A Multimedia Client to the IBM LAN Server

Mark Baugher, Steven French, Alan Stephens, and Isabel Van Horn

IBM Personal Software Products
Internal Zip 9131
11400 Burnet Rd.
Austin, Texas 78758

Internet: mbaugher@vnet.ibm.com, stephens@vnet.ibm.com

ABSTRACT

File system *redirection* is an established technique in personal computer local area networks for providing *location transparent* file access. This paper argues that redirection can be used for multimedia file access, and considers the extensions needed in the client, server, transport and network subsystems to enable multimedia redirection. Results are presented on an IBM LAN Server prototype which provides quality of service guarantees to multimedia clients on the Token Ring.

INTRODUCTION

Today, we have data networks in workplaces and campuses throughout the world, and we have file systems which manage data files. Multimedia networks and multimedia files are distinguished by having a wider range of quality of service (QoS) requirements. Quality of service in data networks usually means a choice between reliable or unreliable service depending upon whether the user of the service wants the system to automatically retransmit lost or corrupted data. But QoS in multimedia systems may include delay, delay variation, throughput, and error guarantees [2, 7]. Thus multimedia networks and file systems encompass all of the functions of data networks and data file systems.

Multimedia applications lie along a continuum with applications that require no guarantees for error, delay, or throughput at one end, and applications that require guaranteed high throughput and low delay service at the other end. Between these two extremes are *multimedia playback* applications such as media on demand (e.g., music or movies on demand), *electronic book* applications (i.e., digital documents that integrate sound, voice, video), and *hypermedia* applications, which are electronic books that permit document navigation through links.

A distinction can be made between multimedia applications that play back recorded multimedia and live multimedia applications in which two or more people communicate, digitally, through a computer network in a teleconferencing session [3]. Playback applications access stored media from disk, and are less sensitive to delay and delay variation when large playout buffers are used. Unlike live multimedia, many playback applications are "standalone," meaning that computer networks are not needed for the presentation. Electronic books and hypermedia applications, for example, may come packaged on one or more compact discs. Here the storage and distribution mechanisms are one and the same.

Standalone multimedia applications are limited by what can fit upon a workstation floppy or hard disk drive. But multimedia playback from remote file servers will greatly expand the variety and availability of digital multimedia. A server *plays back* a multimedia file when it delivers file elements, such as sound, voice or video, to the client at the rate at which these elements are delivered to multimedia devices.

Textual data, though not played back, are often integrated into multimedia presentations. Distributed multimedia systems will usually be required to support the text processing applications which are found on today's personal computer local area networks (PC LAN's). In PC LAN environments, remote file access is commonly supported through file system *redirection* in which access to a remote file is made indistinguishable to local file access by a software component called a *Redirector*. Since practically any multimedia application can and will be run standalone, there are obvious benefits to providing location transparency for multimedia file access, and for using common services for both reserved (multimedia) and unreserved (non-multimedia) access to data files: This permits a gradual introduction of multimedia applications into existing environments; it minimizes the software and hardware costs as well as the risks of running distributed multimedia. This paper describes a file system Redirector which can enable client access to multimedia server and transport services, and considers the resource reservation requirements for a multimedia client implementation.

MULTIMEDIA APPLICATIONS AND OS EXTENSIONS

At the very least, memory resources must be reserved in a standalone computer or client workstation to provide service guarantees to multimedia streams. Dedicated memory buffers between disk storage and multimedia devices are required.

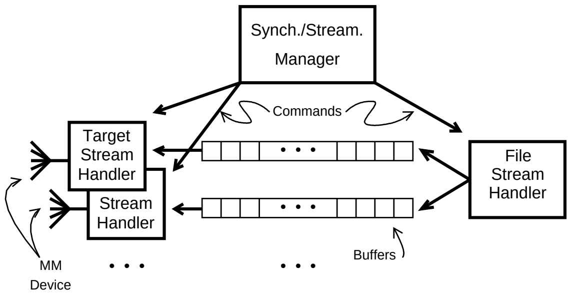

Figure 1. Multimedia synchronization and streaming

Figure 1 shows a simplified multimedia system in which file elements are read by a file stream handler into one or more buffers. The elements are copied from the buffers to the multimedia devices such as CODEC's after they are synchronized and streamed. *Streaming* restores the temporal relationship among file elements before delivery to a multimedia device. *Synchronization* restores the temporal relationship among multiple streams [20]. Following bursty disk accesses, temporal relationships must be restored prior to delivery of stream elements to multimedia devices so the the sound, voice, or video elements are delivered without disruption. Increasingly, extensions are being made to operating systems in order to provide streaming and synchronization services to multimedia applications.

