Université d'Ottawa · University of Ottawa

ATM RendezView:

Enabling Real-Time Multimedia Delivery to-the-Desktop

By:

Keith M. Smith, B.A.Sc.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering

Department of Electrical & Computer Engineering
Faculty of Engineering
University of Ottawa

0-612-26363-0

Acknowledgments

The success of ATM RendezView has relied heavily on a collaborative effort with my colleague and personal friend, Russel Pretty. Not only was Russel's knowledge, expertise and industry experience an asset to this project, but he was a source of continual support throughout. As a mentor, his motivation and encouragement enabled me to explore new ideas and develop innovative new concepts.

I also owe a special thanks to my manager and colleagues at Nortel Technology. Peter Ashton, my immediate manager provided me with the necessary leadership and resources to explore and discover on my own impetus. My colleagues at the COBRAnet lab were also invaluable for their support and technical guidance.

Dr. N. D. Georganas, my thesis supervisor along with all my colleagues at the University of Ottawa's Multimedia Communications Research Lab are also to be credited for their individual support. In addition, Monica Lovas of Nortel should be commended for her efforts as my chief editorial reviewer, her ideas and advice were invaluable and fully appreciated.

I would also like to commend Nortel's efforts in its support of academic research and advancement. Through my research at Nortel Technology, I have learned a great deal, both from a technical and business standpoint. I have also been able to make, what I consider to be, a significant contribution to this corporation as a whole. Nortel's efforts to recruit fresh minds as well as original ideas is well served by this type of program. It enables unique and potentially lucrative ideas to take shape, which would otherwise be discouraged due to their misalignment with the core business values. I hope that my endorsement of this program will encourage its on-going success in the future.

Abstract

Throughout the last decade, computer processors, memory, networking and storage devices have consistently increased in performance, capacity and efficiency in a nearly exponential fashion. With the incredible rate at which technology advances, communication system developers are consistently challenged to both improve and reassess approaches used to realize real-time communication systems. With the introduction of new technologies comes renewed opportunities to capitalize on the re-engineering of existing systems to achieve more cost effective and better featured solutions.

The upheaval underway in the video conferencing market is one such example. Initial video conferencing products released in the late '80s and early '90s by companies such as PictureTel Corp., Compression Labs. Inc., GPT and Video Telecom employed the "Px64" approach. These aggregated multiple (P) DS0 (64 Kbps) channels on switched or leased-line TDM networks in order to achieve a wideband channel (128 Kbps to 2 Mbps) necessary for video conferencing. Despite falling prices for CODECs, the need for expensive hardware (channel aggregation Inverse Multiplexors and Multipoint Control Units) along with hefty bandwidth based tariffs has limited the growth of the video conferencing market.

The advent of highly performant computer technology necessary to process real-time audio and video media streams has recently enabled video conferencing services to migrate to the desktop. Additionally, as networking technologies mature to the point of affording the real-time, high bandwidth transport of these streams, a new breed of distributed multipoint conferencing systems will become widely used.

ATM RendezView is a prototype application which takes a radical shift in thinking to provide a high quality, distributed, multiparty video conferencing environment. No longer are media stream bridging, mixing and distribution functions resident in a centrally located Multipoint Control Unit (MCU), but distributed among end-systems. Real-time network services are provided through the use of native Asynchronous Transfer Mode (ATM) connectivity to-the-desktop. Deployment of such a distributed service is consequently less costly than with MCU based systems.

ATM is instrumental to the success of the prototype since it provides dynamic point-to-multipoint Switched Virtual Connections (SVC) for media stream distribution, each connection having a negotiable Quality of Service (QoS). Dynamic point-to-multipoint connections empower each conferee to independently choose particular media streams to be received from other conferees, thus affording the function of switched presence control. Processing-intensive functions such as audio mixing and video decoding are off-loaded onto the end-systems which exploit real-time facilities such as threads and real-time scheduling to meet processing deadlines.

In addition to demonstrating the superior value of ATM networks for real-time multimedia communications, ATM RendezView proves that recent technological advances in hardware and software have come to a point where the development of high quality real-time multimedia applications is a reality. All that is required are some novel approaches to solving some age old technological problems.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation

Computing, networking and other related technologies have all seen significant change in their technological paradigms. Computers have become significantly more powerful, acquired new graphically-intensive operating environments and begun to address concurrency and parallelism within their operating structure. Similarly, networking technology has seen a significant performance increase as well as new technologies addressing the real-time nature of transmitted data. These technology evolutions have together paved the way for new *real-time oriented multimedia applications* to evolve.

On a large scale however, little work has been done to begin this evolutionary process. Perhaps due to costs, lack of knowledge or faith, or lack of a solid standards infrastructure for multimedia, application developers have been apprehensive about this potentially lucrative market. Instead, the only products which have been produced have contained non-real-time multimedia content, such as entertainment products and educational materials. A common trait shared among these is that each employs a rich set of media types, such as audio, video and graphics in their presentation. However neither have attempted any level of interactivity or presentation fidelity which would impose the need for real-time constraints. It is essentially these two components of multimedia, namely *bandwidth* and *real-time constraints*, which separate the low quality, low interactivity, static multimedia presentations from the high fidelity, multiparty, dynamic multimedia environments. It has yet to be proven whether today's technologies have the potential to support the latter.

1.2 Goals

A rather ambitious goal was therefore proposed: *Using common available technologies, design and prototype a multimedia application necessitating hard real-time performance.* The first step was to determine an application which would necessitate both bandwidth-intensive media processing, and stringent real-time constraints. A multiparty video conferencing application was a challenging choice due to the following characteristics: its need for high interactivity among several conference members, its inherently bandwidth-intensive audio and video streams, and the real-time constraints imposed by end-to-end latency and jitter restrictions required to enable interpersonal communication. In addition, the implementation of a multiparty, high quality video conferencing environment using a computing platform (as opposed to a proprietary hardware based system) had not been attempted previously. The distributed nature of computers could potentially offer major advantages in terms of features and costs of a potential product.

The second step in this process was to select the necessary technologies both for computing and networking. Real-time processing characteristics were a primary requirement of the chosen technologies. A Sun SPARC 20 workstation running the Solaris 2.4 operating system was chosen due to its superior processing bandwidth and support for the POSIX real-time processing standard. Asynchronous Transfer Mode (ATM) was chosen as the networking technology due to its various real-time traffic management facilities. Parallax Graphic's PowerVideo Motion-JPEG CODEC was selected in order to perform processor-intensive video compression routines, thereby releasing the host computer's CPU for other functions.

Initially, several doubts surfaced which questioned the feasibility of such a system: Could a non-deterministic operating environment such as UNIX be expected to support a real-time application? Could ATM live up to all the hype, naming it as the panacea of real-time networking technologies, able to service all types of media? Could a workstation actually exploit the features offered by a real-time networking technology such as ATM? Or would this sort of application require new technology, such as dedicated hardware modules or faster computers in order to function properly? The ultimate challenge of this project however, was to determine whether all these isolated technologies could interwork in order to realize a multiparty, real-time, multimedia application. This was the primary goal of the ATM RendezView prototype.

1.3 ATM RendezView

ATM RendezView is a prototype application which takes a radical shift in thinking to provide a high quality, distributed, multiparty video conferencing environment. No longer are media stream bridging, mixing and distribution functions resident in a centrally located Multipoint Control Unit (MCU) as in legacy systems, but distributed among computing end-systems. Real-time network services are provided through the use of native Asynchronous Transfer Mode (ATM) connectivity to-the-desktop. Deployment of such a distributed service is consequently less costly than with MCU based systems.

ATM is instrumental to the success of the prototype since it provides dynamic point-to-multipoint Switched Virtual Connections (SVC) for media stream distribution, each connection having a negotiable Quality of Service (QoS). Dynamic point-to-multipoint connections empower each conferee to independently choose particular media streams to be received from other conferees, thus affording the function of switched presence control. Processor-intensive functions such as audio mixing and video decoding are off-loaded onto the end-systems which exploit real-time facilities such as threads and real-time scheduling to meet processing deadlines.

In addition to demonstrating the superior value of ATM networks for real-time multimedia communications, ATM RendezView proves that recent technological advances in hardware and software have reached a point where the development of a high quality real-time multimedia application is a reality. All that is required are some novel approaches to solving some age old technological problems.

1.4 Thesis Organization

This thesis is divided into five major chapters following this introduction. Chapter 2 (*Current State of Multimedia and Systems)* provides a cursory introduction to multimedia systems as they stand today, as well as an in-depth overview of video conferencing systems and their implications. Chapter 3 (*Requirements for Real-Time Multimedia)* attempts to remove much of the ambiguity which surrounds the terms *real-time* and *multimedia* and to provide a solid set of requirements for what *real-time multimedia* implies, especially within the context of the ATM RendezView prototype. Chapter 4 (*Asynchronous Transfer Mode)* endeavors to provide an introduction to ATM technology, as well as its operation in the context of ATM RendezView. It is by no means a complete tutorial on ATM, however it outlines many of the fundamental facilities therein. Chapter 5 (*ATM RendezView Prototype)* is the heart of the thesis. It addresses the prototype's design concept, architectural design approach, various implementation issues and methodologies, as well as relevance to industry standards. Chapter 6 (*Future of Real-Time Multimedia Development)* highlights the major barriers facing real-time multimedia development, especially as applied to multimedia communication systems and provides a view of future services which will evolve from research in the field of real-time multimedia. Chapter 7 (*Summary and Future Work)* concludes the thesis by providing a brief summary, and a view of future work which should enhance this research effort.

1.5 Main Contributions

This thesis provides a means for anyone with basic multimedia computing and networking knowledge to quickly educate themselves regarding the important issues of real-time multimedia application development. First, a fundamental introduction to multimedia communications and real-time networking is provided, highlighting some of the most important issues. Second, a pragmatic approach is taken to the development of a sample real-time multimedia communications prototype as a "proof-of-concept". This sort of attack on the material is seldom found in literature, which usually concentrates on component theories, without ever bridging the gap between components. Third, an attempt is made throughout the thesis to highlight the many barriers facing this area of research, which must be surmounted so that the design and development of real-time multimedia systems succeed. Most important, the work found herein should be proof enough that real-time multimedia system development is possible given our current technology, however much effort is required in order to mature this field from its current infant state.

1.6 Authors

The success of the ATM RendezView prototype is a demonstration of the potential of industry and academic collaboration. This project evolved from the efforts of two employees of Nortel Technology's Broadband Networks department, who are challenging their Master's degree in Electrical Engineering in conjunction with the Multimedia Communications Research Lab (MCRLab) at the University of Ottawa, Canada. The two architects, Keith Smith (author of this thesis) and Russell Pretty [Pretty97], applied their respective talents to the project's success. Conceptual design was performed as a collaborative process, however implementation design, development and performance testing were divided between the two. Mr. Pretty's efforts concentrated primarily on the requirements of real-time video, namely the coding, transmission, synchronization and display. Mr. Smith's work addressed audio issues, which also required a detailed investigation of real-time processing issues within non-deterministic computing environments. Each author was able to bring forth their unique expertise, which served to compliment the project as a whole. Both authors owe their gratitude to both Nortel Technology and the University of Ottawa for their overwhelming support.

Chapter 2: Current State of Multimedia and Systems

2.1 Introduction

The term *Multimedia* has been used to describe the fastest growing trend in computer technology today. This trend involves the incorporation of non-textual media types such as graphics, images, audio and video into computer based applications. No matter the context, multimedia can be assured to guarantee a departure from the archaic text-based systems of the past and an introduction to rich media types in the future. This chapter attempts a cursory introduction to multimedia systems as they stand today, as well as an in-depth overview of video conferencing systems and their implications.

2.2 The Many Faces of Multimedia

2.2.1 Applications

Multimedia applications are those which incorporate various media types in addition to standard text in their construction, which seems to include every application environment released today. The media types typically employed include graphics, images, audio and video sequences, however the style and arrangement of their presentation is what makes each unique. Media types catering to our senses of smell, taste and touch have not been explored at this time. Given the subset of media types that are being employed today, several classes of applications may be defined:

- Games

Computer video games were the first applications to actually employ graphics and sound. The quality of audio and video originated as simplistic and monotone, however with the superior processing capabilities of computers today, new three dimensional virtual environments are emerging. Players are now able to be immersed in a virtual world where they may control the navigation and actions of a virtual character. Battles between multiple players, all remotely located, are now possible complemented by realistic graphical animation. Audio quality has also increased to such a point where lifelike sounds may be generated resulting in a highly realistic experience. New games are always challenging the limits of the fastest computing systems available by increasing both the resolution and depth of the media types employed. The days of arcade based video games are gone. New computing technologies are now capable of achieving higher fidelity and outperforming the electronic arcade games of the past.

- Publishing

Publishing applications have also embraced multimedia by adopting audio, video and image content capability. With the release of software packages such as Microsoft Corporation's Microsoft Office and Corel Corporation's Corel Suite, users have been armed with tools able to process multimedia content. Common to these packages are word processing, spreadsheet, database, and presentation tools. Also, advanced drawing or authoring tools are commonly bundled within these packages.

A significant trend which has emerged recently in publishing software is the *package* or *suite* concept. Software vendors are realizing that there is a core group of tools required in the typical office environment (namely word processing, presentation, spreadsheet and database) and are releasing software packages/suites containing all of these tools. These tools do not function autonomously within the package, but actually work together to enable the seamless transfer of data among each other. Facilities such as clipart libraries, spelling/thesaurus dictionaries, and personal configuration information can all be shared and exchanged between the various tools. In addition, documents created with one tool may be shared with another. For example, a graph created within a spreadsheet tool can be easily incorporated into a year-end report created using a word processing tool.

Another common feature which has emerged is the ability to perform some form of *object linking* or *imbedding*. An object in this context, can represent any media type such as graphics, audio, video or even another published document. One or more documents can contain a link to such an object enabling the object's incorporation into each document. For example, a slide presentation could contain a video or audio clip which could be played at presentation time. The advantage of such a linking mechanism occurs when the object is updated or changed. All documents referencing this object need not be updated for new changes to take affect.

- Authoring and Development Environments

Authoring and development environments were one of the first applications to evolve out of the introduction of multimedia capabilities within computers. The superior value of multimedia presentations versus the standard text-based style has long been recognized, especially when applied to Computer Based Training (CBT) and distance learning. For this reason, a simple and efficient means of creating such presentations was required and subsequently gave rise to authoring environments.

An authoring environment is a program which has pre-programmed elements for the development of interactive multimedia. Authoring systems vary widely in orientation, capabilities and required learning curve [Siglar96]. In general, the author uses a graphical interface to specify the characteristics and flow (multimedia elements, sequencing, hotspots, interactions, synchronization etc.) of a multimedia presentation by employing an authoring paradigm. Typical paradigms include scripting languages, iconic/flow control, frame based, card/scripting, cast/score/scripting, hierarchical object or hypermedia linkage paradigms.

There are many authoring products available for all types of computing platforms, each exploiting a different authoring paradigm. Authorware Professional and Director from Macromedia are two popular products available for both Macintosh and Windows based computers. Although most authoring packages allow users to create content using their built-in tools, these tend to be rudimentary when compared with those available in dedicated programs. It is advisable that content be created and edited in dedicated programs and imported into the authoring environment. Programs such as paint/draw, video digitizing/editing/effects, audio sampling/editing and the publishing suites mentioned above can be invaluable for content creation.

- Animation and Virtual Reality

Animation and Virtual Reality (VR) form a class of multimedia applications on their own. Both deal with the graphical rendering and display of objects to give them a lifelike appearance while enabling realistic and fluid motion. There are fundamental differences however. Animation usually follows a static sequence of steps to produce a two dimensional (sometimes three dimensional) illusion of motion. It usually consists of batches of individual images played out in close sequence. Minimal processing is required for these systems since animation primarily involves moving images from secondary storage to a display device.

Virtual reality on the other hand is an interactive form of animation. It uses real-time rendering of objects to create subsequent frames within an animated sequence. Due to its need for high interactivity and real-time rendering, VR systems are processor intensive. They typically involve three dimensional worlds which can be manipulated by a user.

In order to decrease the processing load incurred by real-time rendering, several techniques are employed. Instead of rendering full definition images, geometrical shapes such as cubes, cones, cylinders and spheres are exploited to approximate the desired image. This reduces processor load substantially, while resulting in reasonable approximations of the original image. Once images are composed of geometrical shapes, operations such as scaling, rotations and translations are simpler to perform. Other techniques employed in rendering to impart lifelike characteristics on objects, include shading, texturing and lighting which are more easily performed on geometric shapes than bit-mapped images.

VR has received a great deal of media attention in the past, however it still has a long way to mature. Typical applications of VR today reside primarily in the field of visualization. Some examples include architects requiring a more realistic rendering of their proposed designs, or aircraft pilots clocking hundreds of virtual flying hours in a safe environment [Young96]. A few VR based games have also emerged, allowing players to navigate through a fantasy world using a VR glove or visor to control their actions. True VR systems today require expensive, high performance workstations to operate. However with the advancements demonstrated in processor design and performance today, VR systems will soon become commonplace for the generic desktop computer users.

- Audio and Video Conferencing

Audio and video conferencing, commonly referred to as *video conferencing*, forms a unique class of applications which take multimedia to a new dimension. Unlike previous multimedia classes, video conferencing imposes time-critical deadlines on the capture, transmission and display of multimedia data. A video conference involves two or more geographically distributed conferees, who may both see and hear all other conferees participating within that conference. Because of this multi-directional mode of communication, certain performance constraints are imposed on the supporting computing and network hardware. In order for a remote speaker to be perceived as contributing to a conversation in a natural fashion, lip synchronization (intermedia synchronization) and delay/jitter (intramedia synchronization) must be controlled. Subjective tests have shown that a natural conversation cannot proceed if this is not the case.

Most video conferencing applications offer other features in addition to audio and video connectivity between conferees. They offer a shared whiteboard tool which is essentially a global scratchpad on which any conferee may add or remove text or images. Each conferee is able to view the contributions made by all other conferees, which is useful for brainstorming/work sessions. Conferencing environments should also enable users to share arbitrary third party applications. A conferee may, for example, wish to share a CAD drawing that has been created, in order to get feedback or even to enable other conferees to edit this drawing. File transfers and textual messaging between conferees should also be possible.

Many applications belonging to a subclass of video conferencing, are becoming available, especially through the internet. They typically employ only subsets of the features mentioned above and have significantly relaxed real-time requirements. Such applications include audio phones, one-way video broadcasts and Video-on-Demand (VoD) services. It should be emphasized that true video conferencing environments impose the most stringent real-time constraints on their supporting technology. The failure of the supporting technology to respect these real-time constraints is the most common failing of conferencing environments today. Refer to section 2.3 for a detailed analysis of video conferencing systems.

- Hypertext and Hypermedia Browsing

Browsing multimedia databases using hypertext and hypermedia tools is the most popular form of multimedia application today. With the evolution of the World Wide Web (WWW) in the past decade, there are few people who have not heard of the WWW, let alone browsed it. The value of information cannot be over stated. Whether for advertising, education, services or communications, the need for global access to heterogeneous databases is an absolute requirement.

Hypertext refers to the removal of linear constraints imposed by most textual information, via mechanisms enabling rapid jumps or links between strategic points within linear textual documents. Hypermedia is similar, however the linking is not necessarily limited to textual media. It can be a link to an image, audio, video clip or other media type.

The most common form of hypermedia browsing is using a browser such as the original Mosaic conceived by CERN and subsequently developed at the NCSA (University of Illinois), or the popular Netscape browser produced by Netscape Communications. Each is based on the Hypertext Markup Language (HTML) which enables authors to create hot-spots or hot-links within documents, which when selected using a mouse or other pointing device, instantly issues a request for the retrieval and display of the requested media. In addition to linking, HTML allows users to create rich multimedia documents through the incorporation of colors, fonts, images, graphics, tables, frames, forms and more.

The features offered through WWW browsers are continually expanding. No longer are browsers limited to the archaic client/server architecture of the past. Netscape offers the ability for third party developers to produce *plugins* in order to extend the functionality of its browsers. Hundreds of plugins now exist for functionality spanning multimedia, graphics, sound, documents, productivity and VR. Sun Microsystems has developed a new platform-independent programming language called JAVA. Because of its platform independence, a browser can retrieve a piece of java code (called an Applet) and run it on the local computers as an independent executable program. The growth of the WWW, supporting browsers, and multimedia databases show little sign of ebbing in the near future. New features and tools promise to make the world wide web an entertainment medium for the consumer, a playground for the hobbyist, and an educational tool for the academic.

From the extensive list of applications above, it may seem that multimedia applications have made great progress. This is true to some degree, however multimedia applications as a whole are still well within their infancy. Most multimedia applications released today are experimental in nature, unstable and require significant additional research. Real-time aspects of these applications are seldom addressed, primarily due to non-determinism of supporting computing technology. Clearly, multimedia is a powerful technology full of promise, however it has significant maturing to do.

2.2.2 Hardware

The need for supporting "add-on" hardware technology for computers was quickly recognized as early designers battled with the inherently high bandwidth, processor-intensive, real-time nature of multimedia data. The standard processing operations required to process any multimedia data include analog-to-digital (A/D) conversions, digital-to-analog (D/A) conversions, compression, decompressing, mixing and memory copying. These operations would even be involved in the simple process of capturing a live media source (such as analog audio or video), converting it to a digital representation, and sending it to a storage or output device (such as memory, hard drive, network interface card, speaker or video display). Due to the high bandwidth, real-time nature of multimedia data, current computer technology is ill-equipped to process multimedia data.

Supporting hardware technology, in the form of "add-in" boards, designed to perform particular tasks in an optimal fashion were deemed necessary. In this fashion, the main processor within a computer could be relieved of much of the processing intensive routines required by multimedia data. The first products which appeared in the late '80s were sound cards. These had the ability to sample and digitize external audio signals at CD quality in real-time. These would enable users to perform studio quality audio recording and editing on their home computers. Some sound cards also contained facilities for performing real-time compression/decompression of audio. With the incorporation of the MIDI standard into sound cards, musicians were empowered to compose complete musical scores from their desktops.

Sound card technology caught a lot of attention in the home computer market, which soon led to video hardware technology in the early '90s. These cards incorporated a dedicated processor to produce faster and smoother graphics processing and display, resulting in decreased load on the computer's main processor. Recently, video hardware has emerged to support various video COmpression/DECompression (CODEC) standards such as Motion JPEG and MPEG for the purpose of video conferencing and VoD.

Supporting multimedia hardware, primarily for Input/Output (I/O) has also emerged in past decade. These include scanners (some implementing optical character recognition), CD ROM drives, touch screens, pointing devices, VR glove/head gear and audio/visual accessories such as cameras and headsets. As technology advances, multimedia hardware benefits by becoming faster, cheaper, smaller and of higher quality.

2.2.3 Networking

Networking support for multimedia based applications has unfortunately not seen the same rapid technological advancement as multimedia itself. Most network dependant multimedia applications such as WWW browsing and video conferencing require varying degrees of high bandwidth and low latency network connections to adequately service multimedia traffic. Computer users have unfortunately been limited in their access to affordable, ubiquitous high bandwidth network connectivity. In addition to low bandwidth connectivity, network users are at the mercy of ancient networking protocols, ill-suited to the requirements of real-time multimedia. The most widespread networking protocol suite used today is the Transport Control Protocol/Internet Protocol (TCP/IP), a *de facto* standard which was developed almost three decades ago (initiated in 1968 as part of the US government's Advanced Research Project Agency Network - ARPANET) for the reliable transmission of static computer data. Unfortunately, TCP/IP does not have any provisions for servicing real-time multimedia data, which has seriously hindered any substantial advancement in the development of real-time multimedia applications.

A video conferencing application using ITU's H.261 audio and video encoding standards would require a bandwidth in the range of 128 Kbps to 2 Mbps in order to provide any reasonable quality. Most home users using modems for dialup access to an Internet Service Provider (ISP) today would typically have a maximum bandwidth of 28.8 Kbps, a couple orders of magnitude below any bandwidth which could be considered adequate for real-time multimedia communication. Many home users now have access to pricy Integrated Services Digital Network (ISDN) services offering 128 Kbps. High bandwidth leased lines are also available at various speeds, however these are extremely expensive and typically only used by corporate customers. Future technologies such as Asymmetrical Digital Subscriber Loop (ADSL) will offer access bandwidth in the range of 2 Mbps to 50 Mbps to the home. Users of corporate networks are in the best situation currently since LAN and WAN technology typically offers shared bandwidth in the order of tens of Mbps. Corporate networks can also be cost effectively resized to support real-time multimedia applications as required.

Network transport protocols have been targeted recently in standards bodies such as the Internet Engineering Task Force (IETF) for upgrades. Protocols such as IPv6, Resource reSerVation Protocol (RSVP), and Real-Time Protocol (RTP) are all examples of the ongoing efforts being made to make legacy networks real-time capable. Higher speed transport and access technologies are also emerging such as Asynchronous Transfer Mode (ATM), gigabit ethernet, IP switching and tag switching. The future of real-time networking is quite promising, however it is yet to be seen which networking technologies will dominate.

2.2.4 Standards

On going standards work in the field of multimedia systems can be broken down into two broad categories. The first being those standards which concentrate on specific areas of expertise such as video or audio compression. The second category defines standards for multimedia platforms and environments in order to assure compatibility across heterogeneous computing architectures.

The first category which develops isolated standards for technology ranging from encoding algorithms to networking solutions has received the most attention. Table 2-1 lists some of the more popular standards developed by standards bodies.

Table 2-1 Common Multimedia Standards

Standard Body	Standard Name	Description
International Telecommunications Union (ITU)	H.221, H.242, H.261, H.320	Complete specification of data coding and transmission for conferencing and audio/visual teleservices on multiples of 64 Kbps channel networks
International Telecommunications Union (ITU)	G.711, G.721, G.722, G.725	Defines audio digitization, encoding and compression standards
International Telecommunications Union (ITU)	T.120 (T.121 - T.124)	Specifies a complete protocol suite for network independent audio/video conferencing
International Standards Organization / International Electrotechnical Commission (ISO/IEC)	Motion JPEG (Joint Photographic Experts Group)	Defines a video compression algorithm for high quality image transfer
International Standards Organization / International Electrotechnical Commission (ISO/IEC)	MPEG (Motion Picture Experts Group)	Defines audio and video compression algorithms for real-time transfer at various resolutions and data rates
Internet Engineering Task Force (IETF)	IP Multicast (RFC 1112)	Specifies extensions required to a host implementation of the Internet Protocol (IP) to support multicasting
Internet Engineering Task Force (IETF)	Real Time Protocol (RTP)	A transport protocol for audio and video conferences and other multiparticipant real-time applications over IP
Internet Engineering Task Force (IETF)	Resource Reservation Protocol (RSVP)	A network resource reservation protocol for IP networks suited to real-time multimedia services
ATM Forum	Asynchronous Transfer Mode (ATM): UNI, PNNI, I-PNNI, TM	Networking standards specifying signalling, routing and management protocols for the real-time transport of time sensitive data

These standards are extremely important since they provide the fundamental building blocks for developing multimedia systems. If a video conferencing system is to be built, protocols are required for media encoding, compression/decompression, and display, in addition to protocols for functionality such as signalling and network transport. Standards are an attractive alternative to proprietary schemes, since they allow a great time saving for developers through technology reuse (no "reinventing the wheel"), and are a healthy step towards the development of interoperable systems.

