scale of seconds) occur due to the inherent differences in the scene complexity. Due to this multiple time-scale variation, to achieve efficient utilization, a network would be required to overbook its resources.

If a network overbooks its resources, then packet losses will occur. Although uncompressed video streams are highly resilient to packet losses (due to the inherent spatial and temporal redundancy), compressed video streams are not. Since real-time nature of video playback precludes any retransmission of lost packets, minimizing the impact of packet losses on video quality will require the sender and the receiver to employ additional error recovery methods. For instance, a source can send forward error correction information which can be utilized by a decoder at the receiver to compensate for packet losses. On the other hand, if the network provides multiple priority levels and if the source employs a multi-layer encoder, then the reconstructed image quality at the receiver can be improved by transmitting layers of the encoded video stream at different priority levels (e.g., transmitting the essential and the enhancement layers at high and low priorities, respectively). Although conceptually elegant, the above mentioned techniques are not without limitations. Whereas the additional data traffic yielded by the redundant forward error correction information increases the overall load and hence may worsen the loss rate; layered encoders, in general, are more complex and require higher bandwidth as compared to a standard single-layer encoder. Consequently, packet losses for compressed video should be avoided.

Thus, the key problem is to design network protocols for video that achieve efficiency by statistically multiplexing the network resources, but avoid degradation in video quality caused due to packet losses. A network protocol that achieves this objective has to manage both the short term as well as the long term variations. The relative importance of the two time-scales, however, depends on the application requirements. For example, since stored video applications do not have very stringent end-to-end delay requirements, a network protocol for stored video can smooth out the short-term variations, and only manage the longer term. On the other hand, since interactive video applications have low delay requirements, the corresponding network protocol cannot employ smoothing to remove the fast time-scale variations (and hence, has to predominantly manage the fast time-scale variations). Based on this observation, we have developed different network protocols for stored and interactive video.

3.2.1 Stored Video Transport

Problem

Even when the short-term variations of VBR video are smoothed out by shaping the traffic at the source, the average data transfer requirement continues to differ by a factor of three or more across scenes. Consequently, to achieve high utilization, a network will be required to overbook its resources. In such a scenario, to minimize the degradation in video quality resulting from congestion and packet losses, protocols that are optimized for stored video communication should exploit predictability of the bit rate requirement to completely avoid any packet loss in the network. Such protocols can shift the error recovery functionality to the servers, which know the semantics of the data and can ensure minimum degradation in the perceived video quality during congestion. The design of such protocols has not received much attention.

Our Approach

To address this limitation, we have developed a network service specifically designed for multimedia servers. Since end-to-end delay requirements for stored video are not very stringent, our protocol shapes the traffic such that the rate changes are regular, infrequent, and predictable. A histogram-based characterization of this traffic, when coupled with an admission control algorithm, enables the network to provide heterogeneous statistical QoS guarantees to clients. To meet these guarantees, we have proposed an *overload control* algorithm and protocol that exploits the regularity and predictability in traffic to provide heterogeneous QoS while completely eliminating packet losses in the network. Specifically, it: (1) detects congestion in the network prior to its occurrence, (2) allocates bandwidth to competing sources based on their QoS guarantees, and (3) transmits feedback packets to the sources indicating the rate allocations. By ensuring that the feedback is received by the sources sufficiently prior to overload occurrence, the protocol enables

the sources to employ application specific procedures to adjust their transmission rates so as to conform to the rate allocations, thereby eliminating any packet losses in the network.

We have formulated the problem of allocating bandwidth which meets the QoS guarantees of sources as a linear program, and developed an optimal, linear-time algorithm which exploits the special structure of the problem. Moreover, since the protocol employs traffic shaping, the playback initiation latency and buffer requirement at the client may increase. We have derived bounds for increase in the latency, and have experimentally demonstrated that the increase in initiation latency as well as buffer requirement are insignificant.

The key contribution of our protocol lies in combining open-loop and feedback-based control to: (1) provide heterogeneous QoS to clients in networking environments consisting of switches that may not have any scheduling support; and (2) migrate the functionality of discarding packets, in the event of congestion, to the sources which understand the semantics of the data. The protocol is efficient, makes very few assumptions about the underlying network, is realizable on current switching hardware (supporting FCFS scheduling), and is completely integrated with the architecture of a multimedia server.