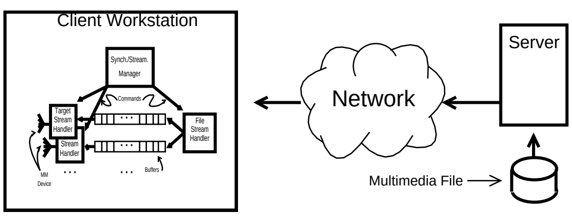

Figure 2. Distributed multimedia playback

Today, there are at least two multimedia systems available for IBM compatible Personal Computers (PC's). Both are extensions to PC operating systems. Microsoft's Multimedia Extensions to Windows, now packaged with Window's 3.1, provides a generic Media Control Interface (MCI) for various digital and analog multimedia devices, and it provides a synchronization and streaming subsystem [12]. IBM's Multimedia Presentation Manager/2 provides similar functions for OS/2 computers [13]. Apple's QuickTime for the MacIntosh is also available on Windows computers. We are not aware of any explicit features in these multimedia extensions for remote playback of multimedia files.

Figure 2 is a simple depiction of distributed playback where the multimedia file is stored on a file server that is remote from the workstation on which the file is played. The streaming subsystem is located in the client workstation of Figure 2. There may be a streaming subsystem in the server as well, but whenever packet networks are used, client streaming is necessary. The client playout buffers shown in Figure 2 perform additional functions beyond those shown in Figure 1 since they remove variations in delay from the network and file server in addition to those from the disk subsystem. In general, the buffer needs to be large enough to sustain a playback during the maximum time it takes to move a block of data between server disk and multimedia device. It is crucial that the network, server and disk subsystems deliver media elements at the rate at which they are displayed otherwise discontinuities occur in which a media element does not correctly follow another media element.

The elements being delivered and presented could range from video frames to audio samples [17, 18]. At the multimedia device, therefore, the presentation unit could be less than a byte or it could exceed 400 bytes, and the period of time it takes to consume a presentation unit could be greater than 1/30th of a second or less than a millisecond. At the file system, however, the granularities are different: Rather than an audio sample or video frame, the file system burst could be a 64 kilobyte READ operation.

Table 1. Video, Audio and Still Frame Flows

	DVI PLV	CD ROM	V/SF
T	150 KBps	175 KBps	37 KBps
B	64 KB	64 KB	64 KB
P	0.43	0.37	1.73

In Table 1, the average throughput, *T*, is the rate of DVI Production Level Video [19], CD-ROM audio, and voice over still-frame. The average burst, *B*, is the READ size, and the period, *P* is computed as in equation (1).

$$P = \frac{B}{T} \quad (1)$$

P is the seconds between successive requests. At minimum, a playout buffer at least as large as the burst, B, is necessary. In practice, the playout buffer size is $(k+1) \times B$. The rationale for setting $k > 0$ will usually be to accommodate transient extremes in delay (i.e., delay variation). If the client/server system cannot supply file elements (the burst, B) to the device at the required rate (the inverse of the period) over k periods, then a discontinuity will occur. A *discontinuity* is a missed deadline [8] in which a multimedia playout buffer experiences underflow; this usually results in disrupted sound, voice or video.

Thus a playout buffer must be maintained in the client workstation and the size of the buffer is a function of delay variation from the server's disk to the multimedia device. This does not mean that the client Redirector necessarily has to maintain playout buffers. The value of Multimedia systems such as the OS/2 Multimedia Presentation Manager/2 to multimedia redirection is that they already provide playout buffers which are adjustable in size. The OS/2 Presentation Manager/2 also provides a configurable READ size, another multimedia client requirement since the burst, B, is the READ size. When these parameters are configurable in the client, we have shown that no change is needed to the Redirector provided that resource reservation is implemented in the server and communications transport (see “Multimedia Client/Server Implementation”).

THE MULTIMEDIA REDIRECTOR

A *Redirector* is a software component installed on the client workstation and server computer that provides at least three functions [10].

1. File abstraction definition, such as the definition of a file as a stream of bytes and also metadata such as size, creation date and extended attributes.
2. File object services such as open, read and write operations.
3. File access protocols including the use of primitive services such as those provided by the communications transport.

IBM's LAN Server uses the Server Message Block (SMB) protocol which has been standardized through X/Open. There are obvious commercial advantages to using a standard protocol such as SMB for multimedia client/server computing since the installed base is quite large. There are also technological advantages to using existing protocols since they will permit rapid introduction of multimedia applications into existing environments. To accomplish this, the multimedia Redirector includes three extensions to the redirection functions listed above.

1. Quality of Service (QoS) metadata is associated with multimedia files.
2. Service guarantees for file operations are provided.
3. A resource reservation protocol exchange is defined between the client and the server and with the service providers that support them, such as the communications transport.