Standards development should not stop here however. Multimedia system interoperability will never be guaranteed through the use of low level standards alone. There need to be standards available defining a *multimedia system*, including both hardware and software, at a high level. These are the plights of newly formed standards groups such as the Interactive Multimedia Association (IMA), the International Multimedia Teleconferencing Consortium (IMTC) and the Digital Audio Visual Council (DAVIC). All these groups have a similar mandate of bringing together all organizations involved in the development of interactive, multimedia teleconferencing products and services to help create and promote the adoption of industry-wide interoperability standards. They typically capitalize on the large existing base of low level multimedia standards to produce a specification defining a complete multimedia system. In this fashion, they align themselves with existing standards, yet provide interoperability across various computing architectures.

Standards are extremely important for cross-platform interoperability of multimedia systems. As our multimedia applications become more complex and incorporate additional media types, standards for interoperability will become a much greater requirement.

2.3 Video Conferencing

2.3.1 Conferencing Environments

The architecture of the conferencing environment chosen is predominantly determined by the type of network service available. This has ramifications in terms of conferencing quality (frame rate, resolution, maximum number of conferees, maximum number of visible conferees at one time) and supporting features available through the environment (whiteboard tools, collaborative tools). The trade-off between network service tariffs and the desired conferencing richness determines which of the three available conferencing architectures will be implemented. These are Wide Area Network (WAN) based, Plain Old Telephone System (POTS) based and Local Area Network (LAN) based conferencing architectures.

2.3.1.1 WAN Based Conferencing Architecture

The WAN based conferencing architecture has been in existence the longest and has the largest market segment at this time. This architecture exploits network services consisting of multiple 64 Kbps (DSO) communication channels. It is based on the ITU's H.320 series of conferencing standards, also known as "P x 64" (P indicates the number of 64 Kbps channels used). This standard defines everything from media encoding, compression and synchronization algorithms to remote camera control and multipoint signalling.

The H.320 series of video conferencing standards was developed with the understanding that only very low bandwidth network services would be available to users, typically through the currently installed base of phonelines. As such, H.320 exploits algorithms which completely remove any redundancy from audio and video streams and provide extremely high levels of compression. Most H.320 based conferencing systems are implemented in proprietary hardware modules in order to meet performance requirements of real-time compression algorithms.

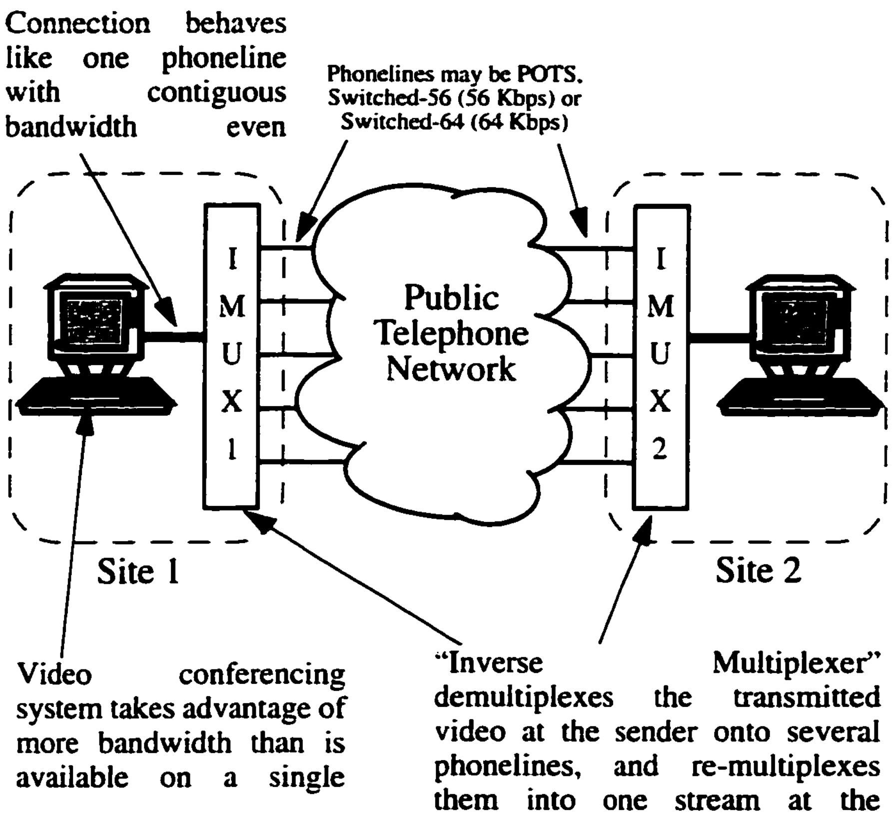
The goal of these systems is to provide sufficient quality such that conferees may communicate effectively with each other. This means that video frame rates can be as low as one to two frames per second, with equivalent frame resolutions. Audio is placed at a higher priority since it is vital to inter-personal communications, however even the audio integrity is sometimes lost. H.320 is usually implemented as a group conferencing node in a specially designed meeting room, or as a "roll-about" system. The luxury of personalized, individual conferencing nodes is not typically afforded by the available bandwidth/financial resources.

The H.320 based conferencing environment exploits DSO based networking services such as switched-56 (56 Kbps), switched-64 (64 Kbps), ISDN (128 to 144 Kbps), fractional T1/E1 (384 Kbps typical), or leased T1/E1 (1.5 Mbps/2 Mbps). It is designed to adapt to the available bandwidth ranging from 128 Kbps (P=2) up to 2 Mbps (P=30). At bandwidths below 384 Kbps, video image quality and frame rate is low, and audio may become choppy and incomprehensible.

Even at bandwidths above 384 Kbps, current systems struggle to provide acceptable quality. It is generally accepted that for reasonable video and audio quality, a minimum of 600 Kbps (P=10) is required, however this is far from common for cost-of-service reasons.

Because of the high cost of network services, a typical means of achieving bandwidth capacity in excess of 64 Kbps in a point-to-point topology, is to perform channel aggregation of multiple low speed links (POTS, switched-56, or switched-64). As shown in Figure 2-1, an Inverse-MUltipleXer (IMUX) is employed at each end-point to multiplex/demultiplex a high speed link into multiple low speed links through the Public Switched Telephone Network (PSTN). IMUXs are very complex components since in addition to reassembling multiple low speed links into one higher speed link, they must compensate for any delay differences which exists between the various phonelines employed. Consequently, these components are quite expensive.

Figure 2-1: IMUX Configuration for H.320 Based Systems



In addition to costs of network services, another significant cost is contributed by the need for a conference bridge or Multicast Control Unit (MCU). H.320 defines the capabilities for video conferences of up to 28 conferees, however when there are more than two conferees, a conference bridge/MCU must be employed. The MCU performs audio mixing and video switching such that each conferee may hear every other conferee, but see only the current speaker. The MCU is necessary since each conferee doesn't have the necessary access bandwidth to receive all transmitted media streams. The need for costly IMUX and MCU hardware is one of the major arguments against WAN based conferencing architectures.

2.3.1.2 POTS Based Conferencing Architecture

The increased popularity of the WWW in the past few years has lead to a significant demand for home access to the internet. Typically, a home user subscribes to an Internet Service Provider (ISP) for access to the internet, which has a usage based or a fixed monthly billing policy. Access is achieved over a regular POTS telephone line using a modem. The maximum bandwidth which can be achieved currently using such a configuration is 28.8 Kbps assuming no data compression. This is a far cry from the megabits of bandwidth achievable with WAN based solutions, however there is a major cost advantage.

First, most home users already have a computer for other purposes and so the additional purchase of a low cost modem is not significant. Second, the cost of internet access is extremely competitive and comparable to the cost of the POTS service itself. In fact, in some areas, community based services which are funded by local advertising can offer free internet access to subscribers. Third, most internet access is used for transmitting computer data as opposed to multimedia information, which implies a much reduced bandwidth requirement. Though POTS access does not imply "speedy" data transfers on demand, it is an acceptable solution at a price that simply cannot be beaten.

Internet users are becoming multimedia "hungry" however. With the release of multimedia games and publishing tools, internet users have had a taste, and are looking to multimedia communications for entertainment and productivity. This means that users are wanting to achieve quality audio and video conferencing over an extremely low bandwidth communications channel. As mentioned above, current technology dictates that for reasonable audio and video quality, users require a minimum of 384 Kbps of access bandwidth. However, the home consumer differs significantly from the corporate customer. The corporate customer expects a significant level of quality and is therefore prepared to pay a premium for it. The home consumer on the other hand is not willing to pay any premium whatsoever and would willingly accept significantly reduced quality. This trend has led vendors to design new audio and video multimedia applications to operate over extremely low bandwidth network connections.

The very first conferencing applications grew out of the IETF's interest in broadcasting its regular meetings to internet users, which gave birth to the Multicast backBONE (MBONE). The MBONE exploits IP multicast to distribute audio and video information throughout the internet. Because of the unpredictable, low bandwidth nature of the internet, a fair amount of research was performed into audio and video encoding and transmission for these networks. These applications provided the first arguments to prove that multimedia was possible over extremely low bandwidth networks, at significantly reduced quality however.

The MBONE has produced many tools all exploiting IP multicast for audio, video, and whiteboard, which are freeware and available to anyone with internet access and a multimedia enabled personal computer. A comprehensive list of multimedia communications tools, which includes IVS, NEVOT, NV, Picture Window, VAT, and MONET to name a few, can be found in [Gibbs96]. Many commercial companies are selling conferencing products, as well as popular internet phones which offer reduced rate long-distance calls to be placed across the internet. These include CU-SEEME from White Pines Software, WebPhone from NetSpeak Corp., and Internet Phone from VocalTec. An attractive selling feature of all of these applications is that they generally do not require any supporting hardware. Due to the low-bandwidth nature of these applications, compression algorithms are quite feasibly implemented on the host computer's own processor.

The main point to be made about all of these tools is that quality is still extremely low. Depending on the internet state, video frame rate can be as low as one frame every few minutes, image resolution can be low, and audio can be "choppy" or non-existent for extended periods of time. Most users tend to be hobbyists or experimenters who are willing to put up with the extremely low quality. Steps are being taken to try and remedy this problem which would concentrate primarily on increasing the access bandwidth to the internet. New access technologies such as ISDN (128 Kbps) and ADSL (minimum 2 Mbps) are options, however these carry significant cost premiums and it is yet to be seen whether the home user will be willing to pay for these services. It is inevitable that future network access services, based on new technologies, will offer significant market competition to the PSTN, in the arena of both telephone line service and internet access.

2.3.1.3 LAN Based Conferencing Architecture

LAN based conferencing architectures are characterized primarily by their high bandwidth network access which is typically in excess of 2-3 Mbps. These conferencing applications are installed on computing workstations which are usually connected to a packet based network, such as Ethernet, Token Ring, FDDI or ATM. Often, a workstation is directly attached to a high bandwidth LAN, however the bandwidth available to the workstation may be substantially less than the total network capacity, due to the shared nature of the network. A 10 Mbps Ethernet for example can only operate at a maximum of sixty to eighty percent of capacity. Once multiple users begin transmitting multimedia streams, the utilization of the network drops significantly, limiting the network to only two or three real-time media streams. Only ATM networks have the ability to guarantee a Quality of Service (QoS) to the application, however no real products exist which exploit these QoS aspects.

Because of the larger bandwidth capacity which is available on LANs, these video conferencing applications are designed to transmit higher quality audio and video media streams. Dedicated hardware "add-in" boards are commonly employed to reduce the processing load imposed on the workstation's processor, due to the increased complexity and bulk of media streams. Instead of exploiting the H.320 series standards for low bandwidth, low quality conferencing, LAN conferencing applications exploit standards such as Motion JPEG and MPEG for video encoding and ITU's G series recommendation for audio encoding. Refer to Table 2-2 and Table

2-3 for details on various common media encoding algorithms and their resource requirements. One unfortunate aspect of these applications at this time, is the lack of any standard guaranteeing interoperability between applications. Unlike the H.320 based applications, these may arbitrarily employ standards originating from different standards bodies or even proprietary approaches. Several standards groups have formed within the past few years (refer to section 2.2.4) to specify interoperability guidelines for multimedia applications, however no standards for high quality video conferencing have been globally adopted at this time.

Table 2-2 Resource Requirements of Audio Encoding Standards

Encoding	Analog Bandwidth	Digital Data Rate	Perceived Quality
G.711	3.5 KHz	64 Kbps	Telephone quality
G.721	3.5 KHz	32 Kbps	Telephone quality
G.722	7 KHz	48/56/64 Kbps	Superior to telephone quality
Raw Stereo CD	22 KHz	1.41 Mbps	CD audio quality
MUSICAM (MPEG Layer 2)	22 KHz	192-256 Kbps	CD audio quality
MPEG Audio Layer 3	Near CD quality	128 Kbps	Approaching CD audio quality

Table 2-3 Resource Requirements of Video Encoding Standards

Encoding	Resolution	Digital Data Rate	Perceived Quality
R.601 (Raw Digital Video)	720 pixels/line x 525 lines/frame x 30 fps	166 Mbps	Standard NTSC video
H.261	352 x 288 pixels (CIF), 5-10 fps	112 Kbps	Small image size, rapid motion may appear jerky
Motion JPEG	640 x 480 x 15 fps	3.0 Mbps	Inferior to VCR quality
MPEG-1	640 x 480 x 20-25 fps	1.15 Mbps	VCR quality
MPEG-2	Up to HDTV quality (1920 x 1152 x 60 fps)	4-30 Mbps	Superior to VCR quality. Targeted for studio quality video and HDTV

LAN based conferencing architectures are somewhat new, primarily due to recent advances in high speed ASIC technology, networking technologies to support real-time multimedia, and operating system support of real-time multimedia applications. As such, there is not the same plethora of products available. Some prominent players include InSoft Corp., InVision Systems Corp., PictureTel Corp., and Intel Corp. just to name a select few. An excellent and in-depth survey of desktop video conferencing products is available from [Hewitt96]. Most products offer compatibility with H.320 series standards in order not to be alienated, however these cannot take advantage of network bandwidth offered by higher bandwidth LANs. These products also tend to offer additional functionality such as a shared whiteboard, a slide presentation tool, file transfer capabilities, and collaboration tools such as document sharing.

Clearly, network bandwidth and services are the primary determining factor in the richness and performance of any video conferencing environment. At this point, access bandwidth is scarce due to cost issues. Once the market matures to such a point that high bandwidth access in the order of tens of megabits per second is common-place, high quality, multiparty, feature-rich conferencing environments will prevail.

2.3.2 Market State

Video conferencing environments can be implemented in one of three styles, namely a video conferencing room, a roll-about video conferencing system or a desktop video conferencing application. A video conferencing room is a total solution where a room is optimized for conferencing in its layout and interior construction. The video conferencing hardware and software is usually only a fraction of the total cost of the complete room, which can range from \$50,000 to \$500,000 [SRI93]. Roll-about video conferencing systems are less expensive (\$15,000 to \$25,000) and are not part of a larger setup in a room [SRI93]. These systems are complete packages built into one or two trollies which may be moved between rooms or offices. Basic components required by these two environments include television(s), camera(s), microphone(s), vendor's hardware and software, and a document camera for slides. Some roll-about systems use their own custom hardware and software, and others are increasingly built around an off-the-shelf personal computer. Both implement the ITU's H.320 suite of video conferencing standards for interoperability, and some also implement more efficient proprietary protocols and algorithms. Network services are typically limited to lower bandwidth WAN connectivity, which is the primary reason for employing H.320 standards.

Desktop video conferencing systems typically operate as add-on functionality to a user's existing desktop computer and may require supporting "add-in" hardware. Video display is in a window on the user's computer monitor, rather than on a separate television. These systems can vary from being freely distributed through the internet, to costing upwards of $8,000 including any required hardware. This price range however reflects the resulting quality of the audio and video transmitted since software based CODECS (which are common with internet tools) aren't as efficient or performant as their hardware based counterparts. Although these systems typically employ H.320 standards for their interoperability and suitability to low bandwidth access over PSTN, their network connectivity is also quite common through LANs in the corporate or academic environment. This makes them a candidate for higher bandwidth, higher quality media encoding algorithms, such as Motion JPEG and MPEG which are available in many desktop video conferencing products. Table 2-4 lists some common video conferencing packages, their features and approximate cost. A more complete and updated list may be found in [Hewitt96].

Table 2-4 Video Conferencing Products Summary

Vendor	Product	Cost (US$)	Audio/Video/Network Support	Comments
INRIA	IVS	Free!	PCM, ADPCM, VADPCM H.261 in software UDP/IP, IP Multicast	Available via FTP for most UNIX workstations. Also supports some common hardware frame grabbers. Supports multipoint and can run strictly in software.
InSoft Inc.	Communique!	Platform dependant	G.7xx CellB, JPEG, H.261, DVE, Indeo, H.320 Packet, Cell and Telephony Networks	Multi-platform, interoperable video conferencing suite for PCs, Macintosh and UNIX. Features collaborative tools, chat, audio, text and graphics tools. Supports multicast.
Northern Telecom	Visit Video	$5319 includes camera	Proprietary H.261 ISDN or Switched-56	Supports PC 386 Windows 3.1 and Macintosh II family of personal computers.

Table 2-4 Video Conferencing Products Summary

Vendor	Product	Cost (US$)	Audio/Video/Network Support	Comments
PictureTel	Live200 Series	$1495 includes hardware	G.711, G.728, G.722 H.261 (CIF, QCIF), H.320 ISDN	PC Windows95 based, shared whiteboard, shared clipboard, Windows95 application sharing and file transfers.
PictureTel	Live PCS 100	$4995	G.711, G.721, G.722, G.728, PT 724 (proprietary) H.261, H.320 Switched-56, ISDN, V.35/RS449 dialed and non-dialed networks	PC 386/25 Windows 3.1 and Windows95 based, features multipoint (with MCU), whiteboard, file transfer, application sharing, messaging and remote control.
PictureTel	LiveLAN	$395	Proprietary Proprietary IPX LAN	PC 486/66 Windows 3.1+, requires additional hardware (video capture card, camera, audio card) and supports application sharing.
Silicon Graphics	InPerson	$495	G.711, G.728, Intel DVI, GSM H.261, RGB8, HDCC (proprietary) UDP and TCP over IP	Based on SGI Indy workstation which contains all necessary hardware. Supports text, image and 3D model sharing. Supports other proprietary environments. Also supports PCs running Windows with a reduced feature set.
VTEL Corp.	VTEL Personal Collaborator	$2495 complete	G.728, G.711, G.722 H.320 (CIF)	PC Windows95 based system. Proprietary application sharing, file transfer and screen capture tools.
White Pine Software	CU-SeeMe	$99 retail, $66 over internet	Voxware, Digitalk, Delta Modulation White Pine 24 bit True Color, CU-SeeMe Grey UDP and TCP over IP	PC (486 with Windows3.1 minimum) or MAC (68020 with System 7.5.2 minimum) software based video conferencing. Can be used as an audio phone over a 14.4 modem or video conferencing over a 28.8 modem. Multipoint communication is possible using a proprietary Reflector.

The prices quoted for video conferencing services can be quite misleading since they are based on per-node costs only, and do not usually include the cost of network hardware or bandwidth. Such costs can be significant and can include such things as line and long distance charges for POTS service, WAN (switched-56/64, ISDN, fractional/leased T1) services, and network components such as an MCU or conference bridge. Although video conferencing may function over a standard, affordable, POTS telephone line, this bandwidth is extremely low and resulting quality is poor. Switched digital services such as switched-56 and switched-64 are more common due to their price/performance advantage and ability to connect on-demand virtually any two points on the globe. Cost of switched-56 services vary widely, however each user typically pays $100 per month as a fixed fee, and then $25 per hour for dual switched-56 services within North America [SRI93]. ISDN basic rate service can be significantly less expensive while offering equivalent bandwidth, however this technology has not been widely deployed and so connectivity is extremely limited. High bandwidth, dedicated point-to-point links at rates of 1.5 Mbps (T1) to 45 Mbps (T3) are more useful where high quality video or high volume use is desired. Charges are typically flat-rate, starting around $1000 per month, which can only be justified by high volume users.

Room based and roll-about conferencing products based on low bandwidth WAN network services have long been the dominant market solution for conferencing environments. Dominant players in this market include AT&T, Compression Labs, GEC Marconi, PictureTel and VTEL. The primary reason for the weak market presence of desktop video conferencing solutions has been the lack of a pervasive adequate networking infrastructure. Most desktop computers are connected to LANs which function to distribute data in a non-real-time fashion. Typically, an end-to-end high bandwidth, low latency network connection does not exist, or is too costly to implement. This makes a dedicated WAN based solution more attractive.

This trend is rapidly changing however. New networking infrastructures such as ATM and gigabit ethernet are evolving, offering higher bandwidth, real-time data transfers to the desktop. Unit sales of add-in boards for video conferencing have grown from 4,500 a year to about 91,600 in 1995, corresponding to a significant reduction in cost for the end user ($5000 down to $1500)

[EET96]. New computers and workstations are increasingly offering video cameras, audio support and even video compression hardware as standard items. Research done by Forward Concepts indicates that while desktop video conferencing far outstrips conference room based systems in terms of units (71,900 to 1,500), sales for the latter are still six times that of desktop systems [EET96]. Projections indicated that desktop systems will dominate by the end of the decade with 7.8 million add-ons sold to only 37,000 for conference room systems. Dollar sales for desktop units will reach about $1.6 billion versus a declining share for conference room systems of about $405 million.

These projections have been attractive incentives for vendors entering into the desktop conferencing market. Apple Corp., Connectix, Creative Labs, Insoft, Intel Corp., and PictureTel are six vendors which currently hold 80% of sales in this market. Companies also releasing video conferencing capabilities include Intel, Compaq, IBM, AST, Creative Labs, and US Robotics. Most vendor's desktop solutions range from less than $200 to $2000 based on different user profiles [Quick96], where DOS/Windows based packages accounted for 80% of sales in 1995 [EET96]. Clearly, the video conferencing market is in a major state of transition, fuelled primarily by the rapid advances in semiconductor technology. As networks become real-time capable, access bandwidths increase, and conferencing hardware becomes widespread, video conferencing environments will naturally migrate away from a group-oriented, room-based solution to a personalized, desktop-based environment.

2.3.3 Applications

There is little doubt that the demand for video conferencing environments will grow due to its wide application and inherent advantages. One of the first markets to fully embrace video conferencing will be that of distance education or telelearning, which has equal applicability to institutional education and business training. Quality Dynamics Inc. predicts that by the year 2000, half of all corporate training will be delivered via technology. A separate study by the Gartner Group projects the demand for technology-based training rising 10% a year for the next two years to $12 billion. "Corporate America spends $50 billion a year on continuing education to improve their employees' skill sets and retrain them to deal with the rapid pace of change in the workplace", says the CEO of the Home Education Network [InfoWeek96].

Telelearning refers to an environment where students and instructors are distributed geographically, and are able to see and hear each other. Students have access to tools such as an overhead slide viewer, a shared whiteboard and other types of collaborative tools which attempt to simulate the typical face-to-face learning environment as much as possible. The high quality of audio and video in these environments is of extreme importance since students must have the illusion that they are actually immersed in the typical classroom setting. Students may also have other tools at their disposal such as email or messaging, access to an external database of learning materials, and functional tools for the exchange of data. Telelearning environments will increasingly employ desktop-based video conferencing applications due to their affordable and personalized nature, however this does not exclude the use of room-based systems for lecture purposes.

The exploitation of video conferencing environments as a substitute for business travel has also gained much corporate attention. The cost savings realized by employing video conferencing technology as a substitute for business travel (airfare and lodging) is significant. Companies can now increase the frequency and spontaneity of scheduled meetings, with the option of including additional persons in these meetings at little or no cost. The age old problem of costly network resources gradually disappears as corporations look to install one global networking solution based on ATM technology. This implies cost savings as ATM can service all of their future networking requirements for voice (telephone), video (conferencing) and data.

Other applications of video conferencing technology exist, however they will only find widespread use once standard video conferencing technology is pervasive. These include pay-per-view VoD services, video kiosks in malls and restaurants, video editing and production tools, telemedicine applications and computer based training. Some of these applications involve only a uni-directional interaction between a user and multimedia repository, however they are very similar in their networking and processing needs to those of video conferencing applications.

Chapter 3: Requirements for Real-Time Multimedia

3.1 Motivation

In the realm of computer techno-jargon, *real-time* and *multimedia* are probably two of the most over used terms today. It seems that almost every new computer product and/or service marketed to the consumer professes at least one of these virtues. These terms have become synonymous with vibrant, colorful, and sensory appealing computer presentations which have the ability to allure the user into a state of absolute awe and amazement. As exciting as these presentations are, the user is still left with little understanding as to what these terms actually mean. The purpose of this chapter is to attempt to remove much of the ambiguity which surrounds these terms and to provide a solid set of requirements for what *real-time multimedia* implies, especially within the context of the ATM RendezView prototype.

3.2 Real-Time Systems

There is much debate surrounding the actual definition of the terms *real-time* and *real-time system* which is most likely due to the widely varying use of these terms. This chapter will concentrate on their application to computers and computer networks. Important factors which define a real-time system are *throughput*, *responsiveness*, *determinism* and *space* [Gallmeister95]. Throughput can be defined as the number of instructions that may be processed per unit time (in a computer context) or the quantity of data that may be transmitted per unit time (in a network context). Responsiveness refers to the speed at which a system reacts to an event and completes the necessary operations associated with that event. This is obviously context-dependant. Determinism refers to how much one can rely on a system to be responsive. With an ATM network, one may obtain a contract specifying a Quality of Service (QoS) provided by the network, however regular time sharing computers usually cannot offer such a guarantee. Space refers to non-time based resources typically required by a real-time system such as memory or hard drive space.