Representative Publication

P. Goyal and H. M. Vin, "Network Algorithms and Protocol for Multimedia Servers", In *Proceedings of INFOCOM'96, San Francisco, CA*, Pages 1371-1379, March 1996

3.2.2 Feedback-based Control for Live Video Transport

Problem

A network protocol for interactive video has to control both short term as well as long term bit rate variations. Long term variations lead to sustained overload on a network. Hence, a network must employ an admission control algorithm that limits statistical multiplexing. Short term variations, on the other hand, lead to queue build up, and consequently to significant packet losses at switches. A network can avoid such packet losses by either: (1) employing source traffic shaping algorithm and removing short term rate fluctuations, or (2) absorbing transient overloads by increasing buffer space at the switches. Since source traffic shaping algorithms increase the end-to-end delay of the frames significantly (by approximately 200ms), they are not suitable for interactive video applications. On the other hand, predicting the buffer space required to absorb transient overloads for video sources is difficult. Furthermore, increasing expensive fast buffer at the switches in high speed networks may not be economically viable. Thus, network protocols for efficient management of short-term burstiness of VBR video is an important research problem.

Our Approach

We have developed a network layer protocol for video transport that is based on a per link, hop-by-hop credit-based flow control algorithm. The basic idea in credit based flow control is to reserve buffer for a flow at all the switches along the path from the source to the destination, and then limit the number of packets an upstream node (i.e., *sender*) can transmit to a downstream node (i.e., *receiver*) such that buffer at the downstream node does not overflow. The buffer reserved for a flow depends on the desired bandwidth of a flow and can be changed dynamically (i.e., *adaptive buffer allocation*).

Our network protocol for video transport: (1) minimizes buffer requirement by adaptively allocating buffer at the switches while ensuring that packet losses do not occur in the network, and (2) minimizes end-to-end delay and jitter of frames. To achieve the former objective, we have utilized receiver-oriented adaptive credit-based flow control algorithm, and have derived the number of buffers necessary and sufficient to ensure reliable transmission. To minimize the end-to-end delay and jitter for VBR encoded video streams, we have: (1) studied bandwidth estimation techniques which exploit the structure of the video traffic, and (2) defined a new fairness criteria with respect to the delays experienced by the video frames during congestion and developed a fair buffer allocation algorithm. The adaptive credit-based flow control algorithm when coupled with the bandwidth estimation and buffer allocation algorithm ensures that bandwidth for each video channel is allocated on-demand. This achieves extremely high network utilization, but does not guarantee the end-to-end delay and jitter. To mask the effects of delay jitter on playback continuity, we have developed a technique for adapting playback point at client sites.

We have experimentally demonstrated that due to the inherent nature of the protocol, the network, rather than the source, shapes the traffic, which in turn yields smaller end-to-end delay for video frames as compared to source traffic shaping algorithms. We have shown that the previously known receiver-oriented buffer adaptation algorithm requires 100% more buffers and incurs 20 times higher delay as compared to our algorithm. Finally, we have demonstrated that our video transmission protocol out-performs numerous other schemes with respect to end-to-end delay, buffer space requirement, and packet loss.

Our protocol does not control the long-term variations, that is, it does not limit statistical multiplexing through an admission control algorithm. However, by eliminating packet losses in the network, it facilitates the use of heuristic admission control algorithms based on measured traffic statistics. Thus, our protocol not only effectively controls the short term variations, but also simplifies the network control for long term variations.

Representative Publication

P. Goyal, H. M. Vin, C. Shen, and P.J. Shenoy, "A Reliable, Adaptive Protocol for Video Transport", In *Proceedings of INFOCOM'96, San Francisco, CA*, Pages 1080-1090, March 1996

3.3 Work In Progress

The following are some of the research issues currently under investigation:

- We are developing a hierarchical link sharing mechanism for managing the heterogeneity in integrated services networks. It can be used by a network to support services that provide heterogeneous QoS, as well as multiple protocol families that support different traffic types and/or congestion control mechanisms. For example, a network can support hard and soft real-time, as well as best effort services by partitioning the link bandwidth between them as per the expected requirements of each of the service. To support high and low reliability soft real-time services, the bandwidth of soft real-time service may be further partitioned. Similarly, the bandwidth of the best effort services may be further partitioned between throughput intensive and interactive services.

  A key advantage of hierarchical link sharing is that it provides isolation between different services while enabling similar services to share resources. Hence, incompatible congestion control algorithms can co-exist while compatible algorithms reap the advantages of sharing. Hierarchical link sharing also facilitates use of different resource allocation methods for different services. This is desirable as hard real-time services may use a scheduling algorithm that performs well when there is no overbooking; soft real-time services may prefer to use a scheduling algorithm that provides QoS guarantees and/or minimizes deadline violations in presence of overbooking; and best effort services may use a fair scheduler for throughput intensive flow-controlled data applications.