The quality of service (QoS) descriptors characterize the traffic for real-time delivery of file system elements: The QoS descriptors express the deadline constraints. The IBM LAN Server and Redirector use file system Extended Attributes, metadata about the file, for storing and accessing QoS information about a file. The descriptors are shown in Table 2, below.

Table 2. Client/Server Resource Reservation Descriptors

Average Read Throughput	T_{read}
Average Write Throughput	T_{write}
Maximum Read Burst	B_{read}
Maximum Write Burst	B_{write}

Throughput, *T*, is expressed in bytes per second and measured over the time interval or period, *P*. The period is a parameter derived from throughput and burst as in equation (1), above. These definitions are consistent with the QoS parameters contained in the ISDN Frame Relay specification [16]. The end-to-end descriptors shown in Table 2 must be mappable into descriptors needed for the disk, network and transport subsystems. Each of these subsystems must provide service guarantees and usually perform resource reservation.

When the application plays back exactly one file, then the reservation can be based upon the resource requirements of the particular file. This is *file-oriented reservation*. There are file formats, such as the Resource Interchange File Format (RIFF) that may contain sound, voice, video text, or practically any media - in a single file. Many hypermedia applications, however, can be expected to use multiple files, and in this case the reservation should be *session-oriented*: If an individual reservation is made for each file when it is opened, then over-reservation will occur when the files are not accessed concurrently. If the application requires that the resource reservation be dynamically changed, then the application must have some way to communicate the resource reservation request to the Redirector, and the Redirector must have some way of communicating this to the server. Resource reservation protocol is needed for dynamic, session-oriented resource reservation of server and transport resources.

THE MULTIMEDIA SERVER

Resource reservation of disk and CPU is needed in the server to ensure that the server can meet the clients' service deadlines P_i from equation (1). It is clear that the number of client multimedia streams which can be supported by each server and each disk depends upon the scheduling algorithms, not considered in this paper, which are used by the server.

The server obtains a quality of service specification from the file's Extended Attributes, in the case of file-oriented reservation, or from the application *via* the Redirector's Application Programming Interface, in the case of session-oriented reservation. When the server is presented with the service specification as shown in Table 2, it must either accept or refuse the resource reservation request. Acceptance means that the server will guarantee to deliver up to B_{read} elements at a rate at least equal to T_{read} within an amount of time not exceeding some multiple of P_{read} seconds. The same contract holds for write operations. Thus, the service specification of Table 2 contains the information which a server needs to perform scheduling: The period is computable, the computation of moving a sequence of bytes must be known for a particular server, and if the size of the client's playout buffer is also known, then server memory usage can be optimized as well. There is an implicit protocol between the server and the client in that each READ request must complete within $(k + 1) \times P$, for some $k \geq 0$.

The propagation of READ requests from the Redirector to the server may be seen as unnecessary overhead when the streams are periodic and access to the file is sequential. For the most part, today's client/server file systems use byte oriented, client-pull access semantics. *Byte-oriented* means that the file is accessed as a stream of bytes, and most PC file systems do not recognize any additional structure beyond a byte structure. *Client-pull* access for file READ operations means that the client determines how many bytes will be retrieved from a particular byte offset in the file. These semantics do not exploit the periodic and sequential nature of multimedia file streams. Rangan and Vin have shown that optimal use of a server for Media on Demand applications can be achieved through *server-push* access where the server streams multimedia file elements at the necessary rate with only timestamp feedback from the client [18]. When the server controls transmissions based on information about needed rates, it may adjust the rate and size of disk block retrieval to maximize the number of concurrent multimedia playback sessions that it can support. In the disk subsystem, therefore, it is senseless for each multimedia client READ to result in a transfer from disk.

In addition to disk scheduling, there are at least three potential advantages to server-push for multimedia. First, read operations from the redirector to the server are not necessary when traffic is periodic, though some means are needed to start, stop and pause flows. The savings in eliminating read operations are in network bandwidth, server interrupts, and server control over scheduling of disk accesses. Regarding network bandwidth, elimination of read operations for a full-motion video session saves approximately one hundred bytes per second which would need to flow from the client to the server for READ requests. Though this is negligible on packet-based local area networks, it may prove valuable over ISDN links, such as T1 service if acknowledgements are also eliminated. A second advantage is the saving of interrupts caused by read requests from the client to the server, though acknowledgements will still need to be sent to the server following frame transmissions (e.g., on a sliding window basis), and elimination of read interrupts will reduce the total usually less than 8%. Finally, there is potential advantage for streaming and synchronization to be performed only in the server, but streaming and synchronization will always need to occur in the client when running over packet networks which have statistical delays and delay variations.