To classify a system as real-time requires the following steps. First, performance metrics must be chosen. Since no single metric applies to all applications, a set of metrics appropriate to the system must be selected. Second, metrics must be specified in terms of maximum, minimum and desired values, known as *performance requirements*. Once these metrics are fully specified, a system is designed and built to respect these requirements. Upon successful completion of the above steps, the system may be regarded as a real-time system. These general real-time concepts may be applied to a specific context such as multimedia in order to realize a *real-time multimedia system*.

3.3 Standards Development

In real-time multimedia, standards play a vital role in guaranteeing the successful development, deployment and interoperability of newly developed applications. This is due to the fact that real-time multimedia relies heavily on a wide array of supporting technologies, each of which has their own independent standards body (if one at all). In addition to these de-jure standards, de-facto standards also exist only further complicating matters.

A typical real-time multimedia application such as ATM RendezView requires supporting technologies such as control protocols, data/information formats, internetworking, an operating system, a delivery platform and a development environment [Minoli94]. It is no longer possible to create an application based on a proprietary implementation. The development task would become enormous due to all of the required technologies and would lead to yet another isolated, non-interoperable application.

Many standards bodies such as ITU-TS, ISO, ANSI, IEEE, ATM Forum, JPEG, MPEG and IETF have tasked themselves with developing standards for everything from encoding, to computing, to transmission standards. All of these efforts are great within their own rights, however the development of real-time multimedia applications requires some full scale standards in order to guarantee cross-platform compatibility. As an example, MJPEG is a standard for compressing and decompressing digitized video streams. However, the MJPEG standard doesn't say anything about the transmission format of MJPEG data. Therefore, if two vendors exploit the MJPEG standard for video encoding, there will still be no interoperability between their respective video streams. This removes much of the advantage of exploiting a standard in the first place.

Essentially what is required is some standardization efforts in the arena of a *multimedia architecture*. If such an architecture existed, developers could confidently produce multimedia applications which would be guaranteed to interoperate with another vendor's software. This is the purpose of the Interactive Multimedia Applications (IMA) standards group. The IMA was formed in June 1991 with the mission of reducing barriers to the widespread application of multimedia technology. It is composed primarily of hardware manufacturers, software developers and multimedia service providers. The Architecture Technical Working Group was formed which produced the Multimedia Layered Architecture specification. This specification attempted to define the architectural details of a multimedia system, from the user interface down to the devices and physical media.

The concept was novel, however the IMA realized that compatibility could only be achieved if agreement could be reached on all of the technical details involved in such an architecture. Because industry was, and still is, in a state of development in various arenas such as video compression, the IMA took the approach of specifying only a small subset of the component technologies rather than a single complete architectural choice, and made the specified set a minimum capability [Minoli94]. This allows vendors to add additional schemes as they deem appropriate while maintaining maximum compatibility.

Standards for real-time multimedia applications are an absolute requirement, however they are extremely difficult and time consuming to produce. Ultimately, a full scale architectural standard is required similar to that proposed by the IMA. In the meantime, developers should be cognizant of the many standards available, and prepare to support them or propose new ones in order for cross-platform interoperability to prevail.

3.4 System Performance

System performance is a crucial requirement for any real-time multimedia application, and the most difficult requirement to achieve. When referring to performance, we are talking in an end-to-end fashion which encompasses all the components of the real-time multimedia system. As they say, a chain is only as strong as its weakest link. This places a constraint on every component from the graphical interface down to the supporting network to be equally performant and to guarantee a certain quality of service.

Performance uses varying metrics to specify these constraints depending upon the context. Network equipment for example, could be required to guarantee a minimum transmission bandwidth, a maximum delay or jitter, and a maximum bit error rate. Software or hardware based CODECs could be required to perform based on data-dependant metrics such as frame rates, resolution and color depth. Every component within the system has its own unique set of metrics specified to describe its performance, yet little relation exists between all of these sets of metrics.

This becomes a problem for the real-time multimedia system designer since he/she must specify individual component performance requirements based on some overall perceived performance requirement. Assume that the ATM RendezView designers wish to provide the necessary inter-stream and intra-stream synchronization of audio and video media in order to provide for a realistic video conferencing environment in which users may be immersed. Table 3-1 [Steinmetz96] lists some synchronization performance constraints which must be respected from the user's perspective in order that such an environment be perceived as realistic and akin to a regular conversation. How should the designers map these synchronization requirements to a performance requirement for an underlying video CODEC for example? The problem becomes much more complex as components such as computer workstations are introduced which are unable to provide deterministic service guarantees (QoS) to a particular task.

Table 3-1 Quality of Service for Synchronization Purposes

Media		**Mode, Application**	**Quality of Service**
Video	Animation	Correlated	± 120 ms
	Audio	Lip Synchronization	± 80 ms
	Image	Overlay	± 240 ms
		Non-Overlay	± 500 ms
	Text	Overlay	± 240 ms
		Non-Overlay	± 500 ms
Audio	Animation	Event Correlation (e.g. dancing)	± 80 ms
	Audio	Tightly Coupled (stereo)	± 11 µs
		Loosely Coupled (e.g. dialogue mode with various participants)	± 120 ms
		Loosely Coupled (e.g. background music)	± 500 ms
	Image	Tightly Coupled (e.g. music with notes)	± 5 ms
		Loosely Coupled (e.g. slide show)	± 500 ms
	Text	Text Annotation	± 240 ms
	Pointer	Audio related to pointed item.	[-500, +750] ms

Frequently, a "brute-force" approach is adopted in order to resolve this problem of mapping higher level performance requirements into low level component requirements. This typically involves the use of excessively performant hardware modules coupled with software modules which are designed to be fast, lightweight, efficient and scalable. In a trial-and-error fashion, components are tuned such that overall performance requirements are met. This unorthodox approach is often the only solution available when using generic hardware and software components employed today. The development and implementation of real-time multimedia processing standards for end-systems and networks is one possible solution, however few of these exist today. Ultimately, a lot of experience, knowledge and luck may be the best assets possessed by a designer tasked with producing a real-time multimedia system.

3.5 Network Transport

By their nature, real-time systems place quite stringent demands on network resources. This is due to the need for timely and high bandwidth connection services between multiple end-systems. When considering networked real-time multimedia systems, several key requirements must be realized:

- *Multimedia applications tend to be bandwidth hungry.* Consider an uncompressed high quality video stream which could typically consume 270 Mbps of bandwidth. Compression algorithms can reduce this figure, however bandwidth requirements remain substantially elevated relative to conventional data streams.
- *Multimedia applications have QoS requirements necessary for realistic presentations.* These types of requirements (Table 3-1) ultimately get mapped down to the network level and take the form of figures such as bandwidth, error rates, latencies, and delay jitter.
- *Multimedia applications require fast, dynamic connectivity.* Whether an application requests point-to-point, point-to-multipoint, or broadcast connections, these should be negotiated and established dynamically in an efficient fashion. Users should not be subject to long call setup delays when requesting new real-time multimedia services.
- *Multimedia applications tend to require additional network services.* Such a service could take the form of guaranteed multipoint communications. This would necessitate multipoint connections or multicasting, and a multipoint transport protocol for robustness.

Since real-time systems place stringent requirements on network resources, the network must be capable of negotiating a service contract stipulating the characteristics of the connection required, and able to honor this contract throughout the duration of the connection. This service contract is commonly referred to as the Quality of Service (QoS) of a connection.

Because real-time multimedia is a relatively new field, few networking options exist which offer any sort of QoS guarantee. Most of today's technologies cannot offer the necessary bandwidths, nor can they provide predictable end-to-end delays. This situation stems from the fact that most of today's network technologies were designed previously for computer data transmission. Real-time network transmission requirements have surfaced only recently and thus have not been fully addressed. Table 3-2 highlights some of the fundamental differences between computer data and real-time multimedia stream transmission.

Table 3-2 Information Transmission Requirements

Requirement	Computer Data	Real-Time Multimedia
Bandwidth	***Low to Moderate:*** data transmission in large bursts, sporadic usage	***High:*** continuous flow of bursty or uniform data streams
End-to-End Delay (Latency)	***Not Critical:*** no implication as long as data arrives within a reasonable time frame	***Critical:*** application dependant, however generally must be tightly constrained
Delay Variation (Jitter)	***Not Critical:*** no real implication on successful data transmission	***Critical:*** most media types especially audio are highly sensitive to jitter
Error Correction	***Critical:*** data transmissions must be error free	***Undesirable:*** occasional loss is acceptable, however large delays incurred due to error correction and retransmission mechanisms are detrimental
Distribution	***Point-to-Point:*** most data transmission is client/server based	***Varied:*** applications may require pt-to-pt, pt-to-mpt, mpt-to-mpt or mpt-to-pt data transmission depending on application

Many solutions have been proposed in order to resolve this problem, however one known as Asynchronous Transfer Mode (ATM) is the most promising. In brief, ATM is a relatively young networking technology designed with real-time multimedia in mind. It is unique in that it offers the ability to transfer all types of data in a fashion which individually addresses the unique requirements of the data type being transferred. It offers all of the desirable features mentioned above including virtually unlimited bandwidth, a QoS mechanism and dynamic connection establishment. Chapter 4 (*Asynchronous Transfer Mode)* discusses ATM in detail as well as its suitability to real-time multimedia.

By their nature, real-time multimedia applications place very stringent performance requirements on their supporting network infrastructure. It is crucial that this infrastructure be capable of meeting these requirements in addition to providing complementary services to these applications.

3.6 End-System Architecture

Most real-time multimedia systems developed today employ some form of a computer workstation also known as the *end-system*. This is due to the rapidly increasing processing power exhibited by these workstations and the ease and speed with which prototyping may be achieved. Generally, the hardware implementation of a system will usually outperform its software based counterpart, however newer end-systems are progressively narrowing this gap. The significant cost advantage of a software implementation makes the computer workstation an integral tool for any real-time multimedia prototyping.

3.6.1 Hardware

When developing real-time systems, especially on end-systems which are inherently non-deterministic, supporting hardware can be a key component to meeting real-time performance deadlines. Even though they can be much more efficient than their software counterparts, specialized hardware add-on cards are usually expensive and so their merits should be thoroughly assessed. The key is to identify the processor intensive tasks of the system as well as those with the most stringent real-time constraints. In ATM RendezView as with most real-time multimedia systems, these would be:

- Video image compression and decompression
- Digital audio mixing
- Media stream I/O
- Network protocol processing
- Memory copies

From this list, video compression and decompression is by far the most processor-intensive due to the sheer size of video images. Digital audio mixing on the other hand has the highest priority for meeting its real-time deadlines due to the human ear's sensitivity to delay and jitter. The remaining tasks are not particularly demanding in either of these areas and may be adequately serviced in software.

From this assessment, we could choose to implement the audio and video functions using third party hardware cards, however there is an additional issue. Any functionality which is implemented in hardware may incur additional (potentially substantial) delays due to the extra memory copying required to move data to the hardware. It should be noted that memory copies within a workstation require non-trivial CPU time. A hardware-based CODEC, for example, would incur two additional memory copies to move data to the CODEC and back. Therefore, the CODEC must be substantially more performant than an equivalent software implementation for any benefits to be gained. This may not be a problem for a video CODEC due to the sheer complexity of compression/decompression algorithms, however audio mixing is relatively lightweight in comparison and may not be of any benefit. One way to circumvent this problem is to employ hardware which performs multiple functions. For example, the video hardware may not only perform CODEC functions, but may also be the low level driver for the video display. This would be ideal since it removes any overhead from memory copying (since a software based CODEC would still require one memory transfer to the supporting display hardware) and reaps the rewards of highly optimized CODEC hardware.

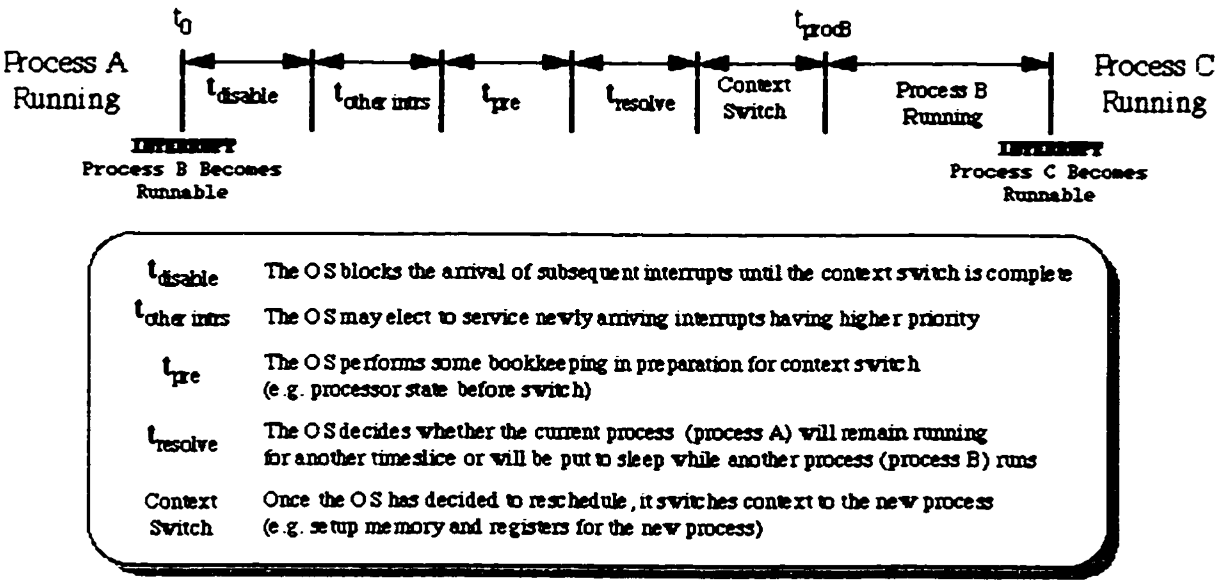
Hardware support within an end-system is an extremely powerful alternative, which if employed appropriately can be an invaluable means of meeting real-time deadlines within an inherently non-deterministic environment.

3.6.2 Operating System

The ideal multimedia Operating System (OS) would be one which offers the ability to negotiate a processing QoS contract for each process or task in much the same way network resources are allocated. However, few of these Real-Time OSs (RT-OS) exist today and even fewer, if any, have actually received any widespread use. Most workstations today employ *time sharing* based operating systems. The basic concept of the Time Sharing OS (TS-OS) is that a scheduling entity dynamically assigns a global scheduling priority to each process. The process with the highest priority gains access to the CPU. The scheduler alters the priorities of each process "on-the-fly" in order to balance the needs of interactive or I/O-bound tasks (requiring quick response) and the needs of compute-bound (processor intensive) tasks [Gallmeister95]. Ultimately, each and every process should acquire the CPU such that its operation proceeds in a reasonably timely fashion based on the number of contending processes.

This sort of scheduling methodology fairs well when non-real-time processes are competing for CPU resources, however it fails miserably for real-time tasks, where latency and jitter in processor scheduling causes real-time deadlines to be missed. In addition to the fact that many processes are competing for the same processing resource, there are other latencies injected into the processing cycle which may cause a process to miss its real-time deadlines even if it acquires the CPU immediately upon request. This is shown in detail in Figure 3-1 [Gallmeister95].

Figure 3-1: Latencies Inherent in OS Processing Cycle



Assume that process A is running on the CPU. Sometime on or immediately before time t_0, process B becomes runnable and waits to acquires the CPU. At time t_0, a hardware interrupt occurs indicating that a *timeslice* has expired and the OS should reassess its scheduling decision. Process B is chosen by the scheduler to acquire the CPU, however before it actually begins to use the CPU, it must wait t_{procB} seconds (*dispatch latency*) which corresponds to the overhead required to remove process A and instate process B as the active running process. It is important to note that the dispatch latency is non-deterministic and can grow large enough to cause process B's real-time deadlines to be missed. Add to this the fact that process B may not even acquire the CPU immediately upon request due to the scheduler's decision and may have to wait for additional timeslices to expire. Alternatively, even if process B does acquire the CPU, it may loose the processor before it has completed its necessary quantum of work, necessitating it to contend with other processes to reacquire the CPU. From these scenario descriptions, it is clear why a RT-OS is required. The lack of deterministic responsiveness inherent in TS-OSs is unsuitable for multimedia data, especially audio which is highly sensitive to delay jitter. In most end-systems today, real-time multimedia applications are not hindered by inadequate or non-performant CPUs, but merely the inefficient allocation of work to that CPU.

One solution which seems to be becoming more and more popular is the addition of a real-time scheduling class to the standard time sharing scheduler. Processes within this class are assigned global scheduling priorities which are not altered by the scheduler. Therefore, a process may be given a high enough priority such that it preempts any other process from the CPU as soon as it becomes runnable. This assures that a process will always acquire the CPU when required (subject to context switch latencies). This is a simple yet "brute force" solution to the above problem, however it does have its drawbacks. First off, this sort of scheduling is by no means truly real-time since a process may monopolize the CPU, thus causing other processes to miss their deadlines. There is no QoS contract negotiated with the scheduler on a per process basis, and so this configuration would not work with CPU intensive real-time processes. However, assuming that all processes in the real-time class are relatively lightweight, this configuration can be tuned such that they meet their real-time deadlines most of the time, and the remaining time sharing processes are not adversely affected.

Though various RT-OSs are available today, these would cause interoperability problems with conventional end-systems if they were employed. By adapting conventional TS-OSs to serve real-time processes as discussed, interoperability is maintained and performance is acceptable.

Ultimately, more research is required in the field RT-OSs to produce an OS which can service both real-time and time sharing applications efficiently.

3.6.3 Application Software

Application software, arguably more so than the OS and hardware, is the key to success in a real-time multimedia application. This component is (ultimately) responsible for providing the fine balance between executing compute-intensive, time-sensitive operations and the requirement to be lightweight and efficient from a processing viewpoint. These characteristics seldom come hand in hand and so it is the software's responsibility to provide all of the necessary functionality while also considering appropriate trade-offs and compromises based on end-system characteristics. Numerous approaches are available to achieve this goal, however it takes an extremely experienced designer to achieve the fine balance between all available alternatives. Performed correctly, this can produce an impressive real-time multimedia application.

3.6.3.1 Exploiting Available Facilities

Especially when attempting to design a real-time multimedia application, the designer must be aware of the facilities available within a particular end-system for real-time application support. This is not a simple task since the designer must first know what facilities would be desirable and then how to determine whether the end-system in question possesses these. Most end-system vendors do not, as of yet, openly advertise the existence of these real-time facilities, let alone their performance characteristics.

As mentioned above, a real-time operating system would be ideal, however this leads to interoperability issues due to the lack of any one globally accepted solution. Support for real-time processing and scheduling of application processes is currently the best compromise. End-system support for MultiThreaded (MT) programming is a close second. Multithreading offers many advantages such as the ability to split one large task into several smaller, more efficient ones for increased efficiency and utilization of the CPU. Threads are less resource-intensive than processes, provide a much simpler form of Inter-Process Communication (IPC), and lend well to the scheduling needs of real-time applications. The POSIX standard which has been widely embraced by the computing community defines a number of real-time facilities in addition to threads which many end-system and OS vendors are striving to support. Please refer to Section 5.10.3.1, "POSIX.4," on page 92.

End-system support for development tools can be invaluable as well. There are many coding languages employed today, each of which has its niche. Fortran, for example, is the fastest language for anything numerically intensive especially when running on RISC based end-systems, however it is not fully featured. Assembler is the most native language to a machine and hence the fastest. However, in addition to being machine dependant, the generated bulk and complexity of code makes it unmanageable especially when debugging is required. C and C++ are the most popular languages used today since they provide the greatest functionality coupled with good performance. C++ is more appropriate for applications which are inherently object-oriented such as multimedia, however coding time and performance may be inferior to C. Many software projects tend to incorporate multiple languages in the case where specific benefits are required from each.

In addition to a wide selection of coding languages, debugging and test tools should be available. The end-system should also optionally support an optimizing compiler to take advantage of the machine's specific architecture. While reducing the size of executing code, such a compiler should provide maximized pipelining assuming a RISC architecture, maximum parallelism on a multiprocessor end-system, memory and I/O optimization and efficient use of dynamically linked shared libraries.

Many facilities exist within end-systems today which can be invaluable in the development of real-time applications. As such, they should not be overlooked but exploited to their maximum potential.

3.6.3.2 In-depth Design Phase

The importance of a solid and in-depth design cannot be overstated, especially when it comes to real-time multimedia applications. Given an improperly designed application, no amount of expertise, optimization or tuning will enable real-time deadlines to be met! After developing a list of application performance requirements based on the real-time requirements, various designs should be drafted to determine the optimal approach to realize the real-time multimedia application. This is an extremely complex and time consuming process since it requires the designer to have a complete understanding of all of the available facilities and tools, in addition to the concept that trade-offs can be made without significantly compromising application performance.

As a starting point, the designer should determine the exact functionality required from each process or task. For each process, determine the functions which will be the most resource-intensive and decide whether these will prevent real-time deadlines from being met. The actual implementation should be designed in such a fashion that resource-intensive operations are optimized as much as possible. For example, most video applications are required to transfer bulky video data from one device to another after manipulating the data in some fashion.This is very CPU intensive and can cause other processes to miss their deadlines. Therefore, a design could optimize the necessary memory copying, perform memory manipulation in a more efficient language, or even employ an alternative such as Direct Memory Access (DMA) if available. These sorts of issues are highly dependant upon the needs of the application and therefore must be treated individually. Generally, the above process can be extremely complex since multiple processes can adversely affect each other in ways which are not initially evident on a microscopic scale. A more realistic approach may proceed as follows:

- *Perform the initial functionality assessment.* As discussed above, determine any potential performance problems and resource-intensive operations which may cause problems.
- *Design code appropriately.* Keeping the above issues in mind, the required functionality of the application, the need to interface with hardware and the various tools and coding facilities available, develop a first draft of the application. Code should be optimized but not excessively so.
- *Employ a lightweight and extensible approach.* The initial versions of the prototype should not be fully featured by any means. They should be extremely lightweight and designed in a modular fashion such that functionality may be added or removed as required.

- *Test the prototype to determine initial design issues.* Run the prototype and assess issues such as CPU loading, resource utilization, need for additional functionality, redundant features etc. This step gives the designer an intimate understanding of what potential problems may arise, it may also indicate any flaws in the original design.
- *Progressively develop the prototype application.* In a progressive fashion, add or remove functionality from the application and continually re-test and re-assess its performance. Continue in this fashion until all necessary functionality has been added and all real-time deadlines are being met.

Designing and developing in a progressive fashion is the most efficient approach. It is an effective method to determine which components of the application are contributing to resource load. and what functionality is additionally required or extraneous. Many developers feel compelled to develop an application with its complete functionality in one pass. This can cause some major problems which are unique to real-time development.

Assume separate audio and video streams are to be received from a network connection and then synchronized before being played out locally. An elaborate synchronization algorithm may be employed to absorb any jitter or skew which may have been introduced throughout the stream transmissions. Typically, multimedia intra-stream and inter-stream synchronization algorithms can be quite complex requiring many calculations and stream manipulation. If such an algorithm was implemented in a one-shot fashion and the program tested, it is quite feasible that the application would not meet any of its real-time deadlines. The designer would be at a loss. After all, this was the panacea of synchronization algorithms and it should not have failed. The reality of the situation could very well be that the extra computations and memory manipulations required by the synchronization algorithm were too resource intensive. In this case, the synchronization algorithm which was supposed to improve the perceived performance of the system actually caused it to fail! Unfortunately, most mathematically developed algorithms pay very little attention to the pragmatics of their feasibility in real-time systems. This is just one example of why it is important to develop real-time applications in a progressive fashion. Frequently, the designer realizes that certain components are redundant or unnecessary.

As a general guide to real-time multimedia development, keep the following points in mind. Applications should be efficient, lightweight and modular. Design and develop in a progressive fashion, keep real-time requirements in mind and do not over design. With multimedia, design for recoverability and robustness. Each media type has specific needs and applications should address these needs when implementing synchronization, recovery or processing algorithms. Always keep an eye on the big picture to assess the trade-offs of various design approaches.

3.6.3.3 Tuning and Optimizing

Once a high level language has been chosen, a working prototype has been produced and still the performance requirements of the application are not being met, the designer should consider tuning or optimizing the source code by hand. A wealth of techniques are available for squeezing the last few cycles out of a program's execution time. Refer to [Cockcroft95] and [Dowd93] for an in-depth approach.

Most optimizing compilers available today can perform much of the necessary optimizations directly, however knowing how to code a high level language efficiently can be beneficial. There are two fundamental reasons why optimizing code is necessary given the current class of end-systems available. First, current RISC processors incorporate pipelining and other parallel techniques to a much greater degree than previous CISC processors. Consequently, if code is structured such that the compiler is able to efficiently exploit the parallelism inherent in the RISC architecture, overall performance will be greatly improved. Second, processors have become much faster in the past decade, but memory speeds have not kept pace. Therefore, if code is accessing memory in a sporadic or non-linear fashion, the CPU is going to spend a lot of time in an idle state waiting for data, which is also undesirable.

An optimizing compiler attempts to streamline code as much as possible in order to make it run faster. As a first cut, this usually means simplifying the code, throwing out extraneous instructions, and sharing intermediate results between statements. More advanced optimizations seek to restructure the program, and may actually make the code grow in size, though the number of instructions executed will (hopefully) decrease [Dowd93]. The compiler must know some details about the machine architecture in order to generate machine language code. It must know about registers and rules for issuing instructions, the cost of each instruction and the latencies of machine resources such as the pipelines.

A compiler is obviously very complex and an invaluable tool in the application development process, however there are instances where a compiler is unable to detect optimizable code. Sometimes, this code can be restructured and optimized manually thus producing significant performance gains. These optimizations could take the form of eliminating clutter/overhead, exchanging macros for subroutines, streamlining branches, removing redundant tests or wordy conditionals, optimizing loops and memory references etc. This is a complete field of study on its own and usually adequately addressed by an optimizing compiler. However there is quite often some conceptual optimizations that the designer can make based on *a priori* knowledge of the code's functional requirements. For example, the designer could use a completely different, less resource-intensive video compression algorithm to achieve similar or identical perceived video quality from the decoded video stream. This type of optimization usually provides the greatest performance benefits of all available alternatives.

Tuning and optimizing code can provide significant performance advantages when designing real-time multimedia applications. Luckily, most of the "dirty work" is performed by optimizing compilers, however there are occasionally some design decisions which can be critical to the successful operation of the application. Further compiler-specific options should always be investigated such as early versus late binding and optimization levels/types in order to squeeze the maximum performance possible from the end-system.