  To support such hierarchical link sharing, we are developing a hierarchical SFQ scheduler, which we plan to implement in an IP router.

- We are addressing several fundamental issues that arise in designing guaranteed services. For example:

  1. What are the desired characteristics of scheduling algorithms? Should they be fair or unfair algorithms suffice? Should they separate rate and delay allocations, or allocate only rate?
  2. Should guaranteed services require traffic specification?
  3. Should the traffic be smoothed at the source or should a burst be allowed to enter the network?
  4. Can guaranteed services achieve reasonable utilization, or will it be necessary to provide ad hoc predictive services?

4 End-station Support for Multimedia

The main objective of this component of our research is to deliver the QoS provided by the file system and the network components of the end-to-end system to the applications. To achieve this objective, we are developing processor scheduling mechanisms as well as transport and higher layer protocols necessary for efficient processing and streaming of data at the end-station.

4.1 Processor Scheduling

Problem

To determine suitable CPU scheduling algorithms, let us consider the requirements imposed by various application classes that may co-exist in an integrated multimedia system:

- *Hard real-time applications*: These applications require an operating system to provide deterministic delay guarantee to various tasks. Conventional schedulers like Earliest Deadline First (EDF) and Rate Monotonic Algorithm (RMA) used in hard-real time systems are suitable for such applications.

- *Soft real-time applications*: These applications require an operating system to statistically guarantee QoS parameters such as maximum delay and throughput. Since a large number of such applications are expected to involve video, to determine a suitable scheduling algorithm for this class of applications, let us consider the processing requirements for variable bit rate (VBR) video.

  Due to inherent variations in scene complexity as well as the use of intra- and inter-frame compression techniques, processing bandwidth required for compression/decompression of frames of VBR video varies highly at multiple time-scales. Furthermore, these variations are unpredictable. These features lead to the following requirements for a processor scheduling algorithm for VBR video applications:

  - Due to multiple time-scale variation in the computation requirement of video applications, to efficiently utilize CPU, an operating system will be required to over-book CPU bandwidth. Since such over-booking may lead to CPU overload (i.e., cumulative requirement may exceed the processing capacity), a scheduling algorithm should provide some QoS guarantees even in the presence of overload.
  - Due to the difficulty in predicting the computation requirements of VBR video applications, a scheduling algorithm should not assume precise knowledge of computation requirements of tasks.

  EDF and RMA algorithms do not provide any QoS guarantee when CPU bandwidth may be over-booked. Furthermore, they assume precise knowledge of the computation requirements of tasks. For example, they require the release time, the period, and the computation requirement of each task (thread) to be known a priori. Consequently, although these algorithms are suitable for hard real-time applications, they are not suitable for soft real-time, multimedia applications. Hence, a new scheduling algorithm that addresses these limitations is required.

- *Best-effort applications*: Many conventional applications do not need performance guarantees, but require the CPU to be allocated such that average response time is low while the throughput achieved is high. This is achieved in current systems by time-sharing scheduling algorithms.

From these requirements, we conclude that different scheduling algorithms are suitable for different application classes in a multimedia system. Hence, an operating system framework that enables different schedulers to be employed for different applications is required. In addition to facilitating co-existence, such a framework should provide protection between the various classes of applications. For example, it should ensure that the over-booking of CPU for soft-real time applications does not violate the guarantees of hard real-time applications. Similarly, misbehavior of soft/hard real-time applications, either intentional or due to a programming error, should not lead to starvation of best-effort applications.

Our Approach

To meet these objectives, we have developed a framework for *hierarchical partitioning* of CPU bandwidth. In our framework, CPU bandwidth is partitioned among various application classes, and each application class, in turn, partitions its allocation (potentially using a different scheduling algorithm) among its sub-classes or applications. This hierarchical partitioning is specified by a tree. Each thread in the system belongs to exactly one leaf node, and hence each node in the tree represents either an application class or an aggregation of application classes. Whereas threads are scheduled by leaf node dependent schedulers (determined by the requirements of the application class), intermediate nodes are scheduled by an algorithm that achieves hierarchical partitioning. Specifically, intermediate nodes must be scheduled by an algorithm that: (1) achieves fair distribution of processor bandwidth among competing nodes, (2)

does not require a priori knowledge of computational requirements of threads, (3) provides throughput guarantees, and (4) is computationally efficient. The Start-time Fair Queuing (SFQ) algorithm that we have described for network scheduling meets all of these requirements. Moreover, it is suitable for soft real-time video applications. We have implemented our hierarchical scheduling framework in Solaris 2.4. Our results have demonstrated that the framework: (1) enables co-existence of heterogeneous schedulers, (2) protects application classes from each other, and (3) does not impose higher overhead than conventional time-sharing schedulers.