For today's packet LAN environments, there are no clear advantages to pushing multimedia frames through the network from server to client. Our lab tests show that maximum reservable capacity can be obtained on the Token Ring without changing the read semantics from a pull to a push (see “Multimedia Client/Server Implementation”).

LAN delivery is the current multimedia server bottleneck. A server that could support hundreds of multimedia video sessions could be attached to multiple 100 Mpbs links as a means to deliver sessions to clients. At present, however, each reserved bandwidth LAN attachment generally will not exceed 16 Mbps, in the case of the Priority Token Ring. Alternatives such as Dedicated Ethernet are either very expensive or restricted (see “Multimedia Transport and Network Services,” below). LAN attachment costs contribute substantially to the price of server hardware. Intel 486 and low-end RISC servers generally have about one to six LAN attachments, and will be limited to about 40 DVI Production Level Video sessions running at 1.2 Mbps on average. As discussed below (“Multimedia Client/Server Implementation”), an Intel 486 server equipped with a disk array can support this number of streams relatively cheaply. And the number of multimedia streams that can be provided at reasonable cost (i.e., without an entirely new network and system infrastructure) is constrained by LAN delivery.

MULTIMEDIA TRANSPORT AND NETWORK SERVICES

LAN's are *shared media links*: Multiple stations share a ring or bus. It is therefore hard to guarantee throughput, delay or delay variation for multimedia traffic from a single workstation when it has no control and little information about the traffic coming from other workstations. A multimedia workstation can do very little to compensate for congested links in present-day networks.

Applications with high bandwidth, time sensitive, or periodic streams (or all of the above) need reserved or dedicated bandwidth service. A *reserved bandwidth* network provides quality of service (QoS) guarantees on shared links to some classes of traffic. The Priority Token Ring, Synchronous FDDI and Dedicated Ethernet are examples of reserved bandwidth networks. Unlike a reserved bandwidth network, a *dedicated bandwidth* network is a circuit or physical link, such as an Integrated Services Digital Network (ISDN) B channel that is not necessarily shared among multiple stations. Primary Rate Interface ISDN (PRI-ISDN) can provide low-quality digital video service at approximately 1.5 megabits per second. But there is some question as to the long-term acceptance of this video quality, especially to the home. And PRI-ISDN may never be ubiquitous in North America, Japan or Europe. Broadband ISDN Asynchronous Transfer Mode (B-ISDN ATM) research and development work promises to integrate our data and telephone networks [9, 6]. Such an integration is needed to remove the communications impediments to generalize use of digital multimedia. Until B-ISDN ATM becomes commercially-available, packet-based LAN's augmented with bandwidth reservation are the best multimedia client/server solution.

In order to make use of reserved and dedicated bandwidth networks, a *traffic contract* is needed between the application and the network. As part of the contract, the application provides descriptors of the traffic it will offer and the quality of service it requires; the network may then agree to meet the needed QoS so long as the traffic stays within the declared traffic parameters [5]. Traffic descriptors and quality of service have been standardized in the CCITT ISDN standards. The ISDN Frame Relay standard [16] defines a set of parameters which is suitable for packet networks (see Table 3), and a bandwidth reservation protocol between an endstation requesting network service guarantees and a network provider.

The multimedia Redirector and server described in this paper do not use the frame relay *protocol*, only the service parameters. Since the B-ISDN ATM specification, Q.93b, should be upwardly compatible with Q.933, use of these parameters seems to meet immediate needs as well as future requirements. The following mappings are defined between the parameters of Table 2 and Table 3.

$$max\_transit\_delay \leq (k + 1) \times P$$
$$desired\_outgoing\_throughput = T_{\mathrm{read}}$$
$$outgoing\_committed\_burst = B_{\mathrm{read}}$$
$$incoming\_committed\_burst = B_{\mathrm{write}}$$

In all cases, the desired and acceptable values are set to the same value. The server will initialize the max_transit_delay to be within $(k+1) \times P$, k being determined by the buffers reserved in the client computer. The exact delay value depends upon how much of the end-to-end delay is consumed by the server, its file system and disk subsystem. The multimedia Redirector and server may choose to initialize the excess_burst parameters to be a function of the playout buffer size since the rate of flow will be exceeded while the playout buffer fills.