3.6.4 Human/Machine Interface

The Human Machine Interface (HMI), also more widely known as the Graphical User Interface (GUI) is yet another integral component of any real-time multimedia application. Multimedia applications manage the capture and presentation of human sensory-oriented digital information, of which visual representations are an extremely important component. Interactive multimedia applications will be successful only if the HMI is carefully developed according to principles of graphic design and cognitive psychology [Minoli94]. In particular, models of user knowledge that include goals and plans in addition to facts, are needed for multimedia to fulfill its goal of improving the "bandwidth effectiveness" of user-to-computer and user-to-user interactions [Minoli94].

Research in the field of HMI is ongoing and extremely complex. There are however some very basic guidelines which can be used in order to produce an efficient interface:

- *The HMI must be performant.* If several live video streams are to be displayed in full color at 30 fps for example, the hardware supporting the physical display must be capable of supporting these requirements. Essentially, the HMI must be able to meet the real-time requirements of the multimedia data to be displayed.
- *The GUI must respect some basic design criteria.* At its most basic level, the GUI should be user centered, easy to use, intuitive, flexible, consistent and conform to some basic linguistic guidelines. [Bot93] and [OSF93] provide some additional detail in each of these areas, however many theoretical references exist on this subject as well.

As a general point, new users of a GUI should invest little or no time becoming familiar with it. If users are having problems navigating or exploiting the application's functionality, a redesign of the GUI should be considered. Ideally, some sort of style guide should be employed throughout the design phase. The importance of the HMI should never be overlooked throughout the multimedia application design process. Even if the application has been efficiently designed, tuned and optimized for performance, it can still fail to meet its real-time deadlines due to an inadequate HMI.

Chapter 4: Asynchronous Transfer Mode

4.1 Introduction

Asynchronous Transfer Mode (ATM) has been touted as the panacea of networking technology solutions available for real-time multimedia communications. This may in fact be true, however it is essential to understand ATM in detail and what makes it so apt for real-time communications. In section 3.5, several key requirements were highlighted as necessary components of any real-time network transport infrastructure. Briefly, these are: high bandwidth, guaranteed QoS, fast dynamic connectivity and supporting network services. This chapter endeavors to provide an introduction to ATM as well as its operation in the context of ATM RendezView. Much of the core ATM technology information was extracted from [Ahmad96], however this chapter is by no means a complete tutorial on ATM since many excellent references exist and should be consulted for completeness.

4.2 ATM Technology

4.2.1 Why ATM?

Fundamentally, Asynchronous Transfer Mode (ATM) was conceived and designed with real-time data transmission as a key motivation. Immediately, this implies the need for high bandwidth, low latency connections which ATM affords. This design characteristic alone makes it ideal for any application requiring real-time network facilities since no other network technology can make this lofty claim. In fact, this is the fundamental problem with most networking technologies available today: they cannot adequately service real-time data, although some fair better than others.

ATM was designed to be fast, flexible and efficient in its handling of time sensitive information. It has the unique ability to simultaneously service various types of data streams in a fashion which addresses each stream's individual needs and requirements. Whether streams take the form of the voices of two telephone callers, a movie being viewed in someone's living room or simply computer data being transmitted between two corporate offices, ATM has the unique ability to service each appropriately without any compromise in perceived quality. This is accomplished through various mechanisms which include Quality of Service (QoS) negotiation on a per connection basis to specify data specific requirements, and traffic policing to guarantee that contracts are not violated. Additional features which make ATM a favorable network solution include:

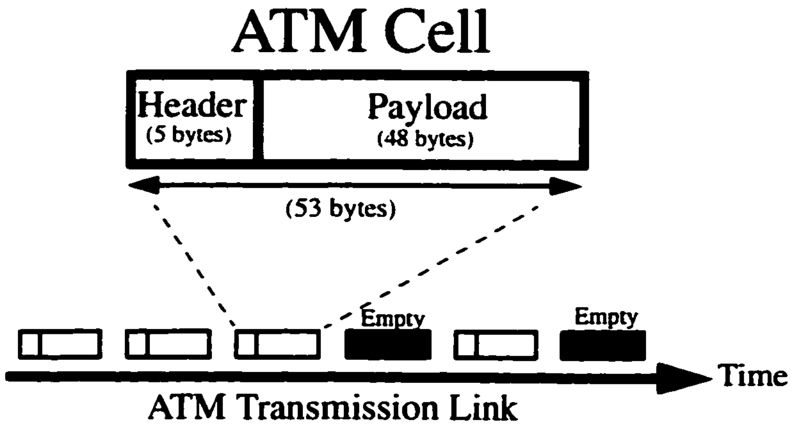
- *Single international standard for public carriers and private networks:* The ATM Forum is the primary international body charged with developing and standardizing ATM related standards. The ATM Forum which is composed of over 700 members worldwide including vendors, providers, R&D institutions and users, has created a unified common technology infrastructure across the LAN, MAN and WAN. ATM therefore not only bridges the gap between a wide array of digital media types, but also the private and public companies requiring seamless end-to-end communications and connectivity.
- *Potential for Integrated Services:* Once a fully functional ATM network infrastructure is in place, the possibility for new services becomes great. Some of these are video conferencing, video on demand, distance learning, work at home, medical imaging, remote design/graphics, scientific visualization and virtual reality. Legacy network traffic can be simply and efficiently mapped into ATM cells for compatibility.
- *Scalability and flexibility:* ATM offers no distance limitations on communications, in addition to gigabit switching capability and real-time multimedia application capability. This enables ATM to transport and service virtually any type of traffic in a point-to-point or multipoint topology.

Though ATM is still a relatively young technology, it has shown much promise. Test trials and demonstrations have shown the technology to be able to provide all of the features highlighted above. Only time is required for standards to mature, wide spread deployment to begin and ATM technology to demonstrate its full potential as the only required networking technology of the future.

4.2.2 The Atomic Unit - The Cell

The success of ATM hinged on the development of a high speed, flexible and efficient means of data communications. The original designers looked to legacy systems such as Time Division Multiplexing (TDM) and packet networks to determine the benefits of each. From the TDM world, they retained the concept of fixed length packets and slot synchronism. From the variable length packet networks, they retained label multiplexing for its flexibility and efficiency. This produced a specific packet oriented transfer mode which uses asynchronous time division multiplexing. The packet flow is organized into fixed length blocks called cells which are 53 octets in length. A cell consists of a 5 byte header and a 48 byte payload as shown in Figure 4-1.

Figure 4-1: ATM Cell Format



ATM is connection oriented which implies that an end-to-end Virtual Channel Connection (VCC) is established initially before any data transmission occurs. At call setup time, information transfer capacity is assigned to the VCC by negotiation with the network based on source requirements and available network capacity. Cells belonging to a VCC (based on an identifier found in the cell header) receive service according to the traffic contract associated with the VCC. Cell sequence integrity on a VCC is maintained in an end-to-end fashion however no payload error correction is implemented.

ATM achieves its high efficiency and speed by keeping cells fixed in length and by exploiting the benefits of statistical multiplexing. Network switches are only responsible for inspecting a cell's header and switching it based on its VCC. This functionality may be performed efficiently in hardware, without the need for slower software-based routing functions as required by legacy technologies. This reduces the necessary processing at switching points and affords highly streamlined and high speed hardware implementations. Because cells are short, they may be efficiently statistically multiplexed with cells of other VCCs without incurring significant jitter. This is highly suitable to real-time applications where latency is a primary concern.

4.2.3 The Road Map - The Cell Header

The primary function of the ATM cell header is to provide routing information. As shown in Figure 4-2, it also has fields for implementing flow control, indicating a loss priority, the payload type being carried as well as a checksum code to provide cell header integrity.

Figure 4-2: ATM Cell Header Format at the UNI and NNI

User Network Interface (UNI)

<table>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>Bit / Octet</td></tr>
<tr><td colspan="4">GFC</td><td colspan="4">VPI</td><td>1</td></tr>
<tr><td colspan="4">VPI</td><td colspan="4">VCI</td><td>2</td></tr>
<tr><td colspan="8">VCI</td><td>3</td></tr>
<tr><td colspan="4">VCI</td><td colspan="3">PT</td><td>CLP</td><td>4</td></tr>
<tr><td colspan="8">HEC</td><td>5</td></tr>
</table>

Network Node Interface (NNI)

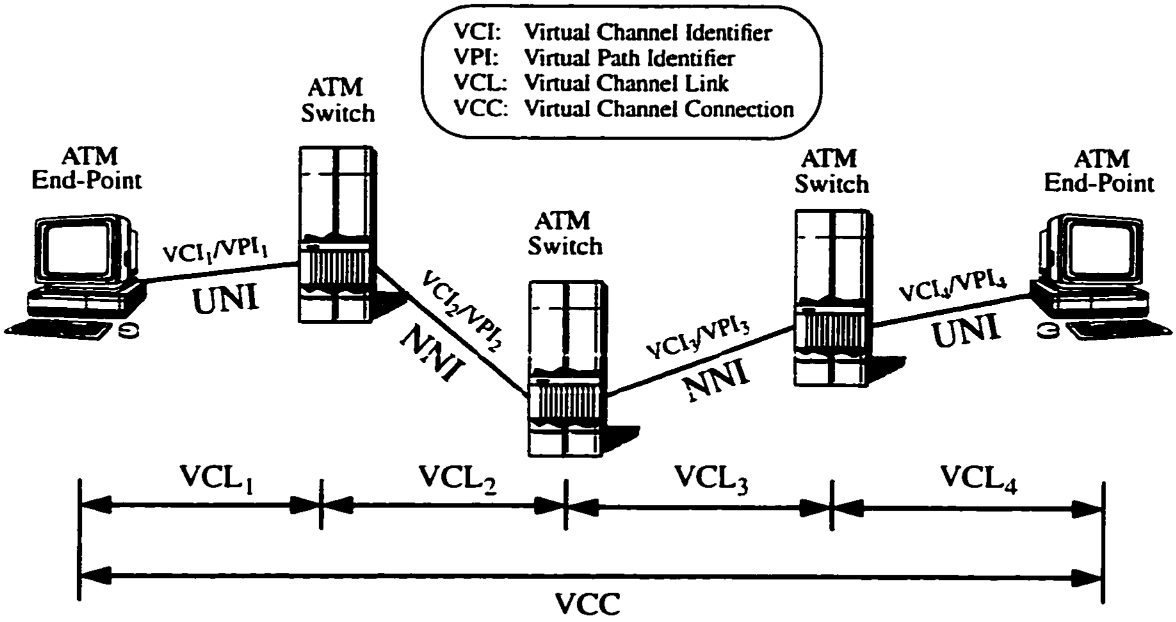
<table>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>Bit / Octet</td></tr>
<tr><td colspan="8">VPI</td><td>1</td></tr>
<tr><td colspan="4">VPI</td><td colspan="4">VCI</td><td>2</td></tr>
<tr><td colspan="8">VCI</td><td>3</td></tr>
<tr><td colspan="4">VCI</td><td colspan="3">PT</td><td>CLP</td><td>4</td></tr>
<tr><td colspan="8">HEC</td><td>5</td></tr>
</table>

GFC: Generic Flow Control
VPI: Virtual Path Identifier
VCI: Virtual Channel Identifier
PT: Payload Type
CLP: Cell Loss Priority
HEC: Header Error Control

Routing of ATM cells is achieved by designating the cell as belonging to a particular Virtual Channel Connection (VCC). A VCC is an end-to-end connection route which is setup in the network before data transmission can begin and has an associated QoS contract which is monitored and enforced by the network. A VCC is composed of a concatenation of Virtual

Channel Links (VCL) which exist on a per link basis as shown in Figure 4-3. A VCL is identified by a unique VCI/VPI pair and so has only local significance since it is usually mapped to a new VCL value as it passes a switch point. ATM makes a distinction between the interface between an end-point (user) and the network, termed the User Network Interface (UNI) and the interface between two network entities, termed the Network Node Interface (NNI).

Figure 4-3: Format of a Virtual Channel Connection (VCC)



The remaining fields within the ATM cell header have the following functions. The Generic Flow Control (GFC) field is used at the UNI to control ATM cell flow. It allows feedback to the sending end-point to indicate congestion has occurred within the network and so data transmission rates should be throttled back. The Payload Type (PT) field functions as a congestion indicator and also distinguishes user cells from OAM or resource management cells. The Cell Loss Priority (CLP) bit is used to indicate the cell's relative discard priority in the case that the network becomes congested and must throw cells away. The Header Error Control (HEC) field employs a CRC calculation to enable ATM cell header error detection and correction in addition to cell delineation.

Although the ATM cell header introduces a significant overhead to data transmission, it enables additional functionality within the network. Because of the cell's fixed length and simple processing needs, the speed and latency advantages significantly outweigh any lost bandwidth due to overhead.

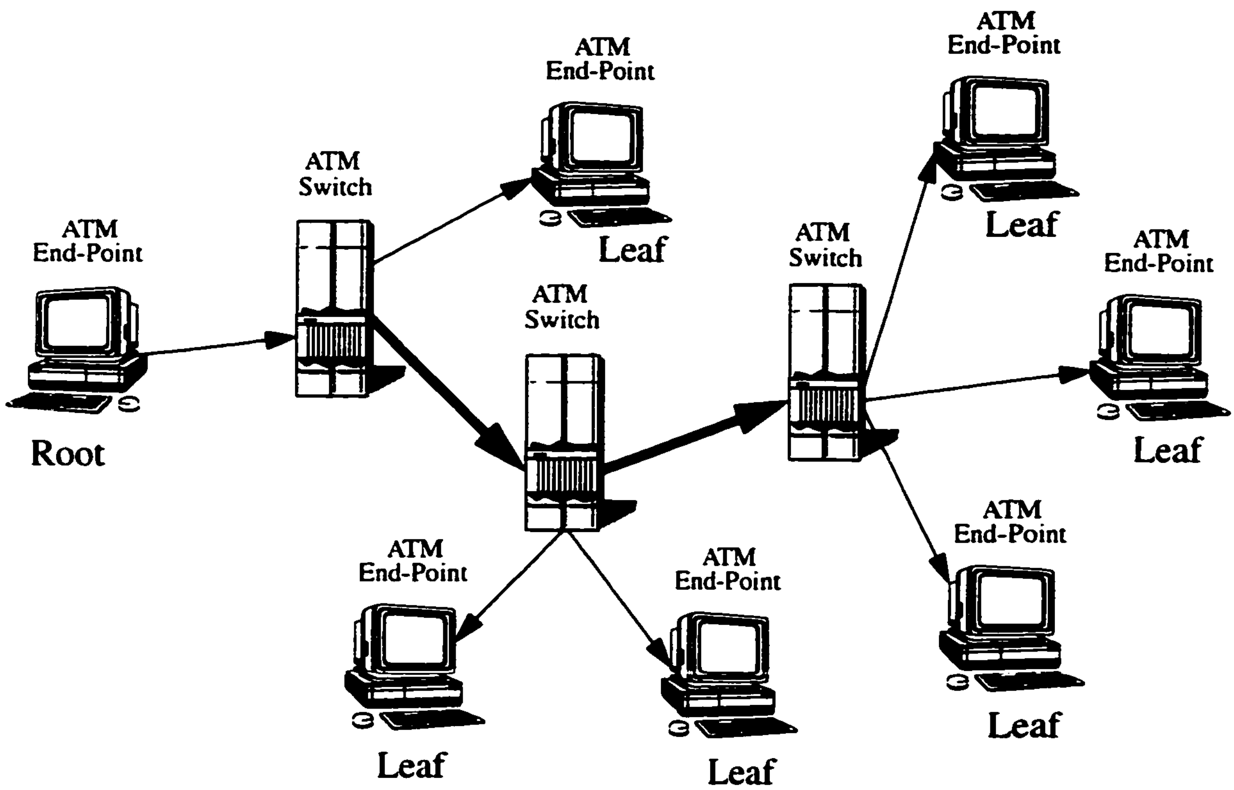
4.2.4 ATM Connections

In ATM, not every end-to-end ATM connection is created equal. A differentiation must be made between the means in which a connection is setup and the type of connection employed.

Connections may be setup either statically in advance, or dynamically as required. A Permanent Virtual Connection (PVC) is an end-to-end VCC which must be setup well in advance of its use, typically by a network administrator. These have an associated QoS and tend to be left "nailed-up" for extended periods of time due to their static nature. They may be used as backbone trunks between geographically separated LANs or for long term connectivity within a LAN. The dynamic connection approach employs Switched Virtual Connections (SVCs), which are VCCs which are "nailed-up" and "torn-down" dynamically on-demand. SVCs require the use of a signalling protocol such as the ATM Forum's UNI protocol. SVCs are the most popular since they offer the ability to dynamically establish a VCC between two end-points through a network without the need for any prior configuration. In a similar fashion to PVCs, SVCs also have a QoS contract associated with them which is negotiated at call setup time.

According to the UNI v4.0 standard, ATM connections can be point-to-point or point-to-multipoint. Point-to-point connections are simple uni-directional or bi-directional communications paths between two end-points. Point-to-multipoint on the other hand allows a uni-directional one-to-many communications path between multiple end-points as shown in Figure 4-4.

Figure 4-4: ATM Point-to-Multipoint Connections

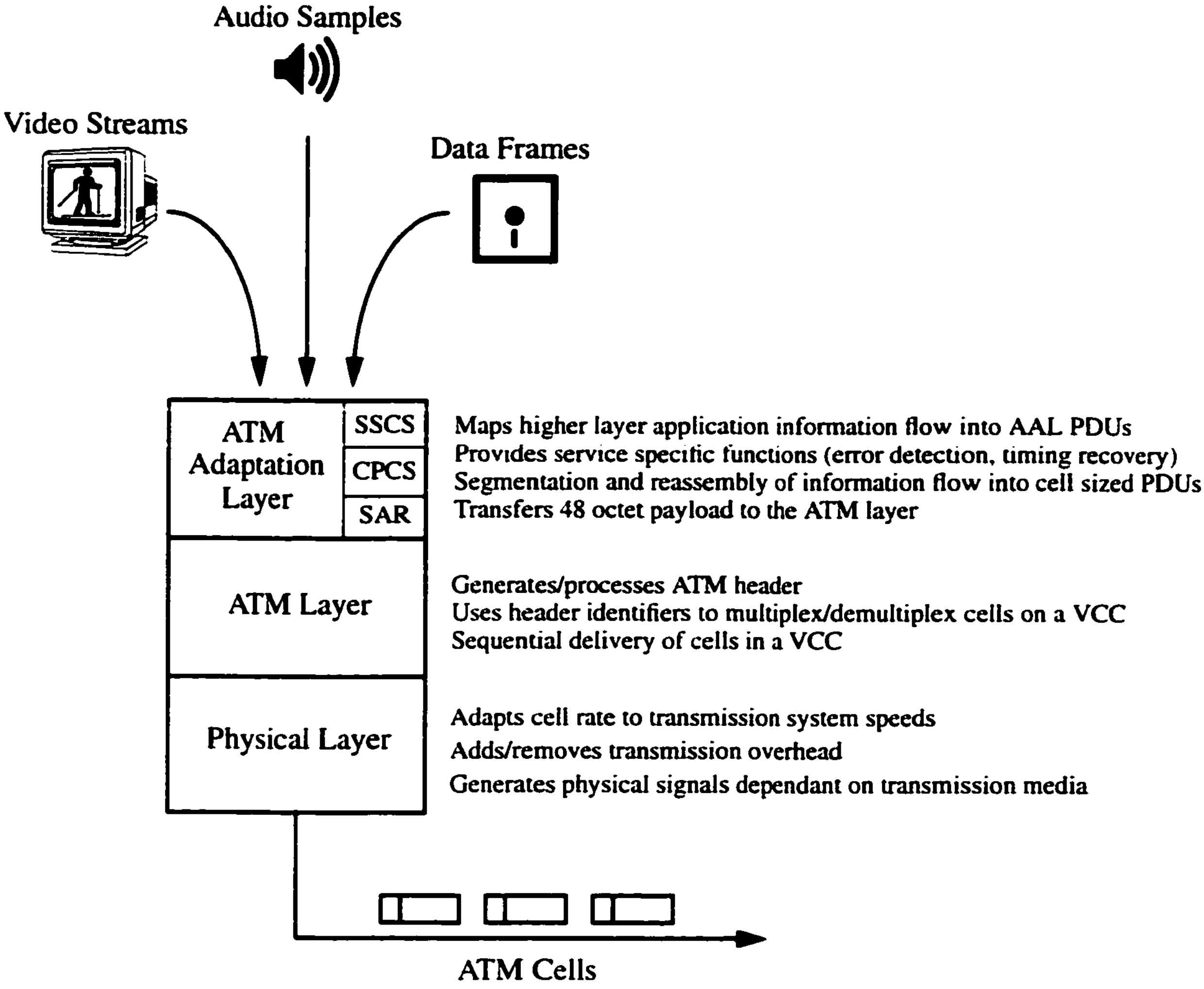


A tree analogy is used to describe the end-points in a point-to-multipoint connection. The sender is termed the root and all of the receivers are termed the leaves. Therefore the root transmits data on the connection and all leaves receive an exact copy of this data. ATM employs methods for allowing leaves to arbitrarily join a point-to-multipoint connection using a Leaf Initiated Join (LIJ) mechanism. Essentially, the end-point wishing to become a leaf of a point-to-multipoint connection signals their intention using UNI v4.0 at their UNI. The network responds by forwarding this request to the root end-point (if notification is required) and then takes the responsibility of duplicating network traffic at the appropriate points in the network such that the new leaf receives all data transmitted by the root. These types of connections are analogues to multicasting or broadcasting in the legacy shared medium networks. Future UNI standards released by the ATM Forum should support multipoint-to-point and multipoint-to-multipoint connection topologies, however point-to-multipoint is by far the most useful at this time.

4.2.5 ATM Protocol Stack

The ATM protocol stack is extremely simple in design which is one of its great selling points. Unlike the OSI seven layer stack, ATM has only three layers all of which can be implemented in hardware.

Figure 4-5: The ATM Protocol Stack



At the bottom of the stack is the physical layer. This layer is responsible for generating and receiving the physical signals on the transmission medium. Existing standards for the transmission medium specify copper wire, coaxial cable, fiber optics and even wireless systems. No physical limitation is placed on the transmission rates which have been standardized up to 622 Mbps. This layer is also responsible for adding/removing any transmission line overhead and adapting cell rates between differing transmission systems. The second lowest layer is the ATM layer. It is responsible for generating/processing the ATM cell header, and multiplexing/demultiplexing cells on a VCC. The ATM Adaptation Layer (AAL) is the most important of all three layers since it interfaces and adapts between arbitrary data units issued by higher layer applications and the cells issued by the ATM layer. The ATM Forum has defined four AAL types which reflect data specific transmission requirements such as constant or variable bit rate service, connection oriented or connectionless transfer, and the need for source and destination timing relationships. Details of the four AAL types are shown in Figure 4-6.

Figure 4-6: ATM Service Class Descriptions

Application Layer		Voice, Video Circuit Emulation	Packet Video	Data	
B-ISDN Bearer Service	Class	A	B	C	D
	Bit Rate	Constant	Variable		
	Connection Mode	Connection Oriented			Connectionless
	Source & Destination Timing Relationship	Required		Not Required	
ATM Bearer Service	AAL	Type 1	Type 2	Type 3/4	Type 5
	ATM Network	ATM Layer			
		Physical Layer			

The AAL layer typically involves three processing sublayers unique to each AAL type, not all of which need to be implemented. They are the Service Specific Convergence Sublayer (SSCS), the Common Part Convergence Sublayer (CPCS) and the Segmentation And Reassembly (SAR) sublayer. The SSCS processes higher layer data units directly and would be concerned with issues such as timestamp processing, data unit numbering/tagging, and message type identification. The CPCS is concerned with end-to-end frame integrity and ordering. The SAR sublayer is responsible for guaranteeing the proper segmentation of AAL data units into ATM cells and their subsequent reconstruction at the receiver.

Table 4-1 ATM Adaptation Layer Types

AAL Type	Main Intended Applications and Success
AAL Type 1	Circuit Emulation (T1, DS3), Voice on ATM, CBR Video (e.g. MPEG2) and any other CBR or delay sensitive services. Primarily employed in PSTNs
AAL Type 2	Variable Bit Rate (VBR) and delay sensitive services (e.g. VBR video transport). No progress or success, AAL2 standard is incomplete and preliminary without much support from industry.
AAL Type 3/4	VBR, delay insensitive services (e.g. data and file transfers) Commonly used for connectionless data services (e.g. SMDS)
AAL Type 5	VBR, delay insensitive data services. Specified by ATM Forum for video (MPEG2) video transport Most popular AAL for computer applications, lightweight and simple processing requirements.
AAL Type CU	New AAL (Composite User) under study in the ITU-T. Intended for compressed voice multiplexing (e.g. wireless/trunking).

The choice of AAL type for a specific type of data is not straightforward (see Table 4-1) and so much debate exists on which AAL is most suitable for a specific service. AAL 5 has proven to be the most popular recently in computing environments since it is extremely lightweight from a processing and overhead viewpoint and handles most traffic types in an adequate fashion. AAL 1 has gained popularity outside of the computing environment for use with CBR traffic such as TDM voice on public telephone networks. Future research is required to determine whether new AAL types are required or whether only specific services require non-generic AAL processing.

4.2.6 Traffic Management Mechanisms

An ATM network's ability to negotiate a traffic contract with a user at connection setup time and to respect and enforce this contract throughout the lifetime of the connection is one of its chief selling points. The primary objectives of traffic management are to provide the required QoS to a user, to maintain network performance objectives, and to optimize the use of network resources. Although ATM traffic control technology and standards are rapidly evolving, there is no single traffic control mechanism optimized for all ATM network applications. This section endeavors to highlight some of the facilities currently available for traffic management.