Representative Publication

P. Goyal, X. Guo, and H.M. Vin, "A Hierarchical CPU Scheduler for Multimedia Operating Systems", In *Proceedings of the Second Symposium on Operating System Design and Implementation (OSDI'96), Seattle*, Pages 107-121, October 1996

4.2 Presentation Processing Support

Problem

Consider the design of a toolkit that provides presentation processing support for multimedia applications. Presentation processing in multimedia applications involve accessing, decoding, and processing different types of media objects (e.g. audio, video, images) which may be stored in a variety of compressed formats (e.g. MPEG, JPEG, H.261). Given that the computing and communication infrastructures are changing rapidly, such a toolkit should provide mechanisms for supporting clients over a wide range of heterogeneous environments. Similarly, the toolkit should support mechanisms that will enable applications to take advantage of emerging compression standards, rather than being obsoleted by them. Finally, since the resource intensive nature of digital video processing makes the quality of multimedia presentations particularly sensitive to variations in resource availability, the toolkit should support mechanisms to adapt its resource requirements to the run-time environment, and thereby minimize the impact of resource scarcity on the perceptible quality of presentations.

Our Approach

We have achieved these objectives by developing: (1) a library of reusable compression and image processing modules, and (2) a framework for dynamically composing these modules to create presentation processing engines (PPE). For instance, since many compression algorithms employ the same set of transformations (e.g. Huffman coding, discrete cosine transform, etc.), our framework facilitates rapid, on-demand instantiation of a compression algorithm by composing reusable, modular implementations of these transformations. The modules are configured at run-time based on the compression format of media object, resource availability, as well as the application's QoS requirements (expressed in terms of frame rate, resolution, and SNR). Moreover, the configuration can be altered to accommodate dynamic changes in resource availability and the QoS requirements of applications (e.g., when CPU becomes overloaded, degrade the quality of presentation by using computationally less expensive dithering and inverse discrete cosine transform functions). Such dynamic adaptation is made possible by providing a run-time selectable set of implementation modules, which differ only in their resource requirement and presentation quality, for each transformation. Finally, through the selective use of static (compile-time) and dynamic binding of modules, the toolkit allows for efficient implementation of PPEs while maintaining a modular, configurable architecture.

The significant functional overlap found among presentation processing algorithms and the desire to eliminate unnecessary functionality motivated a compositional, rather than a monolithic, approach to PPE implementation. Similarly, the need to improve efficiency via integrated layer processing motivated the selective use of static binding. The three main benefits of our approach are: (1) the PPE implementation is configured at run time, and hence, can adapt to heterogeneous environments and changing resource availability; (2) substantial code reuse can be achieved by instantiating new algorithms using the same module implementations; and (3) the toolkit is inherently extensible, since new modules can be easily added to the library.

Representative Publication

E.J. Posnak, H.M. Vin, and G. Lavender, "Presentation Processing Support for Adaptive Multimedia Applications", In *Proceedings of the Multimedia Computing and Networking 1996 (MMCN96), San Jose, CA*, Pages 234-245, January 1996

4.3 Work In Progress

We are currently extending our work on operating system support for multimedia applications along two dimensions:

- Our framework for hierarchical partitioning of CPU bandwidth facilitates the development of a QoS manager that allocates resources as per the QoS requirements of applications. To illustrate, if an application requests hard/soft real-time service, then the QoS manager can use a deterministic/statistical admission control algorithm which utilizes the capacity allocated to hard/soft real-time classes to determine if the request can be satisfied, and if so, assign it to the appropriate partition. On the other hand, if an application requests best-effort service, then QoS manager would not deny the request but assign it to an appropriate partition depending on some other resource sharing policies. A QoS manager can also dynamically change the hierarchical partitioning to reflect the relative importance of various applications. For example, initially soft real-time applications may be allocated very small fraction of the CPU, but when many video decoders requesting soft real-time services are started (possibly as a part of a video conference), the allocation of soft real-time class may be increased significantly. The development of such a QoS manager as well as the mechanisms that an application can use to express its quality of service requirements to the manager form the nucleus of our current research on processor scheduling.