Table 3. ISDN Q.933 Service parameters
max_transit_delay
max_acceptable_delay
outgoing_maximum_frame_size
incoming_maximum_frame_size
desired_outgoing_thruput
desired_incoming_thruput
min_acceptable_outgoing_thruput
min_acceptable_incoming_thruput
outgoing_committed_burst
incoming_committed_burst
outgoing_excess_burst
incoming_excess_burst
X213_priority

Over 70% of packet LAN's are Ethernet, and it is unfortunate that Ethernet cannot provide any of the service guarantees shown in Table 3 since it uses a collision detection protocol which lacks any means for guaranteeing throughput and delay. Adaptation schemes [8] may provide acceptable multimedia delivery for smaller enterprises that are willing to tolerate some amount of disruption to multimedia playback. The extensions needed to a Redirector to support adaptation are not considered in this paper.

An alternative to shared media Ethernet is *Dedicated Ethernet* where a network hub (switching or bridging hub) provides dedicated bandwidth of up to 10 Mbps to the endstation. This solution sounds better than it really is, however, since for many enterprises there will be more workstations than can be supported by a single hub (an expensive, switched hub may accommodate dozens, not hundreds, of workstations). If the hubs need to be bridged, then the Dedicated Ethernet solution requires that 10 Mbps be reserved for each Dedicated Ethernet workstation *throughout the entire LAN subnetwork*. The simplicity of the Dedicated Ethernet solution comes from the absence of bandwidth reservation protocol since this solution assumes that any station will be running at its maximum rate. And this assumption does not scale well when going through internetworks. In order to solve this problem, a bandwidth reservation protocol and network implementation is necessary, and these services will be provided in a standard way by B-ISDN ATM nearly as soon as they can be provided by Dedicated Ethernet products.

Synchronous FDDI, however, is a reserved bandwidth network which has been standardized, and which may become commercially-available as early as 1993. Synchronous FDDI, an implementation of the FDDI bandwidth reservation standard *SMT 7.1*, defines a negotiation process among stations on a FDDI ring (and only on a single ring) for bounding token holding time and token rotation time. In addition to Synchronous FDDI, which is a finished standard, current proposals for 100 Mbps Ethernet include a priority mechanism which could be used for reserving bandwidth on the bus.

Today, the best choice for a multimedia LAN protocol is arguably the Token Ring. Our lab tests show that the Token Ring can reliably deliver multimedia streams, at least within a single LAN segment (see “Multimedia Client/Server Implementation” below). Token Rings are more expensive than Ethernets, which cannot reserve bandwidth at all, but are cheaper than other alternatives such as FDDI, Dedicated Ethernet, and B-ISDN ATM.

The Multimedia LAN Server uses the priority feature of the Token Ring for reliable delivery of multimedia streams [22]. The *Priority Token Ring*, is a Token Ring in which access to the architected IEEE 802.5 priority mechanism is restricted so as to provide service guarantees. It is necessary to prevent too many sessions from being granted reserved bandwidth than can actually be supported on the LAN or LAN subnetwork. The *Priority Token Ring*, a software extension to existing Token Ring network hardware, has been shown to be effective in reserving a proportion of the ring bandwidth to a server for its multimedia sessions. These sessions, moreover, will be protected against normal data traffic on the ring. When there are more than one multimedia server, the amount of reservable bandwidth is limited by the ring speed, and there is no question that at least 50% of the bandwidth can be reserved for multimedia traffic *coming from any server* on the Token Ring. This figure is derived from the *50% Rule* which governs the functioning of most commercially-used Token Ring adapters that release the token after transmitting each frame [1]. When such an adapter is sending the highest priority frames, it will capture no more than 50% of the tokens if there are active data sources on the ring: Each time the server completes a priority transmission, it releases the token to any data station with a frame to send; the server can make a reservation in the passing frame and receive the next token. Thus the amount of bandwidth which any single server can capture is bounded by ratio (2).

$$\frac{\bar{S}_H}{(\bar{S}_H + \bar{S}_0)} \tag{2}$$

Where $\bar{S}_H$ is the average size non-zero priority frame sent by the single high priority server and $\bar{S}_0$ is the average size of the zero priority frames sent by all other stations on the ring. It is clear that the server will capture only half the bandwidth when its frame sizes are equal to the zero priority frames sent by other stations. This is the limiting case of increasing reservable bandwidth by increasing $\bar{S}_H$ since $\bar{S}_0$, the size of zero priority frames, can be increased as much as $\bar{S}_H$. An alternative approach is to decrease $\bar{S}_0$, and we have implemented a prototype which uses this technique to increase reservable bandwidth on the Priority Token Ring to at least 80%. These techniques are beyond the scope of this paper.

MULTIMEDIA CLIENT/SERVER IMPLEMENTATION

It is not surprising that off-the-shelf multimedia files can be installed on a server computer and successfully accessed by a LAN-attached client workstation. The client Redirector locates and accesses multimedia files as it does any other file. And if the system is underutilized, the rates for even full-motion video will be supported between the client and server.