4.2.6.1 Connection Admission Control

At call setup time, before any data may be transmitted onto an end-to-end ATM connection, the sender must first negotiate a traffic contract with the network. Connection Admission Control (CAC) encompasses the actions taken by the network during the connection setup phase to establish whether a connection can be accommodated within the available network resources without QoS degradation. A traffic contract is first submitted by the ATM end-point which is in turn accepted or rejected by the network due to insufficient resources. Upon rejection, the network will propose a new contract which the sender may accept or reject. The process proceeds in this fashion until such time that the call is rejected by either party or a consensus is reached. A traffic contract may include any of the following components:

- *ATM Transfer Capability:* A transfer capability indicates a predefined data service that is representative of the source's data transfer in a broad sense. Possible options include Deterministic Bit Rate (DBR) (formerly CBR), Statistical Bit Rate (SBR) (formerly VBR), Available Bit Rate (ABR) and ATM Block Transfer (ABT). Not all of these services are completely defined, however they are representative of the transmission characteristics of most data types.
- *Source Traffic Descriptor:* The traffic descriptor is a set of traffic parameters, either qualitative or quantitative which can be used to describe the traffic characteristics of the required ATM connection. Typical parameters could be Peak Cell Rate (PCR), Cell Delay Variation (CDV), Sustained Cell Rate (SCR) or burst tolerance.
- *Quality of Service Class:* The QoS class could be either a locally or globally established class which maps into a particular set of traffic descriptors. For example, a class "MPEG-2 Video @ 2 Mbps" could be defined which would map to the proper traffic descriptors and transfer capabilities. This simplifies the user's task of specifying appropriate QoS parameters which can be a very complex process.

- *Option for Tagging Cells:* This option would place the emphasis on the sender to differentiate low priority and high priority cells upon transfer. In the case that network congestion occurs, lower priority cells could be thrown out so that real-time deadlines are not violated for higher priority cells. This option is ideal for data which has inherent priorities such as MPEG video which has three frame types each contributing a varying degree of video frame continuity.

4.2.6.2 Usage/Network Parameter Control

Usage/Network Parameter Control (UPC/NPC) involves the set of actions taken by the network to monitor and control the traffic transmitted by the sender at the UNI and the NNI. The actions taken to control any violation of a QoS contract can vary from passive (in the case that additional network resources exist to accommodate the violation) to active (where user data is discarded so as not to adversely affect other data streams within the network). This is ultimately a dynamic decision based on network state.

4.2.6.3 Priority Control

A simplistic approach to controlling network congestion is to selectively discard network traffic that has been flagged as low priority. This may also be implemented as a UPC/NPC resolution mechanism. Cell priorities are indicated via the cell loss priority bit within the ATM cell header.

4.2.6.4 Feedback Controls

Feedback mechanisms allow for a sender of data onto an ATM connection to be notified if the network becomes congested or when QoS contracts are being violated. This enables a more proactive approach on the sender's side to control data flow. This could, for example, enable a sender to reduce the frame rate or spacial resolution of a transmitted video stream in order to decrease transmitted bandwidth if network congestion occurs. A less favorable alternative would entail the network to arbitrarily discard video frames which could cause major degradation in perceived video quality at the receiver.

Traffic management is an extremely complex field which still requires much study. It does however provide an excellent alternative to the legacy traffic management approaches which are primarily a "best effort" approach with little or no feedback to the sender. Once ATM's traffic management mechanisms are fully developed and implemented, it will become the ideal solution for the transport of real-time information.

4.3 ATM is Superior to Legacy Networks for Real-Time Multimedia

One of the biggest hurdles for real-time multimedia is to find a networking solution which can respect its time critical deadlines (Please refer to Section 3.5, "Network Transport," on page 34.). Most legacy networks such as ethernet and token ring utilize the *de facto* TCP/IP suite of data protocols. Unfortunately, these were designed for data transfer and do not work well for real-time data (however much effort is being made to change this). (Please refer to Table 3-2, "Information Transmission Requirements," on page 35.). In addition, these networks do not offer the necessary bandwidth or quality of service required by real-time multimedia. Probably the biggest problem is the sheer processing power required to implement these protocols given the elevated bandwidth of typical multimedia data such as broadcast quality video. Therefore, by removing these protocols we gain both in reduced transmission overhead and reduced processing requirements imposed on the end-points. This in itself creates a lightweight environment which is more suited to real-time data processing.

Multimedia real-time data requires strict performance guarantees such as end-to-end latency, transmission delay jitter, bandwidth and error rates. This implies that the physical network must be able to provide a QoS guarantee similar to that provided by ATM. Legacy systems do not typically afford these features.

ATM imposes essentially no limit on transmission bandwidth in addition to a negotiable QoS contract. Because of the available bandwidth, processing loads on end-systems are reduced since compression algorithms need not be as efficient and processor intensive. Since a QoS contract exists and ATM networks have inherently low error rates, the need for higher layer transmission protocols is reduced which again offers a performance gain. In ATM, each VCC has its own QoS contract and so different types of time critical data can receive the service and attention needed without wasting network resources.

The dynamic SVCs offered by ATM in combination with multicast connections offer a great advantage to the end user. Not only is stream duplication and distribution handled by the network, thereby off-loading more processing from the end-system, but *switched presence* and *layered algorithms* may be exploited. Switched presence refers to the scenario where a multiparty video conference participant is able to actively view the current speaker only throughout the duration of a video conference. This is achieved by dynamically joining and leaving multicast SVCs at the network level. This is especially useful if a conferee's network access bandwidth is low or their end-system is not performant enough to process multiple media streams at one time. Layered algorithms could apply to media streams such as HDTV which may be encoded such that they are transported on multiple ATM connections. Connecting to more of these connections enables a user to obtain an increasingly higher quality video image up to the full HDTV resolution. Not all users may have the video display terminal or network access bandwidth necessary to view full HDTV resolution.

The ATM protocol stack is lightweight and can be implemented completely in hardware with a significant performance advantage. This offers the capability for very high bandwidth connections in addition to scalability and extensibility unique to ATM. New services may be easily deployed to a large number of users via the quality of service and multicast facilities offered by ATM.

Chapter 5: ATM RendezView Prototype

5.1 Motivation

Fundamentally, the greatest motivation for developing the ATM RendezView prototype was determining whether a true real-time collaborative environment could be realized using currently available technologies. Therefore, the success of the prototype relied heavily on its ability to establish high bandwidth end-to-end media stream connections which would exhibit low latency and low delay jitter characteristics. These characteristics would have to be maintained even when these connections were established between end-systems which were tasked with processing all of the incoming and outgoing media streams in a concurrent fashion.

The demonstration of ATM as a facility for delivering real-time multimedia streams to the desktop in a multipoint topology was another motivation. Because ATM possesses so many features which are well suited to real-time multimedia, such as low latency and jitter, dynamic multipoint SVCs and native transport to the desktop, it was desirable to see if the prototype could take advantage of these features. These results have the potential to provide a strong argument for further standardization and development efforts in the arena of ATM to-the-desktop technology.

A third motivation was to determine whether other involved technologies would interwork successfully so as not to detract from the advantages offered by the ATM network technology. Such technologies include the end-system's Operating System (OS), I/O efficiency and software mechanisms. Ultimately, if the end-system is not capable of processing the media streams in a timely fashion, there would be little reason to adopt ATM transport technology over the standard IP based transport networks.

5.2 System Functionality

As a prototype, ATM RendezView has the goal of being functional more so than fully-featured. Recall that a primary motivation of the prototype is to demonstrate the use of network-based and computer-based technologies to realize a real-time multipoint collaboration tool. Therefore, multipoint video conferencing was deemed to be an application which would demand real-time performance from the multimedia system and its components, thus allowing for a true assessment of the performance of the proposed real-time multimedia environment.

As a video conferencing tool, the ATM RendezView prototype offers the ability for a conferee to:

- Register to the conference environment

This enables the conferee to perform all of the necessary steps of logging into the local environment, entering a photo-booth to establish an identity and learning about scheduled conferences.

- Dynamically join/leave a particular conference

Once registered, the conferee is able to browse a dynamic listing of all of the currently established and/or pending conferences to which access has been granted. One such conference may be joined or left as desired.

- Learn about media streams and services offered by other conferees

Upon joining a particular conference, the conferee obtains a dynamic listing of available media streams such as audio and video being offered by other conferees.

- Advertise the media streams and services being offered to other conferees

In addition, the conferee chooses media streams which are to be offered to other conferees participating in the same conference

- Dynamically receive media streams and/or services on demand

Having selected particular media streams to receive, native ATM multicast network connections are negotiated and established dynamically to accept the selected media streams. Appropriate synchronization is implemented based on the media stream types and respective sources.

If ATM RendezView were to be commercialized, it would require additional functionality to make it a truly distributed, collaborative work environment. Such functionality would include:

- Slide presentation tool

Enable a conferee to host an interactive slide presentation assisted by some basic tools such as a pointer and markup tool.

- Collaborative whiteboard tool

Enable multiple conferees to collaboratively work on a shared workspace to edit drawings/graphics and pool ideas.

- Shared application-independent work environment

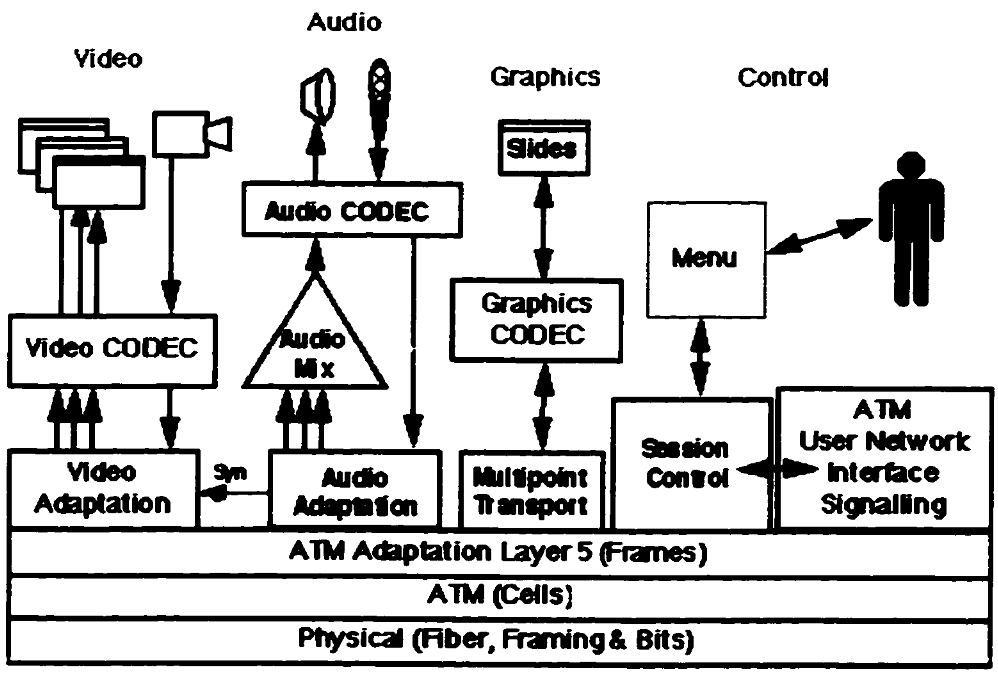
Enable a conferee to share an arbitrary third party application with all conference participants. Applications include CAD tools, publishing tools, information browsers etc. Application control and operation should be possible by any conferee in a non-contending fashion.

It should be emphasized that the framework employed by ATM RendezView is not unique to desktop video conferencing, and may be employed to provide other types of services. One specific example to illustrate is Video on Demand (VoD). ATM RendezView serves more as a motivation to demonstrate what is possible given the new ATM networking technologies and standards which are rapidly evolving.

5.3 System Architecture

In order to create a prototype application which could effectively promote the advantages of ATM technology, ATM RendezView was designed to include audio/video functionality along with a slide presentation tool all utilizing native ATM technology such as point-to-multipoint (multicast) connections and Switched Virtual Circuits (SVCs). A high level diagram of the ideal ATM RendezView architecture is shown in Figure 5-1.

Figure 5-1: ATM RendezView Application Architecture



Compared to the ideal architecture, the ATM RendezView prototype contains some differences. Due to the relative immaturity of ATM UNI standards and lack of implementation, a workaround was required in order to implement the Leaf Initiated Join (LIJ) functionality which was desired. An out-of-band communications channel using TCP/IP was used in order to provide this functionality. Refer to section 5.8 for an in-depth discussion. A second difference pertains to the slide presentation tool. Static image data of which slides are an example, require a guaranteed transport mechanism in order to guarantee that every conferee receive the slide images without error. Because the distribution of these images is over a native ATM point-to-multipoint connection which has no end-to-end error guarantees, it is quite feasible that a specific conferee could receive an erroneous copy of this image and would require a retransmission. This selective retransmission mechanism has not as yet been selected and implemented by the ATM Forum in the UNI standard. Since the development of a proprietary multipoint transport protocol was not of primary interest to this project, the slide presentation tool was not included in the prototype. It should be noted that there exists ongoing work within many standards bodies such as the IETF to develop multipoint transport protocols which would make tools of this type feasible in the near future.

5.4 Operation of a Video Conference Session

A typical ATM RendezView conference would proceed as follows. A conferee would first log into a workstation. After performing a personal registration procedure, the conferee would attempt to gain access to the Registrar's database. Once successful, the conferee would receive a listing of available conferences from which to choose. The Registrar would grant access to a particular conference by returning the current state information of that conference. This state information would include personal information regarding other conferees as well as the media streams they are currently offering. State information is dynamic in nature and is updated almost instantaneously when any changes occur in the conference state.

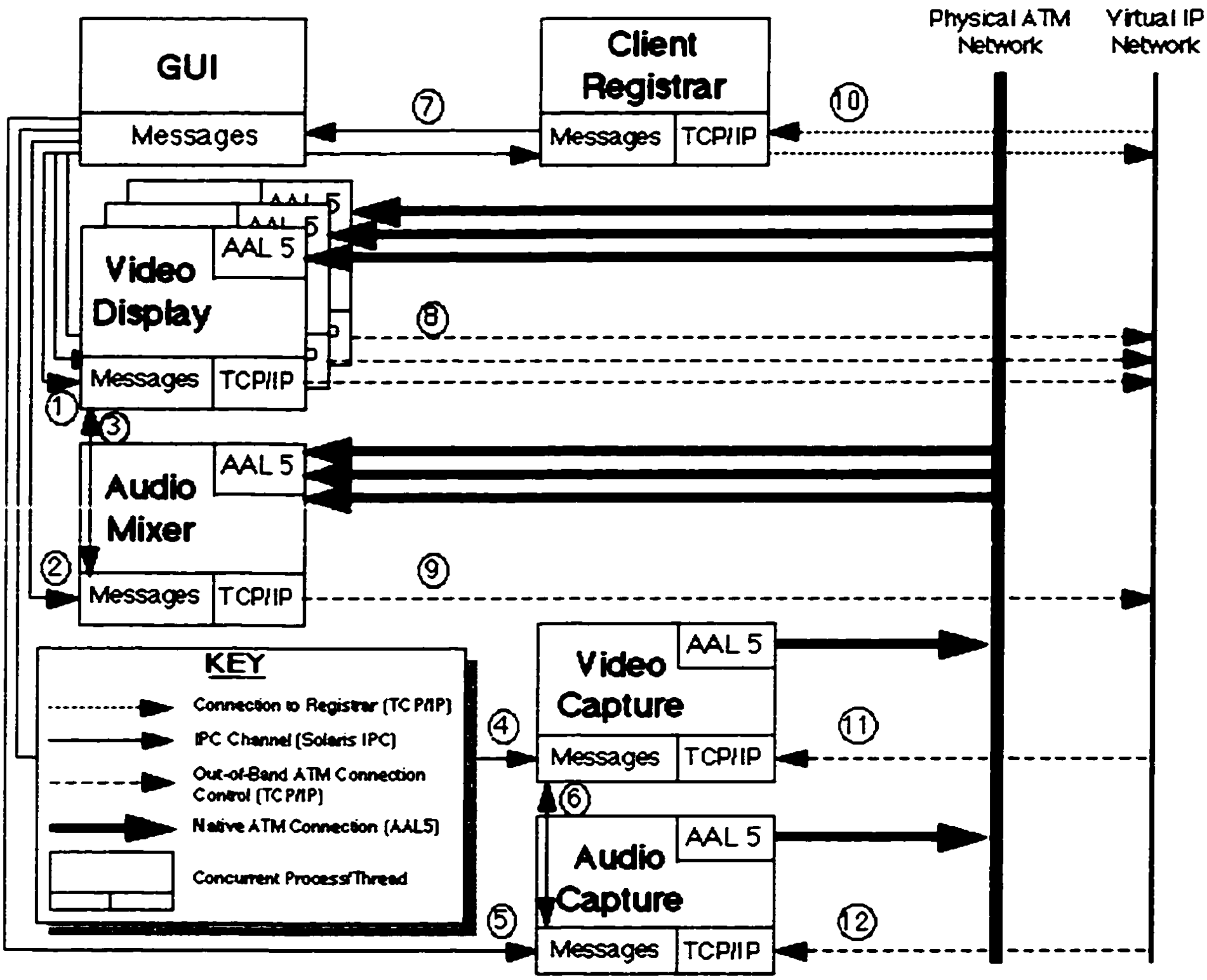
The next step would be for the conferee to indicate via the GUI, which local media streams are to be offered to the other conferees. These state changes would be propagated via the Registrar to all other conferees. The conferee would have the ability to now select from the available list, the media streams of other conferees that are to be received. This would invoke the display components highlighted above.

At any further point in time, if any changes are made to the conference state (due to a conferee joining or leaving, or changing their media streams offered), every conferee would become aware and would be able to respond appropriately.

5.5 System Components

The functional software components of the ATM RendezView prototype for an end-system (workstation) are shown in Figure 5-2. This figure shows rectangular boxes which can be thought of as representing unique UNIX processes or threads and hence can be expected to operate in a concurrent fashion. The joining arrows indicate the direction of information flow between components, where lighter lines indicate message passing functions, such as IPC, and heavier lines represent the actual media stream transmissions. Each IPC channel is labeled with a number and shows the primary direction of message transmission, however ACK or NACK type replies may be transmitted in the opposite direction. This section describes the function operation of each component in Figure 5-2 in detail.

Figure 5-2: ATM RendezView End-System Software Components



5.5.1 Conference Registration Mechanism

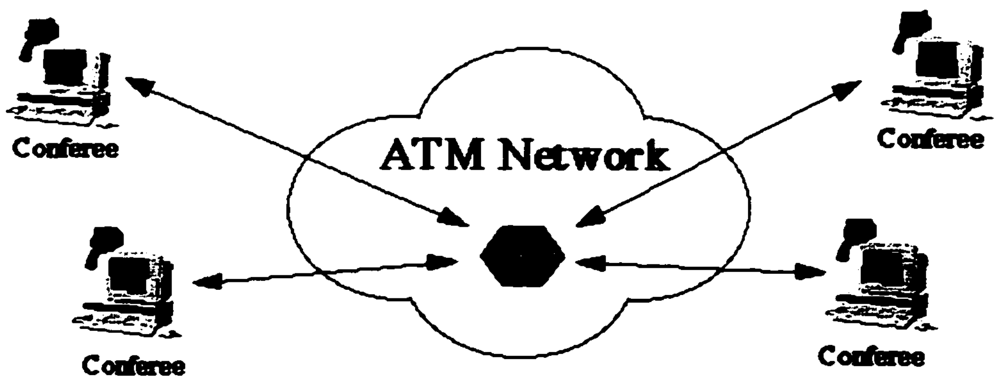
5.5.1.1 Overview

A registration mechanism is a facility which is required in every conferencing system. This facility encompasses the actions which allow potential conferees to gain access to the virtual conferencing environment. Additionally, it may be employed repeatedly throughout the lifetime of the conference as new conferees join or others leave. This facility typically enables the guaranteed distribution of state or configuration messages specific to a conferee which would enable others to become globally aware. The message content could be either of a personal nature such as a conferee's name, mugshot, email address etc., a media-specific nature such as media types, encoding style, data rates etc., or it could be network type information such as end-system addresses. Irrespective of the actual information being distributed, this facility is necessary to guarantee the initial and on-going operation of the conference.

5.5.1.2 Centralized vs. Distributed Messages

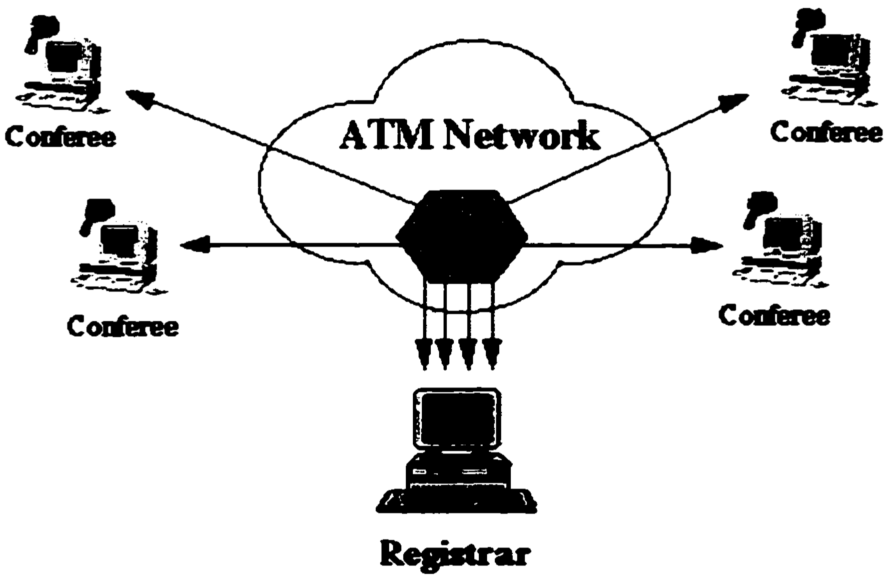
The ideal topology for the exchange of registration messages would be via a guaranteed (error-free) end-to-end multipoint transport connection. Therefore, every conferee within a particular conference would be connected as shown in Figure 5-3. If a conferee transmitted a message onto this connection, every other conferee would receive an exact duplicate of that message simultaneously and without error. Consequently, every conferee would have to maintain a copy of the state information for all conferees in a distributed fashion, and update it appropriately as new messages arrived. There are two fundamental problems which immediately present themselves. First, there is no such concept of multipoint connections as defined by the ATM Forum. Even if there was, or multiple point-to-multipoint connections were established to mimic a multipoint connection, there is still another problem. ATM connections by their very nature are connection-oriented and non-guaranteed transmission pipes. Therefore, some kind of multipoint transport protocol would have to be designed and implemented in order to offer the characteristic of a guaranteed end-to-end multipoint transport connection.

Figure 5-3: Guaranteed Multipoint Transport Connection



Because this sort of design effort is not the primary goal of the prototype, a simpler centralized approach is adopted. A new component called the ***Registrar*** is introduced. The Registrar is a central database which keeps an updated record of all pertinent state and configuration information of all registered conferences and conferees. Through this means, the Registrar is now responsible for providing a *virtual* guaranteed multipoint transport connection. This is achieved by using point-to-point TCP/IP connections between each conferee and the Registrar as shown in Figure 5-4. The ATM network may still be exploited for its low latency and high bandwidth by encapsulating IP frames within ATM cells. Upon the receipt of a message from a conferee, the Registrar is responsible for updating all other conferees in a unicast fashion. Even though this method is not eloquent or highly efficient, it is functional and assuming that messages are infrequent and small in size, it is adequate.

Figure 5-4: Registrar Connection Topology



5.5.1.3 Message Semantics

5.5.1.3.1 Message Types

Messages fall into four types. Only one of these may be issued by the Registrar and the remainder are issued by a conferee. They are characterized as follows:

- REGISTER

This message type is issued by a conferee upon initially joining a conference. It identifies the conferee's personal configuration information. A response from the Registrar takes the form of either a rejection notice or an UPDATE message indicating the current conference state. Additionally, all other conferees receive an UPDATE message indicating the existence of a new conferee.

- UNREGISTER

This message type is issued by a conferee who wishes to leave a conference. It causes the conferee to be removed from the Registrar's database, and an UPDATE message to be sent to all remaining conferees informing them of this state change.

- AMEND

This message type is issued by a conferee who has changed the state of their local conferencing environment and wishes to propagate this information. The Registrar distributes this new state information by issuing an UPDATE message to all conferees.

- UPDATE

This message type is the only one issued by the Registrar. It informs conferees of any changes which have been made to its database through one of the message types indicated above. It enables conferees to update their own local view of the conference state.

5.5.1.3.2 Message Contents

Message contents may fall into one of two categories: conferee-specific or media-specific information. Conferee-specific information may include an *ATM End-System Address* (AESA), a full name, a nickname, a password, an email address, various telephone numbers, a mailing address, a mugshot and any other personal related information that may be informative to other conferees. Media-specific information would include a verbose media title, a media stream type, a media stream specific encoding structure and the media stream's *Global Call ID* (GCID).

5.5.2 Message Channels

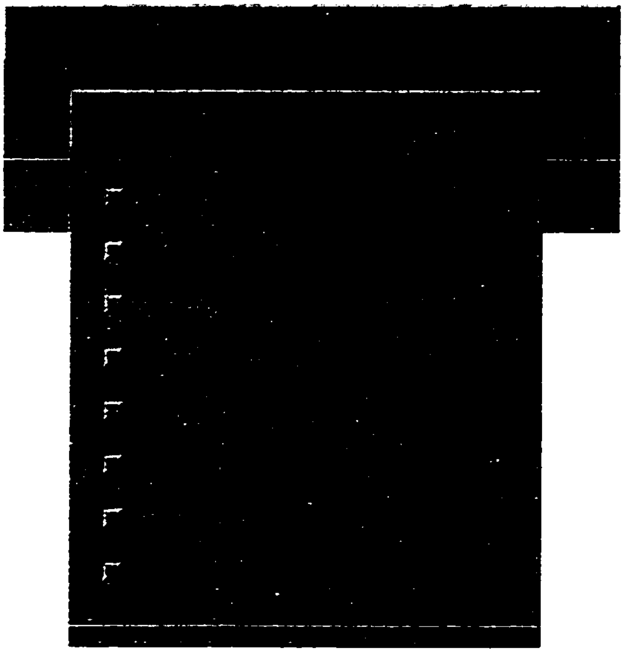
The messages exchanged between the various components are classified according to the channel upon which they travel, which have been numbered from one to twelve in Figure 5-2. These message channels are an integral part of the system as they allow for communications between various components residing on both local and remote end-systems. Channels one through seven are for local Inter-Process Communication (IPC) and employ Solaris 2.4 IPC messages. This form of IPC allows for two processes to exchange variable length messages in a bidirectional fashion, while being able to classify messages by type or priority. In addition, messages may be retrieved in any order of priority. Channels eight through twelve employ the TCP/IP protocol to provide a guaranteed point-to-point communications channel between networked end-systems. These TCP/IP connections are transported across the physical ATM network in the ATM RendezView prototype however this is not a requirement. These channels (excluding channel 10) are used for out-of-band ATM signaling to implement Leaf Initiated Join (LIJ) functionality. Channel ten is used to communicate with the Registrar which redistributes state information to all other conferees. The solid lines in Figure 5-2 represent the transport of actual media streams natively in AAL-5 frames over the ATM network.