- We are extending our approach to building configurable presentation processing engines (PPEs) to the entire transport and higher level protocol stack. Specifically, we are: (1) implementing transport and higher level services as a library of composable modules in the user space, and (2) providing mechanisms to compose the modules into protocols that can meet the application requirements. This framework will support on-demand composition, and even recomposition, in the event of dynamic changes in resource availability and/or application requirements. Such a framework, when coupled with the mechanisms for specifying the quality of service requirements (in terms of bandwidth, delay, jitter, synchronization, etc.) of applications, will yield a complete end-station QoS architecture.

5 Distributed Multimedia Computing Laboratory

The research work summarized in this report was carried out at the *Distributed Multimedia Computing Laboratory (DMCL)* at UT Austin. This laboratory was established in the Fall 1994, and is largely the result of resources provided by grants from the National Science Foundation, NASA, IBM, Intel, Mitsubishi Electric Research Laboratories (MERL), Sun Microsystems Inc., Applied Research Laboratory (ARL) at UT Austin, Electrospace Systems Inc., and the University of Texas at Austin.

The laboratory currently consists of a 4-node IBM POWERParallel SP2 system, 2 dual-processor ultraSPARC stations, 7 Sun SPARCstations, 10 Pentium- and PentiumPro-based desktop workstations. Each node of the IBM SP2 is interconnected by a high-performance, multistage, packet-switched network (namely, the SP2 switch), and consists of the latest POWER2 technology RISC Systems/6000 processors, an array of disks, an adapter for the high-speed processor interconnect switch, and an ATM adapter. The Sun SPARCStations and the pentium-based workstations are equipped with digital audio and video compression/decompression engines, ATM network adapters, as well as other peripheral devices (e.g., video camera, speakers, etc.). The Pentium-based workstations are also equipped with Intel's ProShare video conferencing system. In addition to the disk arrays associated with each node of the IBM SP2, the laboratory also contains 2 external disk arrays - one connected to a Sun SPARCStation 20, and the other to a dual-processor Pentium-based server. All of these workstations are interconnected by a 10 Mbit ethernet, as well as a 155 Mbits/s FORE systems ATM switch. The current hardware configuration of the laboratory is shown in Figure 1.

Each node on the IBM SP2 is running AIX. The Sun SPARCstations are running Solaris, and the Pentium-based workstations are running Windows NT as well as Solaris. We have access to Solaris source code, and consequently, all of our code development is under Solaris.

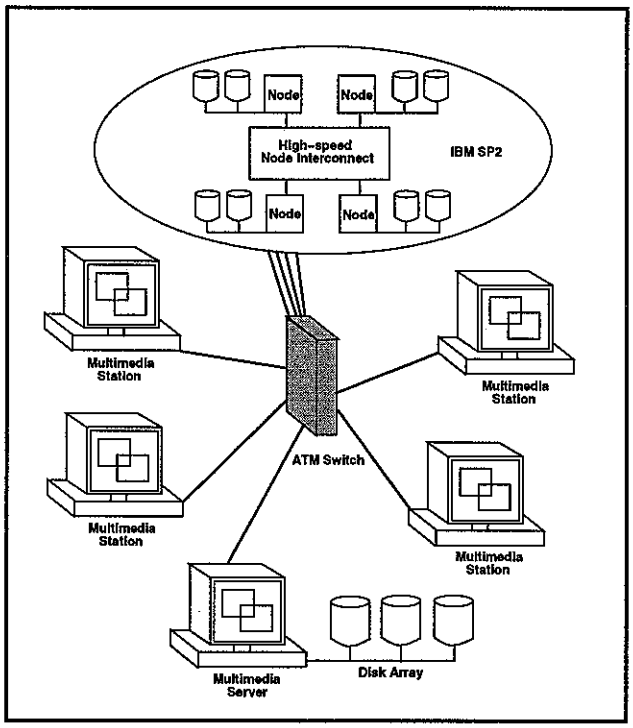

Figure 1: Hardware configuration of the Distributed Multimedia Computing Laboratory

6 Concluding Remarks

The main objective of our research is to create an integrated multimedia computing environment by designing and implementing: (1) a multimedia file system for storage and retrieval of multi-resolution multimedia objects, (2) network algorithms and protocols for transmission of multimedia objects over integrated services networks, and (3) operating system mechanisms (including processor scheduling as well as transport and higher layer protocols) for efficient processing and streaming of data at the end-station. This is an experimental systems project that involves both theoretical and practical work, and will significantly advance the state of art in designing large-scale distributed multimedia systems.