It is surprising, however, how many video sessions can be supported by commercially-available PC servers. Recent work on high performance servers at the IBM Boca Raton lab has used a 50 MHz 486 PS/2 Model 95 server running the OS/2 LAN Server to support twenty eight DVI Production Level Video sessions. The video was delivered over three 16 Mbps Token Rings, and the workload consists of repeated play of video clips of varying duration, and run at an average rate of 150 kilobytes per second [15].

The current release of the IBM LAN Server, though it can support dozens of video sessions playing different files on the configuration described above, provides no service guarantees. Work on the LAN Server implementation for multimedia has focussed on three goals.

1. To protect multimedia sessions on the network from non-multimedia data traffic and from too much multimedia traffic.
2. To protect multimedia sessions on the server from non-multimedia data traffic and too much multimedia traffic.
3. To increase reservable disk capacity on the server.

Table 4. Zero Priority 16 Mbps Token Ring Server Discontinuity Counts over 5 Minutes

MM Client	Unreserved Data Frame Size			
	2 KB	4 KB	8 KB	16 KB
1	1,169	3,333	5,679	6,586
2	1,397	3,308	5,074	6,727
3	1,373	3,480	5,096	6,742
4	1,052	2,969	4,863	6,458
5	1,288	2,935	4,820	6,555
6	1,278	3,268	4,946	6,546
7	927	3,373	5,094	6,777
8	1,063	3,026	4,914	6,468
9	1,002	2,834	4,823	6,417
10	1,263	2,684	4,704	6,399

Table 4 provides measurements from the playback of ten video clients (DVI PLV at approximately 150 kilobytes per second) from a PC server running the current LAN Server product release which has no special support for multimedia. The cells of the table are counts of discontinuities, per client, when data traffic was introduced at various size frames. A *discontinuity* occurs when a media element *N* is not followed by media element *N+1*, at least not in the correct temporal order [8]. The traffic generator is configured to capture a free token and send a frame of a specified size at every opportunity. Although these clients can run indefinitely with a negligible or zero number of discontinuities, once data traffic is introduced the degradation becomes noticeable. And when the size of the data frames increase, the discontinuities worsen. We have instrumented the application program provided with the ActionMedia II video product to detect *buffer starved* messages from the video device, and to record the amount of time between the receipt of the message and the replenishment of the buffer. The results of Table 4 demonstrate what happens when there is no resource reservation on the LAN for multimedia. The situation is analogous for unreserved file transfers going to and from a disk which supports both multimedia and non-multimedia file access.

A test similar to that of Table 4 was run using eight Ethernet stations. Since Ethernet does not have a fair medium access protocol, a single file transfer introduced discontinuities in only one or two stations. Three or four stations were affected by a second file transfer. As more file transfers occurred from different stations on the bus, all of the workstations experienced discontinuities which completely disrupted the playback. The solution to the problem of client discontinuities is to reserve bandwidth for multimedia sessions and give them priority over unreserved sessions. This can be done on the Token Ring, provided that a system is in-place to specify the service requirements of the multimedia sessions.

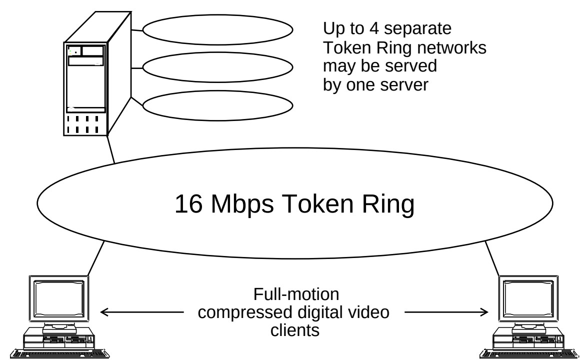

Figure 3. Prototype hardware configuration

The Multimedia LAN Server prototype provides resource reservation in the disk, transport and network subsystems. Figure 3 shows the hardware configuration used by the Multimedia LAN Server prototype. The hardware configuration of Figure 3 is a relatively low cost multimedia client/server platform: A 50 MHz 486 platform supports the OS/2 LAN Server which has software extensions in the file system and transport for disk, server computer and network resource reservation. The server is connected to several Token Rings, possibly with a high-performance multi-priority adapter to provide efficient service for both reserved and unreserved traffic to the server. Our prototype configuration uses the IBM LANStreamer Token Ring adapters which have two transmit channels. The multi-priority service can be implemented in software as well as hardware with some loss in throughput. Our first prototype relies on new network hardware - in the server. The client workstations are PC's, 386 or above, and may use inexpensive Token Ring adapters. All of the multimedia flows are single segment. Bridges and routers may be attached to the server's rings, but reserved flows do not traverse them since multimedia bridges and routers are not yet available.