5.5.3 Graphical User Interface

The Graphical User Interface (GUI) provides the interface between the end-user and all of ATM RendezView's features (Figure 5-5). In essence, it is the heart of the prototype since it assumes the role of command and control center. Its functionality enables a user to:

- Register to the local environment
- Apply to the Registrar for membership within a particular conference
- Obtain a dynamic view of the conferees within a conference
- Selectively receive media streams being offered by other conferees
- Selectively advertise and offer media streams to other conferees
- Leave a conference

Figure 5-5: ATM RendezView Graphical User Interface



Based on received configuration and state information from the Registrar, as well as conferee selections within the GUI, control messages are sent to the appropriate components. For example, assume that the Registrar has conveyed the existence of audio and video streams being offered by conferee *Lab* to all other conferees. The local conferee may choose to receive these two streams and may indicate this via the GUI. The GUI would in turn issue a control message to the Video Display component of the form open(*Lab*,video), and another control message to the Audio Mixer component of the form open(*Lab*,audio). These two components would then take it upon themselves to establish the appropriate ATM connections to enable the receipt of these media streams. Upon reception of the streams, these components would also be responsible for properly displaying these streams in a synchronized fashion. In a similar manner, the GUI controls the function of all other components.

5.5.4 Client Registrar

The Client Registrar has the function of providing bidirectional information exchange between the Registrar and the GUI. On the network side (message channel 10 in Figure 5-2) it must monitor for UPDATE messages being generated by the Registrar and convey them to the GUI in an appropriate fashion. On the GUI side (message channel 7 in Figure 5-2) it must monitor for state information changes generated by the GUI and formulate them as either REGISTER, UNREGISTER or AMEND message to the Registrar on the network side.

The reason for implementing this functionality within a separate process and not integrating it within the GUI itself is for implementation reasons. Because the receipt and transmission of data onto a network connection is inherently bursty and subject to delays, this would compromise the responsiveness of the GUI while it performed network functions. Through this means, the Client Registrar can efficiently queue incoming messages from the Registrar and communicate them to the GUI using IPC when available.

5.5.5 Video Capture

The Video Capture component's main function is to code (digitize and compress) a live video signal from a camera or other video device and distribute it across the network to any conferees who request it. The coding process uses Motion JPEG for compression/decompression of video streams. Once this component is launched by the GUI, it is able to work relatively autonomously requiring no control from the GUI. The component has the added responsibility of monitoring the network for incoming LIJ requests from other conferees who wish to receive this video stream. It then assists in negotiating the required ATM connection. Video data units are time-stamped and numbered in order to facilitate synchronization with associated audio upon playback by remote conferees.

5.5.6 Audio Capture

The Audio Capture component's operation is similar to that of the Video Capture component. Its function is to capture and digitize a live audio signal from a microphone or other audio source and distribute it across the network to any conferees who request it. Once this component is launched by the GUI, it is able to work relatively autonomously requiring no control from the GUI. The component has the added responsibility of monitoring the network for incoming requests from other conferees who wish to receive this audio stream. It then assists in negotiating the required ATM connection. Audio data units are timestamped and numbered in order to facilitate synchronization upon playback by remote conferees.

5.5.7 Video Display

The function of the Video Display component is to playback a video stream to the local display in a synchronized fashion as the data is received from the ATM network. As with other components, this one is launched and configured appropriately by the GUI. The component must first negotiate an ATM network connection between itself and the conferee's end-station which is the source of the video stream. It must then begin receiving the video stream and display it in a synchronized fashion. Assuming that an audio stream is currently being received which is associated with the video stream, the Video Display component must communicate with the Audio Mixer component (message channel 3 in Figure 5-2) to achieve inter-stream synchronization. As shown in Figure 5-2, the GUI will launch a separate process/thread to handle each individual video stream being received by the conferee. Therefore, there can be multiple Video Display components communicating with the Audio Mixer Component at any one time.

5.5.8 Audio Mixer

The function of the Audio Mixer component is to playback all of the incoming audio streams to the local audio device in a synchronized fashion as the data is received from the ATM network. As with other components, this one is launched and configured appropriately by the GUI. Unlike the Video Display component, only one Audio Mixer component can ever exist. This is due to the fact that audio streams must be mixed together before playback. The GUI uses message channel 2 in Figure 5-2 to indicate when new audio streams are to be received and mixed or removed. The Audio Mixer is then responsible for negotiating the required ATM connections with the audio stream sources and consequently adding these streams to the pool of audio streams to be mixed. Synchronization is performed based on the received timestamps and sequence numbers associated with media data units.

5.6 Designing for Performance and Recovery

The design of a real-time multimedia application such as ATM RendezView requires a certain awareness of issues which must be considered when dealing with real-time deadlines. Also, the multimedia aspect of such an application immediately implies potentially high bandwidth data transfers coupled with the need for media synchronization techniques. Ultimately, a certain level of robustness and recoverability from violated time deadlines is a necessity. Some of these issues and the methods adopted to address them are highlighted as follows:

- Exploit available soft real-time facilities

The bane of current workstations is their Time Sharing Operating Systems (TS-OSs). Ironically, most workstations have audio/visual facilities, however no support for hard real-time processing. Hard real-time differs from soft real-time in that the workstation hardware and software has been engineered specifically to enable processes to negotiate a QoS contract which the processing subsystem honors. Soft real-time takes the form of software based facilities which attempt to offer some real-time processing characteristics to executing processes. This may take the form of a preemptive operating system, a scheduler which supports a real-time class or programming libraries which allow for lightweight processes or threads. All these soft solutions tend to aid the designer in achieving real-time performance and should therefore be exploited, however these aren't nearly as effective as a hard Real-Time Operating System (RT-OS).

- Recovery from media stream discontinuity

Loss of media stream continuity is usually caused by the network transport more so than by the local end-system. If ATM is being used for transport, losses are possible since ATM only provides statistical guarantees for error-free data delivery. Even though error rates should be quite low within an ATM network, the application should be aware that losses are possible and should deal with such situations in a fashion which disrupts the perceived quality of the media presentation as little as possible. Such mechanisms are achieved using complex synchronization algorithms which deal with media stream discontinuity.

- Inter-media and intra-media synchronization

Any multimedia system inherently requires both inter-stream and intra-stream synchronization which is dependant upon the media streams being presented. For example, if related audio and video streams are being presented, studies indicate that the maximum inter-stream skew should be limited to 80 milliseconds and that the one way delay from sender to receiver should be limited to 150 milliseconds [Steinmetz96]. For the multimedia application to be perceived as performing correctly, these real-time constraints must be respected.

- Proper use of buffers

Buffers are commonly used within multimedia applications in order to absorb jitter, remove skews and delay media stream presentation. Though they can be quite useful, the designer must carefully size buffers so that they do not have a negative impact on overall performance. Small buffers are attractive since they tend to reduce end-to-end delays, however these tend to incur high processing overheads within the operating system which can degrade performance. Conversely, large buffers tend to be much more efficient from an OS perspective, however they force a coarser grained control of media units and increase transmission latency which is not ideal for continuous media.

- Exploit hardware for performance gains

The designer must be aware of the options available in the form of hardware modules. In the majority of cases, a hardware vs. software implementation of some functionality is more performant and usually significantly more so. ATM RendezView exploits a hardware based Motion JPEG CODEC, audio digitizer, and AAL5 SAR implementation. All of these hardware implementations off-load processing from the main end-system CPU thus making the system much more performant.

- Lightweight software modules

A key to designing software modules is to keep them as simple and as lightweight as possible. Certain functions such as mixing and synchronization algorithms can be CPU intensive. However, an efficient design which removes any redundancy or extraneous functionality can make a significant improvement. A synchronization algorithm such as the one implemented within ATM RendezView acknowledges the need for a multilevel synchronization scheme. A very tight (CPU intensive) synchronization mode is only required when synchronization is lost which is infrequent, and so a three level synchronization scheme is adopted (section 5.7.3.4). Coding style can be a major factor in reducing the CPU load of a particular software module. Various code tuning and optimization pointers can be found in [Cockcroft95], [Dowd93].

- Respect QoS constraints for media streams

Multimedia processing implies that media streams must receive a certain quality of service in an end-to-end fashion. If the network transport is a high bandwidth, low latency medium such as ATM, and the end-systems possess less than adequate processing power, the system will never function well enough to meet the QoS demands placed by the media streams. Consequently, a complex synchronization algorithm running on the end-systems will be of little use, and will serve to be more of a detriment to the overall performance of the system.

- Design for scalability and robustness

The multimedia system must be designed such that it is scalable and robust in an end-to-end fashion. In a video conference application such as ATM RendezView, it is quite feasible to have six or eight conferees participating in a video conference. The prototype should be designed in such a fashion that it is capable of processing the audio and video streams of six conferees as a minimum. If the Motion JPEG CODEC is unable to process more than three video streams for example, this reduces much of the functionality and strength of the prototype.

5.7 Media Stream Synchronization

5.7.1 Philosophy

The single most important element of any real-time multimedia communication system is the synchronization mechanism which must be employed in order to guarantee timely and sensory-appealing playback of multimedia streams to a user. The synchronization philosophy adopted in ATM RendezView is based loosely on concepts expressed in [Chen96], [Nicolaou90] and [Chen95]. The philosophy can be generalized as *master/slave synchronization with a delay/drop policy* and dictates the following: Given *n* real-time media streams which are to be received at a workstation and subsequently synchronized and displayed, one of these streams is classified as the master stream and the remaining *n-1* are classified as slave streams. The master stream is usually chosen as the one which requires the most stringent QoS and hence receives the highest priority of service within the end-system. All slave streams are synchronized using the master stream as a reference.

Since the master stream is processed with the highest priority, we assume that all of its presentation deadlines are met in an end-to-end fashion. The slave streams are then processed and displayed using the master as a time reference, and at a lower priority. Assuming a resource (such as the network or end-system) is not able to process a slave stream in a timely enough fashion, the delay/drop policy is implemented. Essentially, if a slave stream's data unit arrives late and misses its deadline, it will simply be discarded. Otherwise, if it arrives before its scheduled display time, it will be delayed for later display.

The above philosophy was adopted for the following reasons. There is a general assumption that all the components (network and end-system) in the system are performant, largely error free and bound in terms of the delay and jitter they inject into a media stream. Therefore, for the most part there will be little need to interrupt the natural flow of the media stream. Thus a lightweight synchronization scheme is desirable. Since we are dealing with live media streams, a minimum end-to-end delay is also desirable to enable conferees to converse in a natural fashion. End-to-end performance is favored over absolute stream continuity, which supports the need for a simple, lightweight synchronization protocol. Ultimately, the increased bandwidth offered by ATM and hence the reduced processing overhead required to perform media stream compression/decompression operations, coupled with the reduced processing overhead associated with network protocols, reduces the necessity for complex, compute intensive synchronization protocols. This overall decrease in end-system processing requirements is the fundamental advantage of adopting ATM as the network transport facility of choice.

5.7.2 Issues

There are a number of fundamental issues which must be addressed in attempting to realize a synchronization mechanism on a UNIX based workstation. The most important being "how does one implement a time-dependant mechanism in an inherently non-deterministic system?". The following list highlights many of the issues that must be addressed:

- Inter-media synchronization
- Intra-media synchronization
- System resource reservation

- Network end-to-end delay, loss and jitter
- Processing delay and jitter
- Buffer starvation and overflow
- Processor, I/O and network performance
- Implementation of QoS specifications under UNIX

These points highlight some issues that must be resolved. Many technical reports deem UNIX to be a less than adequate solution for multimedia processing [Chen96] [Bulterman91], however with new developments such as POSIX.4, Solaris threads and processing classes, real-time multimedia is becoming more and more of a reality.

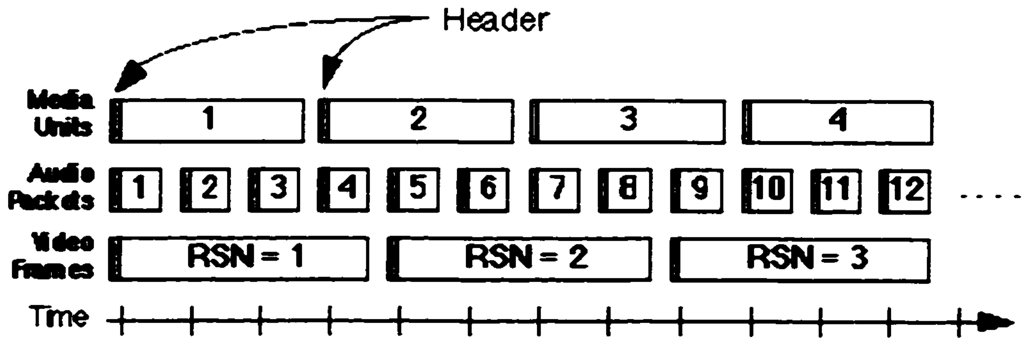
5.7.3 Concepts

In order to provide a synchronization mechanism which addresses the issues highlighted in section 5.7.2, the concept of *headers*, *media unit lifetimes*, *stream priorities* and *synchronization modes* are introduced.

5.7.3.1 Headers

Every media stream within the ATM RendezView prototype is composed of *media units*. Media units are comprised of fixed or variable length data blocks which represent a manageable set of data. The actual length of the media unit is media dependant and is chosen as a trade-off between a low processing overhead (long media unit) and a low end-to-end transmission latency (short media unit). A typical media unit could be an encoded video frame or a group of audio samples for example. Each media unit is prepended with a header at the transmitter (remote end-system) as shown in Figure 5-6. Each header contains a *Remote TimeStamp (RTS)* and a *Remote Sequence Number (RSN)*. The RSN is simply a counter which is incremented for every consecutive media unit. The RTS is the time at which the media unit was created (digitized) at the remote end-system. These two parameters are used by the receiver (local end-system) in order to re-synchronize the media streams for local playback.

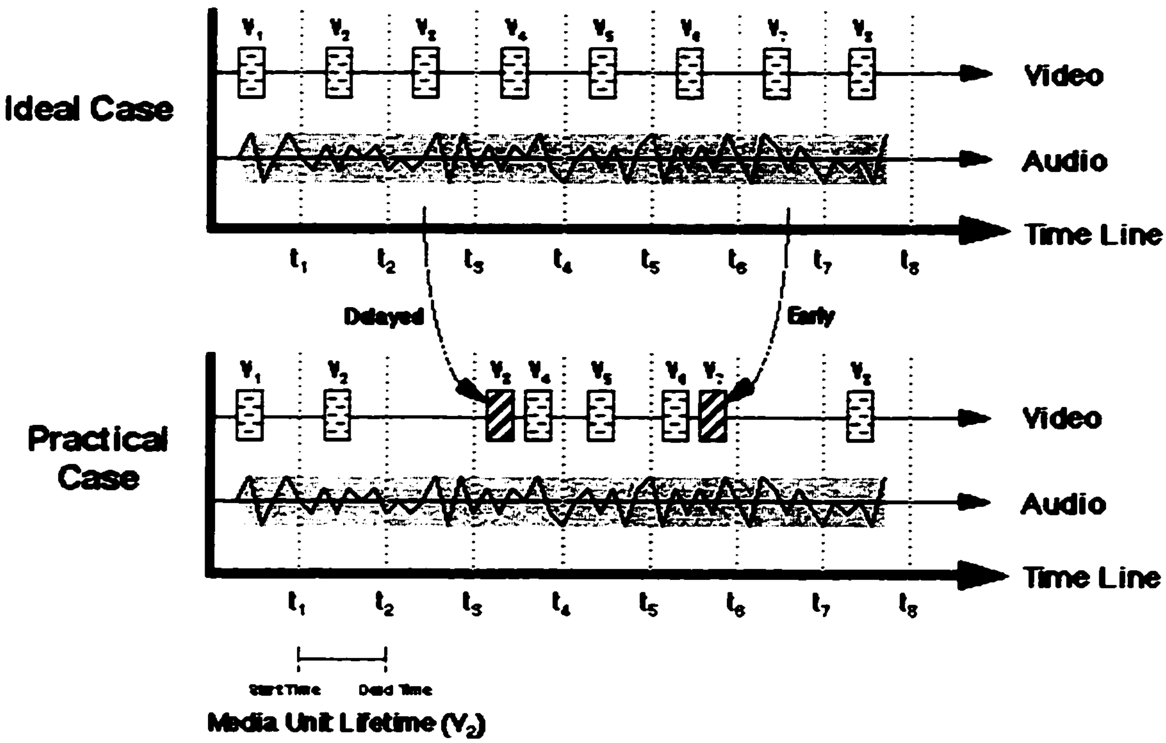
Figure 5-6: Media Stream Format



5.7.3.2 Media Unit Lifetime

Each media unit within ATM RendezView possesses the concept of a lifetime which determines whether the media unit has arrived early, late or on-time. This lifetime may be designed for any level of synchronization granularity. In ATM RendezView, it has been set to the time duration between successive media unit arrivals. This concept is depicted in Figure 5-7. Therefore, if a media unit arrives early or late as determined by its lifetime, it will be delayed or thrown out respectively. This is the means by which the delay/drop policy makes its decisions.

Figure 5-7: Media Unit Lifetime



5.7.3.3 Stream Priorities

In ATM RendezView, one stream must be chosen to be the master from which all other streams will be synchronized. Audio is chosen as the master stream since it has the greatest perceived sensitivity to delays, errors and jitter [Steinmetz96]. Therefore, the audio processing components are assigned the highest priorities for scheduling within the UNIX operating system. The slave streams are then assigned decreasing priorities to reflect their relative synchronization importance. For example, it is common to place video processing at a low priority since a lost frame will most likely not be detected, whereas a lost audio frame would produce quite an audible break or clipping noise.

5.7.3.4 Synchronization Modes

There are three modes of synchronization which may be enforced namely *tight*, *loose* and *free-flow*. Each determines the granularity to which synchronization of media streams will be performed. Tight synchronization dictates that media units are synchronized based on a timestamp in conjunction with the receiver's local clock. Loose synchronization dictates that media units are synchronized based on their RSN. The RSN is used to verify continuity of the media stream (intra-stream synchronization) and is not based directly on timing properties. Free-flow synchronization essentially implies no synchronization, simply a servicing of media units as they arrive from the network.

These various modes of synchronization are implemented in order to allow a varying degree of synchronization as required. Some modes may induce others based on some observed condition. For example, if a stream is being loosely synchronized and media unit continuity is violated, the stream synchronization will have to be recovered via a switch to a tight mode of synchronization.

Various modes of synchronization are desirable since different media types inherently require different levels of synchronization. Additionally, a highly reliable network or a I/O hardware device which performs its own timing may exist, which obviates the need for processor intensive synchronization algorithms. This three level synchronization operation offers a dynamic solution adaptable to most media types.

5.7.3.5 Inter-Stream Synchronization

The inter-stream synchronization is achieved in a very simple fashion. Each media stream being received acquires its own autonomous thread. The only link required between all these threads is a parameter which indicates the time skew which exists between the sender's (remote) and receiver's (local) end-systems. This parameter is called the *Synchronization Time Differential* (STD) and is determined by the master stream's thread and is distributed to all of the slave streams' threads. In this fashion, once this parameter is distributed, all threads can work autonomously and schedule their media presentation accordingly. The advantage being that this configuration is scalable, does not impose any limitations on the encoding formats of other media streams and eliminates the need for ongoing signaling between media threads.

5.7.4 Methodology

- Phase 1: Establish Initial Parameters

In the first phase of operation, media streams begin arriving and each stream's thread determines the STD between the local and remote end-systems, as well as the initial RSN. The RSN is obtained from the header of the very first media unit to arrive. The STD is obtained in one of two ways. If a master stream exists, then all slave stream threads obtain the STD from the master stream's thread. If no master stream exists (which is highly unlikely), the slave stream thread will determine its own STD.

The master stream's STD parameter is determined as follows. Upon arrival of the very first media unit of a stream after opening a network connection, the RTS found in the header and the local system time are subtracted. This figure indicates a skew which exists between the local and remote end-systems' system clocks in addition to processing and propagation delays. A small time delay (δ) is added to this skew in order to produce the STD. This δ corresponds to a playback delay required in order to allow jitter buffers to be populated.

- Phase 2: Initiate Media playback

Once the initial parameters are established, the synchronization and playback process may begin. The synchronization style employed is dependant on the media type itself, which is summarized as follows:

—Audio

Audio playback operates in a loose synchronization mode. As audio media units arrive, their RSN is verified to guarantee that no data has been lost in transmission. These media units are copied to output audio buffers without any additional timing mechanism. The audio device hardware is responsible for intra-media synchronization and so the audio stream thread need only worry about output buffer overflow or starvation. In the case that one or more media units are lost or buffer starvation occurs, resynchronization and recovery is effectuated through a switch to a tight mode of synchronization which employs the STD. This mode is implemented until such time as the synchronized media unit flow is re-established.

—Video

Video playback operates in a tight mode of synchronization in a continuous fashion. Therefore, every media unit has its RTS inspected in order to determine its presentation time. The video frame lifetime is used in order to implement the delay/drop policy. Unlike the audio output device, the video output device has no intra-stream synchronization mechanism, therefore video media units must be synchronized continuously.

—Generic Media

Any arbitrary media stream has the ability to be synchronized in any one of the three modes outlined in section 5.7.3.4 which affords maximum flexibility and extensibility. The mode(s) chosen should reflect the degree of sensitivity of the media stream to loss, delays and jitter as perceived by the user.

5.8 Connection Architecture

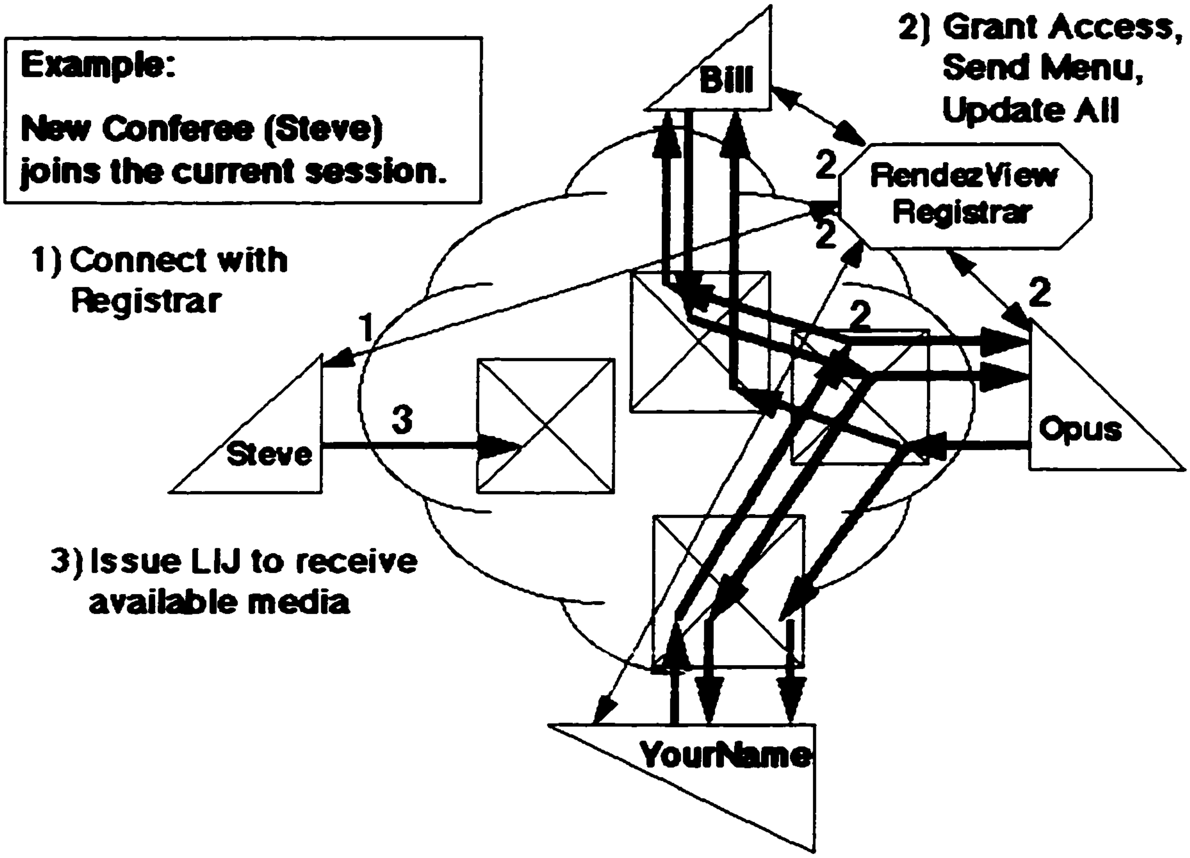
In ATM RendezView, conferees must become members within native ATM multicast connections to receive media streams. An ATM multicast connection, as defined under the ATM Forum's User Network Interface (UNI) standards documentation versions 3.1 and later, employ the tree analogy to produce the concept of a point-to-multipoint connection. There exists one root which is the source or sender of any data on the multicast connection, and multiple leaves which are the receivers or sinks of data on the connection. A multicast connection is owned by the root. Not only does the root determine the QoS details of this point-to-multipoint connection, but it also determines who is allowed to become a leaf. The root may optionally empower the network to accept anybody's request to become a leaf of its point-to-multipoint connection without being notified. To become a leaf, a user must signal their intentions at the UNI. The network is then responsible for accepting or rejecting this request. If the leaf is accepted, the network then becomes responsible for any data replication which must occur. It should be emphasized that multicasting is a native ATM network-based service; the root of the connection is not responsible for routing or duplicating data streams.

An important advantage of native ATM point-to-multipoint connections is that they are dynamic, hence the number of leaves may change throughout the lifetime of the connection.

This is extremely useful in a video conference where a switched presence environment is desired and bandwidth may be at a premium. Assume that each participant is multicasting their video stream into the network and are each the root of a unique point-to-multipoint connection. Other conferees wishing to see only a subset of these video streams may dynamically add or remove themselves from any of the multicast connections. Conferees may easily and efficiently implement a policy such as "view the video of the current speaker only" or even "view the video of the three most recent speakers", without the need for an expensive and complex conference bridge. This type of service demonstrates the potential value of ATM's native multicast capabilities.

One important component of ATM multicasting is the method in which leaves indicate their desire to be added to a multicast connection. Two methods exist today namely Leaf Initiated Join (LIJ) (implemented in ATM Forum's UNI v4.0) and Root Initiated Join (RIJ) (supported in previous standards such as FORE Systems' SPANS and ATM Forum's UNI v3.1). LIJ involves a leaf signaling their intention directly to their ATM User Network Interface (UNI) whereas RIJ requires the leaf to use some external out-of-band method of informing the root of their desire and having the root perform the network signaling on their behalf. LIJ is the most useful and functional since it does not need to involve the root, however there are no implementations to date due to the immaturity of this technology. Therefore, the ATM RendezView prototype was forced to employ RIJ and chose to employ FORE System's SPANS Application Programming Interface (API). Figure 5-8 demonstrates how ATM RendezView would operate utilizing a LIJ mechanism as implemented in the UNI v4.0 standard.

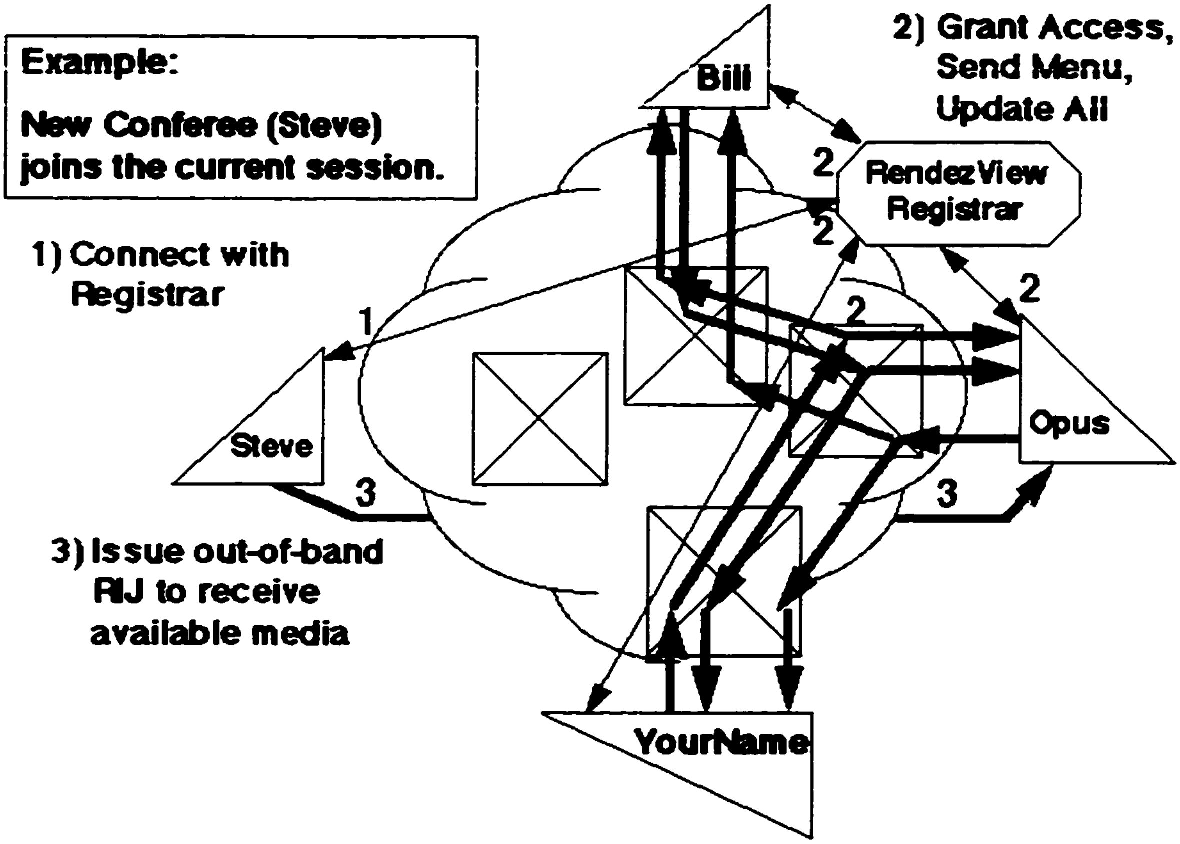
Figure 5-8: ATM RendezView Exploiting Leaf Initiated Join (LIJ) Functionality



Steve must first connect with the Registrar in order to obtain state information about other users. In step 1, Steve establishes a point-to-point connection to the Registrar located at a globally known address. After completing a security validation, the Registrar returns conference state information to Steve describing the current state of the virtual conference environment (step 2). This state information provides personal details on each of the three conferees (name, address, email address, mugshot etc.) as well as media specific information on each conferee (media stream types offered, encoding specifics, ATM multicast addresses etc.). In step 3, Steve is globally aware of his conference environment and can now issue LIJ requests to the ATM network in order to be added to any of the three conferees' existing multicast connections to receive audio, video or other media. At this point, the network will establish the appropriate connection to add Steve as a leaf. Any information subsequently transmitted by the root onto these multicast connections will be duplicated and sent to Steve.

The ATM Forum has only just recently completed the UNI v4.0 specification and so it is far too early to see any product implementations. Consequently, ATM RendezView doesn't have the luxury of Leaf Initiated Joins (LIJs) and therefore employs Root Initiated Joins (RIJs) as defined in the previous UNI v3.1 standard. The connection setup phase for RIJs proceed in a similar fashion, however each requesting leaf must first establish a point-to-point out-of-band signaling connection with the root to request that a RIJ be performed on their behalf. ATM RendezView uses a TCP/IP/ATM connection for this purpose. Once a leaf makes its request, it waits to be called back by the network in order to be added as a leaf to a point-to-multipoint connection. The root end-system must have the functionality required to respond to incoming requests from leaves. Figure 5-9 demonstrates this scenario.

Figure 5-9: ATM RendezView Exploiting Root Initiated Join (RIJ) Functionality



In the above figure, steps 1 & 2 are identical to the LIJ case. Once Steve has been granted access to the conference environment and has received complete topological information about the virtual conference environment, he must then issue a request using out-of-band signaling to the root of each point-to-multipoint connection (step 3). In the figure, Steve would like to receive Opus' video and so he establishes a TCP/IP/ATM point-to-point connection with Opus to indicate his desire. Opus then submits a native ATM RIJ request at his UNI on Steve's behalf requesting that he be added as a leaf to his own multicast connection. The network completes Opus' request by calling Steve up to complete this request. If Steve confirms, he is added as a leaf to Opus' multicast connection and is able to receive the video stream. A similar process would ensue for receiving other media streams such as Opus' audio stream.

ATM's multicast connections are extremely versatile since they may be altered dynamically while the connection is in use. Leaves may be added or removed on the fly and the network guarantees the QoS contract initially negotiated by the root for every leaf. LIJs are the most efficient and natural connection method, however until the newer standards have been implemented by ATM hardware manufacturers, the non-scalable RIJ approach is required. The native point-to-multipoint facility offered by ATM demonstrates the advantages of operating a video conference in a distributed mode as opposed to a centralized approach where expensive MCU hardware and multiplexers would typically be required.

5.9 Hardware and Software Platform

In order to implement the ATM RendezView prototype, several fundamental hardware and software choices were made. Due to time and resource constraints, hardware design and development was not an option for the ATM RendezView prototype. It was determined however that many of the functions of the prototype such as media stream coding, decoding and digitizing would be much more efficiently implemented in dedicated hardware components. Therefore, the architecture implementation involved selecting appropriate off-the-shelf hardware components and additionally designing highly performant software components to act as a glue.

ATM RendezView was developed using the C programming language because of its large installed base, versatility and speed. Motif/XWindows was used to produce the GUI, and a

POSIX.4 compliant library along with Solaris Threads was employed to support real-time functions. The hardware employed was more selectively chosen for the following reasons:

- Sun SPARC 20 workstation running Solaris 2.4 OS

This workstation configuration was chosen for several reasons which lend well to the development of real-time multimedia applications. The Sun SPARC's SBus is supported by many third party hardware vendors. The SPARC workstation sports some impressive processing and bus bandwidth figures which are inherently required in any multimedia application. Solaris 2.4 has fairly complete support of IEEE's POSIX.4 real-time programming standard. It has support for multithreading facilities which provide the equivalent of the POSIX.4a standard. Solaris 2.4 also defines a real-time processing class which enable processes to obtain real-time service and scheduling in an otherwise non-deterministic scheduling environment.

- Native Sun Audio CODEC

Because Sun's native audio device [SunSoft94b] is bundled with the computer and is fully-featured, it was ideal. The CODEC possesses both D/A and A/D converters which are able to operate at a multitude of encoding rates and formats ranging from basic u-law up to CD quality audio. The device contains microphone and line-level inputs as well as headphone and line-level outputs.

- Parallax PowerVideo Card

Parallax's PowerVideo Card was chosen because of its on-board Motion JPEG CODEC. It has a video development environment [Laird94] including a complete set of X/Motif API widget libraries which allow for full control of the CODEC hardware thus off-loading any compression/decompression functions from the main CPU. The video hardware is capable of providing the encoding and decoding of a Motion JPEG video stream of 640 x 480 pixels, 24 bit color at frame rates in excess of 25 fps in real-time. As video resolution is decreased, additional video streams may be processed in real-time.

- FORE SBA-200 ATM NIC

The ATM RendezView prototype is dependant on many features offered by the ATM Forum's UNI v4.0 [Samudra95] standard. These include the negotiation of QoS parameters on a per-connection basis as well as the dynamic establishment and modification of point-to-multipoint ATM connections using SVCs. The advantage of employing FORE Systems ATM NIC was that it offered an API library interface to much of this functionality via its proprietary SPANS [FORE95a] signalling protocol. This signalling functionality is instrumental to the success of the ATM RendezView prototype.

Because the ATM RendezView prototype is designed to function natively over ATM in an end-to-end fashion, it is able to exploit many of the features offered by ATM. One important feature is ATM's support for switch-based multicasting. By exploiting a signalling protocol such as SPANS or UNI v4.0, ATM RendezView is guaranteed to be completely scalable to an unlimited number of conference participants. This is an ideal feature for a real-time multimedia system where most media streams such as video require high data transmission rates as well as substantial end-system resources. Thus, each end-user in a video conference experiences a workstation load which is independent of the number of conferees actually receiving their media streams.

5.10 Implementation

Not only were implementation decisions made throughout ATM RendezView's design phase, but also during the development phase as well. All of these had the goal of enabling the real-time functionality required by the prototype. The following discussion details the fundamental decisions made and how they impact the development and success of the prototype.

5.10.1 Real-time Networking

Real-time networking is a fundamental requirement for any real-time multimedia system since the network provides the most vital link between all of the communicating end-systems. This implies network connections which can offer:

- High bandwidth/Low latency
- Bounded error-rates
- Bounded delay-jitter
- Low end-system processing overhead
- Multicasting (point-to-multipoint connections)

- Negotiated QoS

Classical applications have often used IP as a network layer protocol and TCP or UDP as the transport layer protocol. There are several reasons why these are undesirable. The most fundamental is that these protocols were designed for the transport of computer data streams as opposed to time-sensitive multimedia streams. Their goal is to guarantee error-free transport of data at the cost of a high processing overhead and latency. Multimedia streams on the other hand, do not have this requirement for error-free transport, but have a time critical transmission requirement. Since these classical protocols do not address the requirements of real-time multimedia streams, there is no reason to employ them especially when they cause such a significant performance hit [Guha95] to the throughput of the end-systems involved!

The next most logical approach is to select a network technology such as ATM in order to satisfy the requirements specified above and to circumvent the need for classical protocols. Ultimately, working *natively* over ATM is the most ideal scenario, therefore an AAL5 over ATM configuration was chosen as the most practical. FORE Systems' ATM NIC implements all network protocols such as AAL5 within the hardware which off-loads processing load from the end-system.

The next question was how can a programmer access this ATM facility? As it turns out, an additional reason for choosing the FORE NIC was that it possesses two library APIs which enable a programmer access to all of ATM's network facilities. These include the SPANS API (employing FORE System's own proprietary signalling protocol) and the SD API (employing ATM FORUM's UNI v3.0 signalling protocol). Even though it is less interoperable, the SPANS API was chosen over the SD API since it is much more mature, supported and stable. Thus the final network topology chosen was AAL5 over ATM, using the SPANS API for data transfer and signalling.

5.10.2 Real-time Scheduling

Another fundamental requirement of a real-time system is its ability to process multimedia data in a timely fashion. This is extremely important for time-sensitive functions such as audio mixing where media units are short in duration and any missed or delayed processing deadlines manifest themselves as degradations in the output stream. *A processor may be performant enough to perform the necessary data computing within the required deadlines, but if the scheduling mechanism is not responsive enough, this system may not be able to adequately service this data.*

This is the scenario which exists in the UNIX operating system. Typical UNIX schedulers operate within a *time-sharing class* which has the goal of providing good response times to interactive processes and good throughput to CPU-bound processes. This is achieved by *priority scheduling* which dictates that the process with the highest priority acquires the CPU for a quantum of time called a *timeslice*. Once this quantum of time expires, the scheduler enables the next process with the highest priority to acquire the CPU for the next timeslice. The scheduler raises the priority of any process which sleeps only after a little CPU use, and lowers the priority of a process that uses the CPU for long periods without sleeping [SunSoft94e]. This has the overall effect of providing fair use and maximum utilization among system resources. [Bulterman91] reports that "the general dilemma of processing multimedia data remains that those applications requiring the most processing support are probably the least likely to get it in a general UNIX environment". Clearly this is not ideal for processes which require real-time processing since the degradation in performance of the process depends heavily on the workload of the complete end-system.

It would seem that the only solution to this scheduling problem would be that of a real-time kernel which is able to adequately service processes with real-time constraints. The drawback to this approach is the limited availability of such a kernel and the interoperability issues which would be introduced if one was used.

A more eloquent solution has been implemented by Sun Microsystems and is available under its Solaris 2.4 operating system. Real-time scheduling allows users to set fixed priorities to processes within a *real-time class* which the system cannot change. The real-time class has a higher scheduling priority than the timesharing class. Thus, the highest priority real-time user process always gets the CPU as soon as it can be run, even if other system processes are also eligible to run [SunSoft94e]. Since the SUN workstation contains a preemptive processor, this eloquent solution provides a simple means of providing the responsiveness required by many real-time processes.

5.10.3 Real-time Coding

5.10.3.1 POSIX.4

The POSIX.4 standard defines a multitude of real-time extensions to standard UNIX which are integral to any development process where real-time performance is desired as well as interoperability with other POSIX compliant systems. Of these *memory locking*, *real-time signals* and *real-time timers* are exploited by the ATM RendezView prototype.

5.10.3.1.1 Memory Locking

One major source of performance degradation for real-time applications which is independent of the sophistication of the scheduling mechanism being employed is *virtual memory*. Virtual memory is employed in most operating systems which exist today and involves the mapping of virtual addresses to different physical locations. Virtual memory has an extremely important role to play within multiuser systems since the aggregate memory requirements of all executing processes usually exceeds the physical memory facilities of the end-system. Briefly, virtual memory acts to move pages of memory out to disk when they are not being used, and re-uses this memory for a process which is about to be scheduled on the CPU. Additionally, it must return this process memory back from disk when the process reacquires the CPU.

Two facilities which comprise virtual memory, namely *demand paging* and *swapping* inject significant delays between the time a process is granted exclusive access to a CPU and the time when processing actually begins. The problem is that paging takes a lot of time and one never knows when the operating system is going to decide to page a process' memory out to disk. [Gallmeister95] reports that it takes in the neighborhood of 8 milliseconds to return one page of paged data from disk! Swapping is much more costly since a process' complete memory segment is copied to disk. This is clearly intolerable for real-time applications where such latencies are significant.

A simple solution to this problem is provided by a POSIX.4 real-time extension. It enables an application to lock down sections of its memory segment in order to disable any type of paging or swapping. Though this may be considered somewhat wasteful in terms of memory resources, the actual memory segments locked down are small. This simple solution can make a world of difference in terms of the performance and responsiveness of a real-time application.

5.10.3.1.2 Real-time Signals

Signals as defined under conventional UNIX, provide an asynchronous means of notifying a process or thread when a special event has occurred. Such events could include the expiration of a timer, arrival of an IPC message, or completion of an I/O operation. Signals are quite commonly used within UNIX and can be quite useful for multimedia programming for operations such as synchronization. There are however some drawbacks to signals as defined under conventional UNIX. These are:

- Not enough signals for application usage

Aside from the signals defined for OS usage, only two additional signals exist for use by applications. Real-time programming would require many more signals for this facility to be of any value.

- Lack of signal queueing

In traditional UNIX, signals are not queued. If the same signal arrives multiple times and cannot be immediately delivered to the process, only one is recorded and the rest are lost.

- No signal delivery order

If multiple signals are pending on a process, there is no specification as to what order these signals will be delivered. This is undesirable if each signal has a different priority as could be the case in a real-time program.

- Poor information content

The only information that is delivered upon the receipt of a signal is the actual signal type itself. There is no ability to specify additional information that may be relevant to the event that caused the signal in the first place.

- Slow

Traditional UNIX signals are inherently very slow due to the overhead they carry. The receipt of a signal typically requires a lengthy context switch, and so this facility is not well suited to real-time applications.

All of these drawbacks to the traditional UNIX signal facility justify POSIX.4's specification for real-time signals. It proposes a much more functional and real-time oriented signals facility which provides the following features:

- Additional set of application level signals
- Real-time signals are queued and not lost
- Real-time signals may carry context specific data
- Real-time signals are delivered in order of priority, not arrival
- Real-time signals are faster since no context switch is required

All these features lend well to the use of signals in real-time application development and promote POSIX.4 as an instrumental tool for any real-time UNIX based development.

5.10.3.1.3 Real-time Timers

POSIX.4 real-time timers allow an application to define thirty-two high-resolution timers per process, up to a resolution which is based on the end-systems clock. They can be either one-shot or periodic timers and employ the POSIX.4 real-time signals facility to notify an application of a timer expiration event. This allows the application to take advantage of all the features of POSIX.4 signals and additionally allows the application to specify priorities for individual timers.

5.10.3.2 Solaris Multithreading

Threads, *mutexes* and *conditional variables* are all multithreaded programming mechanisms designed to enable concurrency and parallelism within computer software. These are well suited to applications which demand high performance, responsiveness and efficient utilization of available computer resources. This is just the case with real-time multimedia since the processing, transmission and synchronization of media streams is conceptually a highly parallel activity.

POSIX.4a defines a standard for a multithreading extension to the standard POSIX.4 implementation called *pthreads*. However, this standard had not been finalized early enough to be incorporated into the Solaris 2.4 operating system. Therefore, Sun Microsystems included its own multithreading extension *Solaris threads* which is functionally the same as pthreads but possesses a different interface. Solaris threads were designed in such a fashion that future porting to pthreads would be straightforward [SunSoft94a].

A thread is a sequence of instructions executing within the context of a process. There are many advantages to using threads within both real-time and non-real-time applications alike, these are:

- Improves performance

Because the primary operation of threads is at the *user level* and not the *kernel level*, less system resources are required which improve the performance of the overall system in a dramatic fashion. Specifically, programmers can expect increased application throughput, increased responsiveness, enhanced IPC and efficient use of system resources [Lewis96].

- Improves application responsiveness

Any program which performs many activities which are not dependant upon one another can be designed such that each activity is allocated to a thread. This is especially useful where activities are conceptually required to be completed in a concurrent fashion.

- Uses multiprocessors efficiently

Programs which are designed around threads are free to execute without concern for the number of processors which may exist within the end-system. If only one processor exists, the threads facilities will schedule tasks in the most efficient fashion in order to achieve maximum concurrency among threads. If multiple processors exist, these tasks shall be scheduled among all processors to achieve the maximum level of parallelism possible.

- Uses fewer system resources

Threaded tasks share memory segments and hence need not task the operating system with costly context switches, operating system state tables or other system dependant mechanisms. Threads function at a user level and thus unload many functions from the kernel and operating system.

- Improves program structure

Regular programs are inherently synchronous flows of commands with inefficient mechanisms available for implementing concurrent activities. Multithreading allows an extremely simple method of creating concurrent programs without the need for complex, heavy-weight IPC mechanisms for inter-thread communication.

Threads provide a simple yet powerful method of delineating concurrent tasks and enabling an end-system to service these tasks properly and in a timely fashion. Inter-thread communication is also required to enable independently executing threads to communicate, share data and synchronize appropriately. This is implemented through the use of mutexes and conditional variables.

A mutex provides mutual exclusion for access to a particular data structure or resource shared among multiple threads. This is an absolute requirement for threads since there are times when only one thread should be accessing or changing a memory location, otherwise inconsistencies could arise. Mutexes provide the ability for a thread to acquire a lock before actually performing a sensitive operation, and then the ability to release this lock once the operation is complete.

Conditional variables work very closely with mutexes. Typically, it is desirable to acquire a shared resource which is protected by a mutex but only at such time that some pre-condition is satisfied. Therefore, the thread would wait (sleep) until such time as this pre-condition is met and then attempt to acquire access to the resource via a mutex. The conditional variable enables the thread to be signalled (woken up) by another thread once this pre-condition is satisfied and then to attempt to acquire the desired resource.

5.10.4 Audio Subsystem

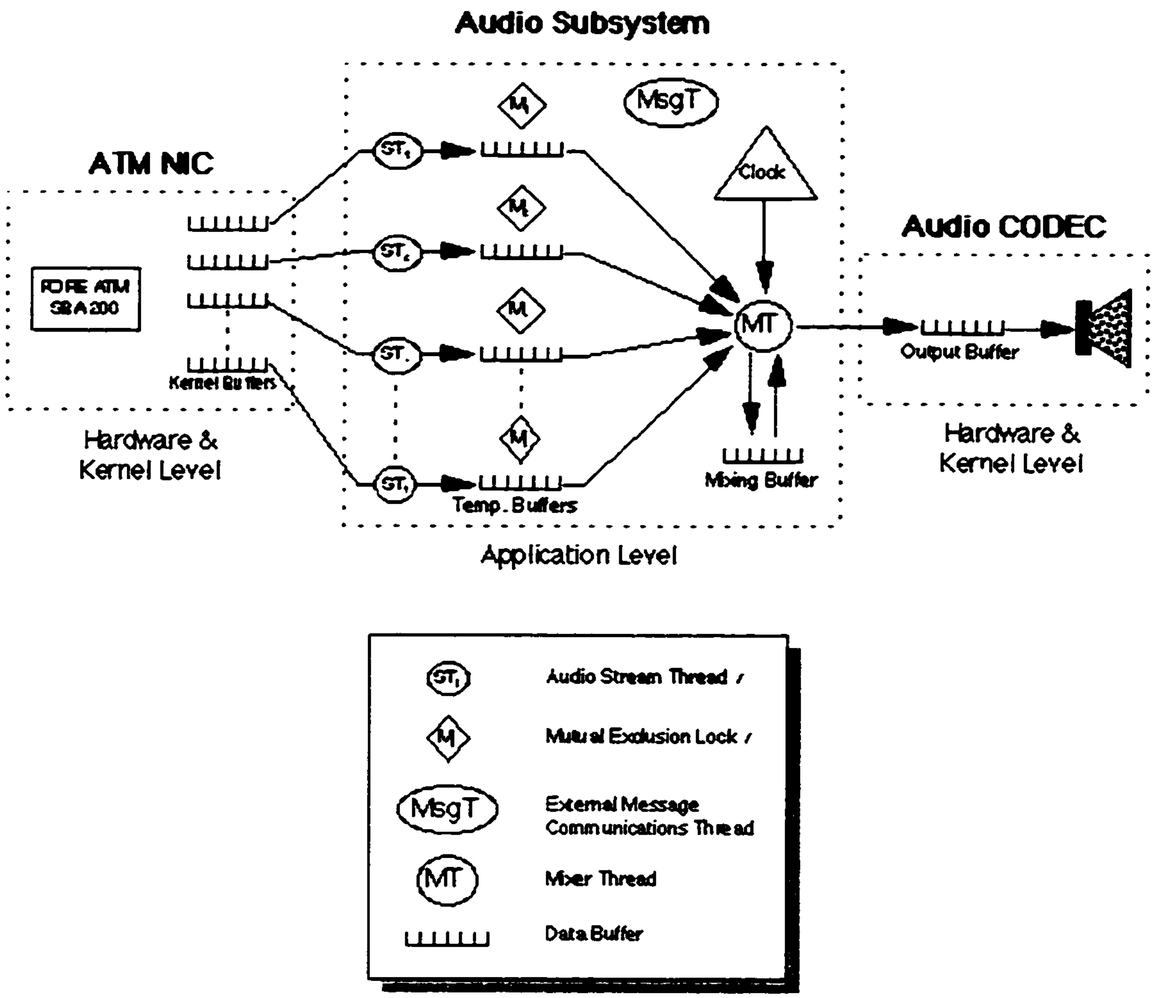
This section details the operation of the Audio Mixer component shown in Figure 5-2. The purpose here is to demonstrate a subsystem of the ATM RendezView prototype which exploits threads to perform concurrent activities. The Audio Mixer component is responsible for retrieving audio media units from *n* incoming ATM network connections, mixing these media units and presenting them to the audio output device in a timely fashion. The notion of "in a timely fashion" is key. The human ear is extremely sensitive to delay and jitter in audio, and so the playback process must be performed in a timely fashion with very little margin for skew [Steinmetz95].

For this reason, the mixing function cannot become dependant on the retrieval functions for timing (e.g. these two cannot be processed in a synchronous fashion). It is quite feasible that unpredictable delays are injected into a media stream before mixing occurs due to network/processing delay and jitter. These delays would cause the mixing process to have to wait until it received media units from all *n* incoming media streams. This could definitely cause buffer starvation and a degradation in the perceived audio quality from the output device. To circumvent this problem, an audio *Stream Thread* (ST) is designed to perform media unit retrieval and synchronization functions. and an audio *Mixer Thread* (MT) is designed to perform audio mixing and playback functions. Through this means of implementing two concurrent activities, audio mixing and playback can be tightly scheduled and at the same time made immune to any delay or jitter introduced by the retrieval and synchronization of audio data.

5.10.4.1 Structure

An overview of the audio subsystem architecture is shown in Figure 5-10. Each incoming media stream possesses its own ST, however only one MT exists. As mentioned previously, STs are responsible for retrieval and synchronization of media units and the MT is responsible for audio mixing and output. The STs are expected to perform their functions within the time constraint of the media unit sampling period, however some delays and missed deadlines are permitted. The MT however must rigidly respect its timing constraints and has the option of skipping media units which are delayed. All threads operate in a concurrent fashion thus enabling equal service to each media stream. Thread processing may be prioritized in a relative fashion allowing more important functions such as mixing to be favored.