We tested the prototype using the application program provided with the ActionMedia II digital video product. ActionMedia II provides configurable playout buffers and READ sizes for playback of DVI files. We used Production Level Video (PLV) files for our tests which stream at approximately 150 kilobytes per second. Table 5 shows the result of tests which were run using priority service for the multimedia sessions: The zero discontinuity counts in the cells indicate that playback delivery was satisfactory. As shown by the number of concurrent multimedia clients in Table 5, *our implementation reserves over 75% of the ring bandwidth for multimedia flows, and this reservation is protected against unreserved (zero priority) flows on the ring*. We tested the prototype with twelve ActionMedia II clients, and all clients were protected against data traffic when the unreserved frames did not exceed 8 kilobytes in length. Thus *over 90% of the ring bandwidth can be reserved when the frame size on the ring is below 8 kilobytes*. For most commercial or industrial environments, it is recommended that the ring be configured to keep utilization below 80% [21] though a dedicated, unbridged multimedia Token Ring probably may be run up to 90%. We plan future tests to validate the 90% claim, but it seems clear that *a client-pull implementation can obtain the maximum reservable capacity on a Token Ring - there is no benefit to changing READ semantics from a pull to a push across this LAN type*.

Table 5. High Priority Token Ring 16 Mbps Server Discontinuity Counts over 5 Minutes

MM Client	Unreserved Data Frame Size			
	2 KB	4 KB	8 KB	16 KB
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

Note: Change to IEEE 802.2 protocol used

For our tests, there was no change to the client Redirector apart from instrumenting the application program to detect underflow in the ActionMedia II device. No time protocol was needed since *client pull* file access implicitly synchronizes the flow to the client's clock. Nor did we split the media flow from the control flow though this is, in principle, desirable for multimedia delivery [3]. Unlike the MPEG standard [14], multimedia encoding standards such as DVI, Indeo, and Ultimotion all assume reliable delivery so the file elements require the same service as the control flow.

Reliable delivery of multimedia packets is not a problem on single segment Token Rings: Table 5 shows zero discontinuities after five minutes of play, though the error rate on the shielded twisted pair Token Ring medium is 10^{-9}, and approximately one error occurs every thousand seconds for a flow which is on the order of 1 Mbps. For ten clients, we expect a frame to be lost due to error once per hundred seconds. Although the error would result in retransmissions of up to an entire window (four 4 kilobyte frames were used in the test), no discontinuities were recorded.

Resource reservation was implemented in the server and on the network using priority service for the multimedia sessions. In order to maximize the reservable bandwidth for the single, high priority server, a change was implemented in the IEEE 802.2 protocol, which is not described in this paper.

The server reservation in the prototype is file-oriented rather than session-oriented though we have designed the extensions to support dynamic, session-oriented resource reservation. In order to support many multimedia streams having different periods to the same application, some indication of the period (or deadline) needs to be conveyed between the server and its transport [11, 4]. There are two ways in which this can be accomplished: Either a separate session or connection is established for each stream, or the streams must be differentiated across the API. We have chosen the former, and have implemented a **reserved-send** operation, so that the deadline can be passed across the transport interface. B-ISDN ATM solves this problem using a virtual channel and virtual path relationship where each virtual channel has its own QoS and there is a 1:N mapping of paths to channels [9, 6].

Session-oriented resource reservation permits the client application to dynamically change the resource reservation. When there are many opened files which are never concurrently played, or when it is desirable to release reservations when an application pauses, client/server resource reservation protocol is used: A file system control call from the client Redirector causes an SMB protocol Remote Procedure Call to be invoked from the client to the server requesting a change in the resource reservation for the application's session.

Resource reservation protocol is multi-layered: It can occur between the client and the server, and between the network service user and the network service provider. In our prototype, we have extended the IEEE 802.5 priority mechanism, the IEEE 802.2 message passing protocol and the file system Server Message Block protocol to achieve multi-layered resource reservation.

CONCLUSION

Extensions to the IBM LAN Server provide guaranteed service for multimedia file access. The guarantees include an upper bound on the amount of time it takes for a multimedia file READ operation, and they are achieved through disk and network resource reservation. For a large class of applications, no change is needed in the client computer to support multimedia redirection to the IBM LAN Server: When the client computer supports configurable playout buffers and READ sizes, then no changes are needed to the Redirector for reserved access to multimedia files.

Reliable playback of ten video streams to ten clients has been shown in which zero discontinuities occur when abnormally high data traffic is introduced onto the Token Ring. Up to 80% of the Token Ring bandwidth can be reserved for multimedia sessions, though an extension to the IEEE 802.2 protocol is required to protect multimedia sessions from active data sources on the ring. All software and hardware used in the client workstations are commercially-available and unchanged for the test except for instrumentation to detect buffer underflow. Resource reservation and priority transmission occur in the server computer only.

When playout buffers and READ sizes are not configurable, these functions must be provided in the Redirector. And when hypermedia applications open many multimedia files but do not access them concurrently, then session-oriented resource reservation protocol exchange is needed between the client and server for dynamically allocating resources on a session, rather than file-oriented basis.

ACKNOWLEDGEMENTS

The authors wish to thank Baiju Patel, IBM Research, and the anonymous reviewers for helpful criticisms of an earlier draft of the paper. We also wish to thank Dennis Walker of Ralph Kirkley Associates who ran the Token Ring tests, and Victoria L. Humphrey, of IBM Personal Systems Products for LAN Server and Token Ring performance analyses.

REFERENCES

[1] K. Amer, K.J. Christensen, T. Toher, Multimedia applications on IBM 16-Mbps Token-Ring LAN's, IBM Networking Systems, Research Triangle Park, N.C., 1992.

[2] D.P. Anderson, S. Tzou S., R. Wahbe, R. Govindan, M. Andrews, Support for continuous media in the DASH System, Proceedings, The 10th International Conference on Distributed Computing Systems, Paris, France, 1990, pp. 54-61.

[3] M. Baugher, Media transports and distributed multimedia flows, Proceedings of the 1992 SIGAPP Symposium on Applied Computing, ACM Press, 1992, pp. 1132-1143.

[4] D. Ferrari, D.C. Verma, A scheme for real-time channel establishment in wide-area networks, IEEE Journal on Selected Areas in Communications, vol. 8., no. 3, April 1990.

[5] D. Ferrari, Client requirements for real-time communications services, IEEE Communications Magazine, Nov. 1990, pp. 65-72.

[6] R. Händel, M.N. Huber, Integrated Broadband Networks - An Introduction to ATM-based Networks, Addison-Wesley, 1991.

[7] R.G. Herrtwich, R. Nagarajan, C. Vogt, HeiRAT: The Heidelberg Resource Administration Technique - Design Philosophy and Goal, Kommunikation in Verteilten Systemen, Gesellschaft für Informatik, Munich, March 1993.

[8] K. Jeffay, D.L. Stone, F.D. Smith, Transport and display mechanisms for multimedia conferencing across packet switched networks, Technical Report, University of North Carolina at Chapel Hill, Draft of October 20, 1992.

[9] J-Y. Le Boudec, The asynchronous transfer mode: a tutorial, Computer Networks and ISDN Systems 24 (1992), North Holland, pp. 279-309.

[10] P.F. Linington, The virtual filestore concept, Computer Networks 8 (1984), North Holland, pp. 13-16.

[11] C.L. Liu, J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, JACM, Vol 20, No. 1, January 1973, pp. 46-61.

[12] Microsoft Corporation, Microsoft Windows Programmer's Reference Library, Microsoft Press, Redmond, Washington, 1991.

[13] Multimedia Presentation Manager/2, Programming Guide, International Business Machines Corporation, June 1992.

[14] Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media Internation Organization for Standards, ISO Draft Standard, ISO DIS 11172, rev 2, ISO 1992.

[15] W.L. Nusbickel, J.P. Albee, Networked full motion digital video, OS/2 Developer, Fall 1992, pp. 81-89.

[16] Report of the meeting held in Geneva from 9 to 20 March 1992 - Part II.55 - New Recommedation Q.933, Study Group XI - Report R 240, CCITT, COM XI-R 240-E, April 1992.

[17] P.V. Rangan, H.M. Vin, Designing File Systems for Digital Audio and Video, Proc. 13th Symp. on Oper. Sys. Principles (SOSP '91), Oper. Sys. Rev. Vol. 25, No. 5, pp 81-94, Oct. 1991.

[18] P.V. Rangan, H.M. Vin, Designing an On-Demand Multimedia Service, IEEE Communications Magazing, July, 1992.

[19] G.D. Ripley, DVI-a digital multimedia technology, CACM, Vol. 32, No. 7, July 1989, pp. 811-821.

[20] R. Steinmetz, Synchronization properties in multimedia systems,IEEE Journal on Selected Areas in Communications, vol. 8., no. 3, April 1990.

[21] N. Strole, IBM Networking Systems, personal communication.

[22] P. Zafiropulo, H.R. Mueller, F. Closs, Data/voice integration based on the IEEE 802.5 Token-Ring LAN, IBM Research Report, RZ 1463 (Log #53321) 4/25/86, 24 pages.