Figure 5-10: Audio Subsystem Software Structure



From the figure, it may appear that this is not a highly scalable system due to the added processing demand placed on the MT and the requirement for additional STs and buffers as new media streams arrives. Since synchronization may be performed in one of three modes namely tight, loose and free-flow, processing load may be reduced by operating in free-flow for the majority of the time. Only periodically is a tight synchronization mechanism required. The additional buffers required are quite small, yet instrumental in the implementation of the synchronization mechanism. The MT is the most processor-intensive thread since it must constantly perform audio mixing. Even so, this processing load is insignificant compared to the load incurred by video processing and is not a concern.

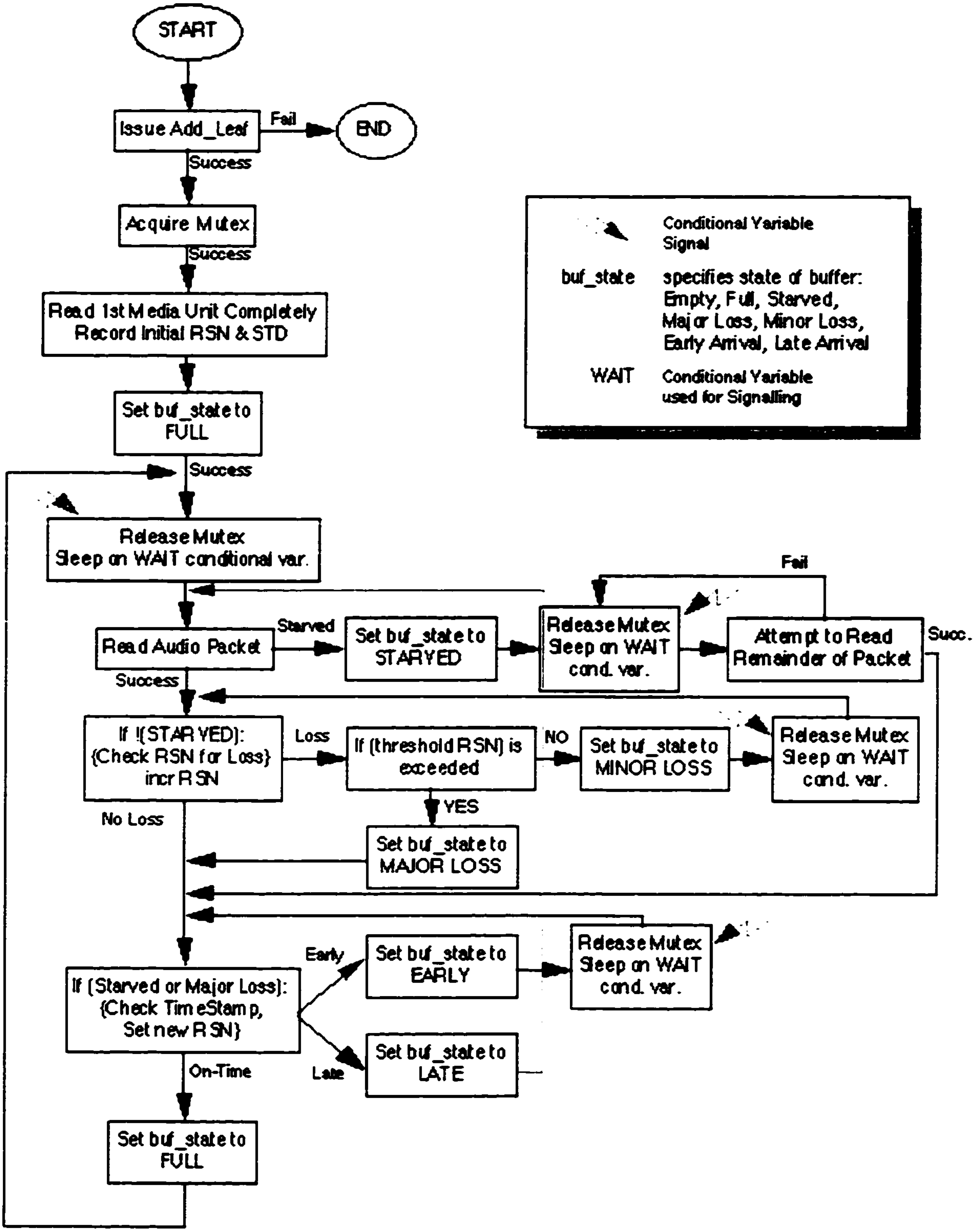
The advantages of this architecture are great. Based on the design of the STs and the MT, the subsystem is immune to any sort of impairments in media stream continuity whether they be due to loss, delays, or jitter. As long as these impairments do not last for extended periods of time, the audio subsystem will recover with little or no perceived errors. This should indeed be the case since any multimedia system should consist of highly performant components from end-to-end.

5.10.4.2 Stream Thread

The primary goal of the audio Stream Thread (ST) is to provide real-time handling of media units before mixing occurs. In the case where a media unit is either lost, delayed, early or in error, the stream thread can deal with this condition without causing any delay to the mixing process. If a media unit is not ready for mixing at the appropriate time, it is simply skipped and ignored by the MT. It is then up to the ST to re-synchronize the media stream. Obviously, a missed deadline should occur infrequently.

As shown in Figure 5-11, the ST performs the following tasks. It acquires its own mutex, reads a media unit from its network connection into a buffer, performs all necessary synchronization, sets its buffer state indicator to FULL, and releases its mutex. It then waits for a signal from the mixer thread before proceeding with the next media unit. A signal from the MT indicates that the media unit has been retrieved, mixed and the buffer is now empty. The ST can be designed to be as complex or simplistic as necessary. Multiple conditions can be detected and dealt with such as starved and overflowed buffers, transmission data loss, and early or late arriving media units. The ST's complexity is determined by factors such as the processing required by the various synchronization algorithms, the network reliability, and the media's perceived sensitivity to loss and delay. Ultimately, this is a complex design decision.

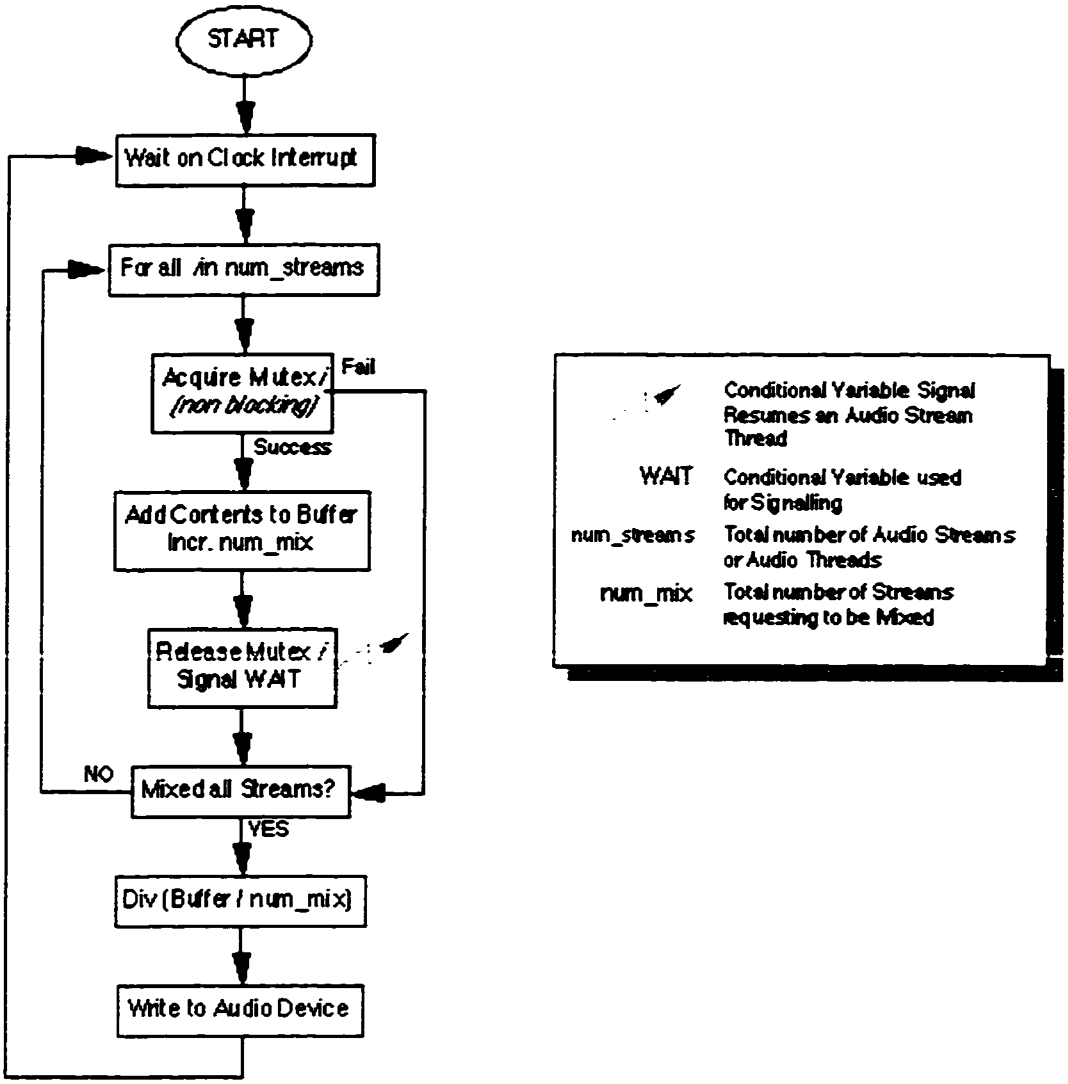
Figure 5-11: Audio Subsystem Stream Thread (ST)



5.10.4.3 Mixer Thread

The audio Mixer Thread (MT) is simplistic in nature compared to the ST but is the most compute-intensive due to its mixing requirements. As shown in Figure 5-12, it awaits a periodic clock signal before beginning a sequential process of *polling and mixing*. It polls the first ST to determine whether data is available for mixing. If so, it retrieves this data and signals the ST to continue and retrieve its next media unit. It then proceeds to the next ST and performs the same polling and mixing functions. Once it polls and mixes all STs, it outputs its own buffer to the audio device for playback and then sleeps awaiting the following clock signal. The goal here is to give the MT a high enough processing priority such that it completes a full cycle before the arrival of the next clock signal.

Figure 5-12: Audio Subsystem Mixer Thread (MT)



Audio is a jitter sensitive media type and so we do not want to delay the mixing process any more than necessary. Therefore, in the case that a ST is unable to meet its own deadline, the MT will simply skip the ST's media unit, thus averting any unnecessary delays. This creates a robust audio mixer which is insensitive to small losses of continuity within a particular media stream.

5.10.5 Video Subsystem

Details of the design and implementation of the ATM RendezView Video Subsystem can be found in [Pretty97]. This subsystem is implemented in a similar fashion to that of the audio subsystem. The network functionality of the video subsystem is identical in terms of ATM connection establishment and negotiation. TCP/IP is used by requesting leaves in order to be added to a point-to-multipoint connection.

The main differences involve the method of media synchronization employed. Video is designated as a slave stream and so it must retrieve its Synchronization Time Differential (STD) parameter from Audio Mixer component as discussed in section 5.7.3.5. Once this parameter is know by the Video Display component, it is able to independently perform its own intra-stream and inter-stream synchronization based on the local system clock and consequently display the video stream. Every single frame must be fully synchronized in a tight mode since there are no lower-level timing mechanisms available in the video hardware. The video display is the most resource intensive component of the ATM RendezView prototype since large amounts of data must be transferred between end-system components in a time-intensive fashion. These synchronization and display functions put the end-system to the greatest test which ultimately determines whether it is real-time multimedia capable.

5.11 Prototype Value

Prototypes and products should not be construed as having similar motivations. Prototypes are typically exploited as a "proof-of-concept" of a hypothesis or technical design. They are typically produced as rapidly as possible in an *ad-hoc* fashion in order to prove technical feasibility and justify further development. Products on the other hand, are developed at a later stage, where additional factors become important such as market demand, costs, and usability. It should be emphasized that ATM RendezView is very much a prototype and could never be marketed as a product.

The fundamental value provided by this prototype is found in the way it exploits readily available hardware and software, to produced a multiparty video conferencing environment which is of superior quality and performance to those products available today. It simply provides a strong argument towards the feasibility of realizing such scalable, high quality, multiparty video conferencing environments with today's technologies.

5.12 Performance Results

The subjective performance of the ATM RendezView prototype largely determined its success. Though an objective quantitative analysis could be adopted, the user's impression of the resulting presentation was deemed to be the ultimate performance metric. Two main components were assessed, namely media *quality* and *integrity*. Quality refers to the resolution and fidelity of the media as produced at the original source, whereas integrity refers to any impairments which have been imparted on the media stream as it propagates from sender to receiver.

The prototype offered excellent quality and integrity for the majority of its operation. Video could be set to a maximum quality of 640 by 480 pixels, 24 bit color depth and 30 frames per second, along with CD quality audio. There was no loss of integrity of these media streams upon display at the receiver. Audio was clear and smooth without any annoying clips, gaps, or loss of quality and video was fluid and realistic, comparable to NTSC video quality. Synchronization between video and corresponding audio was judged to be perfect. The prototype could be safely operated without any detectable loss of quality or integrity for conferences with up to five participants, after which the perceived stream integrity became compromised. This loss of integrity was an indication of the scalability limitations of the prototype, and was due to several factors:

- CODEC Performance

The first component of ATM RendezView to show its scalability limitations was the Parallax Motion-JPEG CODEC. This product has a maximum performance of the compression of one 640 by 480 pixel video stream at 30 frames per second. Typically, a video fidelity of 320 by 240 pixels at 15 frames per second would be sufficient for inter-personal communication. In this fashion, four to five video streams could be concurrently compressed/decompressed. Beyond this point, performance would degrade, manifested by lower video frame rates.

- ATM NIC Architecture Immaturity

The next component which showed its scalability limitations was the ATM Network Interface Card (NIC). Even though the NIC is based on OC-3 (155 Mbps) transmission rates, its interrupt-oriented interface with the workstation and its less than adequate hardware architecture placed significant limitations on its bi-directional throughput. This was especially problematic for the older NIC models which showed a best-case throughput of 12 Mbps. Newer revisions of this hardware have reduced this limitation significantly.

- Workstation Processor Limitations

The need for audio and video stream processing at the application level requires that large volumes of data be copied to and from workstation RAM. Though the Sparc 20's internal buss bandwidth is high enough to deal with this data, all of this memory copying requires significant CPU usage and incurs interrupt-related latencies. This sort of processing requirement is wasteful and is another limitation on the scalability of ATM RendezView.

- Real-Time Processing Implications

A real-time processing class within a best-effort scheduler can be invaluable especially when dealing with audio data which is extremely sensitive to processing jitter. An undesirable characteristic of such a processing class, however, is that whenever a real-time process requires the CPU, it preempts the current task being processed. Whenever a task experiences a context-switch (relinquishes or acquires the CPU), there is a certain amount of associated overhead. As CPU loading increases and context-switches become more frequent, this can rapidly spiral out of control. This occurs because the time associated with the context-switch overhead begins to approach the actual processing time required by the task. This is a significant scalability problem, however will only be exhibited during periods of high processor utilization.

A couple of design flaws relating to audio were noted which had a negative impact on the perceived performance of the prototype. The Sun Sparc workstations were designed to accept audio input from an "active" type microphone, similar to the one shipped with the workstation. ATM RendezView however employed a headset with a "passive" boom microphone. The effect was a considerably attenuated microphone signal which forced recording levels to be elevated excessively, thereby introducing additional noise and corruption. The resolution to the problem involved the design of a simple microphone preamp circuit, which elevated microphone input levels to the appropriate threshold. This remedied the problem quite effectively. A second design problem was related to the audio mixing process at the receiver. In the case that audio sources being mixed were not of similar levels, the softest audio signals would be masked by the louder ones. This would occur if music was being mixed with dialogue due to the added spectral content of the music. Ultimately, the mixer should take into account the differing spectral power of the signals to be mixed, in order that weaker signals not be lost. Assuming that all audio signals to be mixed were of a similar strength, the masking effect would not be observed.

Barring these noted deficiencies, the ATM RendezView prototype performed impressively, providing a communication medium with similar realism and interactivity to that of an actual face-to-face meeting.

Chapter 6: Future of Real-Time Multimedia Development

6.1 Introduction

The future of real-time multimedia is not so much in question, but instead the form that it will take. It is certain that multimedia applications will always challenge the limits of available technology by continually offering superior richness and quality of media information to the end-user. However, multimedia applications require a plethora of supporting technology to enable such functions as media digitization, compression, digital processing, transmission, synchronization and presentation. All these functions may be implemented in a variety of ways involving *de facto*, *de jure* or even proprietary approaches. As more functionality is desired within a multimedia application, the resulting complexity increases and as a result, the application becomes less interoperable. Because a lack of interoperability among multimedia applications is highly undesirable, especially those which are required for inter-personal communication, the global development and adoption of multimedia standards is a priority. The development of standards by standards bodies is the best alternative for guaranteeing the highest level of interoperability, however these introduce some additional problems which serve to retard the efficacy of these bodies in the first place. This chapter attempts to highlight the major barriers facing real-time multimedia development, especially as applied to multimedia communication systems and provides a view of future services which will evolve from research in the field of real-time multimedia.

6.2 Networking Barriers

Network services is the most important component of any real-time multimedia application due to its need for high bandwidth, low latency connectivity. Assuming that real-time multimedia implies per stream throughput in excess of a few megabits, end-to-end delay below 50 ms, and a bounded jitter characteristic, most networks employed today would be considered inadequate for supporting real-time multimedia. The reality is that real-time multimedia cannot function properly with the currently installed base of network technology, and so newer technologies must be exploited. A large number of solutions are available from which to choose, however there is usually an overriding factor such as cost, which predominantly influences the choice. Consequently, this choice is not always the best from a technology viewpoint. The networking barriers inhibiting real-time multimedia development can be broken down into three categories namely: the *network technology* used for real-time connectivity in the network's core, the *end-system access* technology used to connect the end-system to this core network, and the *Application Programming Interface (API)* which is the program's or application's interface to overall network services.

6.2.1 Network Technology

If the cost of networking technology was a non-issue, ATM would be the sure winner as the networking technology of choice, however this is not the case. ATM offers by far the greatest performance and suitability to real-time data transfer. Its bandwidth and user scalability, quality of service, traffic policing and management facilities, coupled with its seamless ability to service any type of traffic, promote it as unique among all network technologies. No other networking technology available today can claim superiority over ATM. The greatest forces facing ATM however, are its initial start-up costs and the delays which have been incurred in producing complete standards and bringing products to market. At the lowest level, ATM is an extremely complex technology, especially in the realm of traffic management and routing. This has led to delays in the completion of ATM Forum standards such as TM (Traffic Management), MPOA (Multi-Protocol Over ATM) and I-PNNI (Integrated Private Network Node Interface).

Such issues have opened doors to individual vendors to produce, what *should* be regarded as no more than interim solutions, to the need for low cost, high bandwidth, low latency network technology to support real-time multimedia. Certain large network vendors have produced proprietary solutions in order to give their own products a marketing edge, and other smaller vendors have formed alliances to achieve the same. The end result of all these efforts however, is commonly a solution which lacks the necessary ability to service real-time multimedia adequately. In the rush to bring a product to market, no global standardization effort is made (which compromises interoperability), characteristics necessary for *true* real-time transport are compromised, and the efforts being made by standards bodies such as the ATM Forum are undermined.

The plethora of networking technologies on the market today which have been offered as alternatives for high speed (broadband) communication can be roughly characterized as follows: they either offer a unique scheme for realizing the high speed physical transport of data, or they present a unique set of network layer protocols for achieving a QoS facility of some kind. The more effective schemes actually employ a combination of both to realize true real-time transport networks. As shown in Table 6-1, each has its own advantages and disadvantages, however the common thread that exists is that all solutions are competing fiercely for a dominant position in the lucrative broadband network communication's market.

Table 6-1 Competing Technologies in the Broadband Networking Market

Technology	Operational Overview	Pros	Cons
Cells-in-Frames (CIF) Developed by the Cornell Information Technologies (CIT) organization at Cornell University. CIT has formed the Cells-In-Frames Alliance to which many prominent networking companies belong. CIF v1.0 completed Oct '96. Standardization efforts proceed to add new features to v1.0. Chip design has already commenced at companies such as Connectware, however no products have been released.	ATM AAL PDUs are encapsulated within Ethernet MAC frames on a switched Ethernet segment between a WS and a directly attached CIF switch. The CIF switch then deencapsulates and forwards ATM cells to a directly attached ATM network. Only a s/w upgrade is needed on the attached WS. WS applications can use both native ATM and legacy Ethernet interfaces.	Provides a cost effective migration path from standard ethernet technology to native ATM to-the-desktop technology. Preserves current installed base of Ethernet NICs. Encourages the development of ATM aware applications, which can in turn promote the deployment of native ATM technology to-the-desktop. Offers ATM QoS facilities, however at the mercy of collisions on the Ethernet segment.	Requires additional processing at WS due to need for AAL functions, signalling and traffic shaping. Bandwidth is still limited by Ethernet technology. Requires investment in CIF switch which may not be recaptured. Relies heavily on the success of future ATM to-the-desktop technology which is somewhat questionable at this time. There is much competition with longer term technology solutions.

Table 6-1 Competing Technologies in the Broadband Networking Market

Technology	Operational Overview	Pros	Cons
Gigabit Ethernet Promoted initially by large vendors such as 3COM Corp. and Bay Networks Inc., it is now being developed by the Gigabit Ethernet Alliance (GEA). The GEA is a consortium of some 75 vendors. The standard is expected to be complete in early '97 and presented to the IEEE 802.3's (Ethernet) High-Speed Study Group for review. Standard incomplete (due March 1998)	Proposed as a backbone switch-to-switch and switch-to-server high speed connection medium. Operates as a 1 Gbps Ethernet with the same CSMA/CD access method and standard Ethernet packet format. One difference is the physical signalling scheme which is borrowed from ANSI's Fibre Channel specification. Physical transport medium is coaxial cable (25m max.) with plans for UTP5 implementation.	The anticipated cost is about half that of 622 Mbps (OC-12) ATM technology. Identical management requirements to legacy Ethernet and Switched Ethernet, along with ease of migration. Gigabit Ethernet can be designed as a shared medium (using CSMA/CD) or as a more efficient switched medium (without CSMA/CD). **Caveat**: the IEEE will not adopt the standard under its 802.3 subgroup if CSMA/CD is not employed.	There are no QoS facilities available, and no real proof that Gigabit Ethernet can handle delay sensitive traffic. Maximum network diameter is a critical problem since it is limited to about 25m over coax and even less over UTP5. Projected throughput in a typical shared Gigabit Ethernet will be limited to only 500-600 Mbps. The number of WSs which can actually benefit from such high bandwidth connectivity over a shared medium is in question.
IP Switching A proprietary technology developed by start-up Ipsilon Networks for routing IP over ATM. Products are available immediately from Ipsilon Networks. Ipsilon is looking for partners to put its switch software into LAN-to-ATM edge devices. It is also hoping to present its standard to the IETF for approval. IP Switch (US$46,050) can be configured with up to 16 OC-3 ports. A four slot IP Gateway (US$18,900) can be fitted with either Ethernet, FDDI or 100Base-T LAN modules.	Ipsilon has created a proprietary means of routing IP packets over ATM networks. Edge devices (IP Switch Gateways) and ATM network switches (IP Switches) compose the network. The network may operate in a *full-routing* or a *cell-streaming* mode. The former requires full IP routing at each intermediate IP Switch and is suited for short lived connections (e.g. DNS, SMTP, POP, SNMP). The latter involves a cut-through operation where ATM cells are switched end-to-end once a long-term IP stream has been identified and configured within the network (e.g. FTP, NFS, HTTP, multimedia).	Allows network managers to maintain separate IP subnets across an ATM backbone without the need for additional routers or hardware. (This is not possible with either ATM Forum's LANE or IETF's Classical IP over ATM standards) No need for network managers to learn the complexities of ATM which is a significant cost savings. Just plug and play! Long term connections can have a pseudo QoS associated with them, however it is based loosely on IP address, TCP/UDP port and RSVP messages (future). Uses four priority levels. Supports IP multicast.	IP is the only network protocol supported. Routing performance of bursty sources is only as fast as midrange routers. Cannot take advantage of QoS or signalling functions of ATM switches due to its proprietary approach. IP switching success is based upon the existence of long-lived traffic within the network. Much debate exists as to the actual ratio of long to short-term traffic in typical networks. Not a proven technology at this time. Performance projections are inconclusive, ranging from highly superior to sub-standard depending upon the source. More testing is required.

Table 6-1 Competing Technologies in the Broadband Networking Market

Technology	Operational Overview	Pros	Cons
Tag Switching Cisco System's version of *label switching* technology. Developed initially as a proprietary technology, but is now being presented to the IETF for consideration by its Multiprotocol Integrated Switch-Routing (MISR) working group. Similar proposals have also been made by IBM and Toshiba. Cisco is looking to license Cisco IOS Tag Switching technology to partners, in addition to releasing its Cisco 7500 series routers in the first half of 1997 for field trials of this new technology.	Tag Switching integrates the functions of routing and switching to achieve high performance packet forwarding in the backbone network. It assigns tags to multiprotocol frames for transport across packet-based or cell-based networks. Components of the system include: Tag Edge Routers (edge devices which perform network services and apply tags to packets at backbone ingress), Tag Switches (perform switching based on tags or Layer 3 functions) and a Tag Distribution Protocol (TDP) for the distribution of tag information among components.	Does not require any additional hardware in the edge device, simply a software upgrade to a router. This topology can reuse the existing ATM / frame relay network infrastructure. Traffic from multiple sources going to the same destination can share the same tag. This reduces a common scalability problem of *label explosion*. Short lived or dynamic traffic flows can exploit permanent network connections, thus avoiding network saturation by frequent connection setup calls. ATM's QoS facilities can be exploited for time sensitive data. No dependance on a particular physical or network layer technology.	Proprietary approach at this time. Not a proven technology due to lack of products. Investments in a Cisco solution to label switching will prove to be a problem when the IETF standardizes its version of label switching. Doesn't provide a hard real-time transport mechanism. QoS contracts allocated to a connection still cannot be explicitly indicated by the application. Requires some generic form of mapping from the higher layer network protocol (e.g. IP) to the lower level transport mechanism (e.g. ATM).

Table 6-1 Competing Technologies in the Broadband Networking Market

Technology	Operational Overview	Pros	Cons
Resource Reservation Protocol (RSVP), Real Time Protocol (RTP) Two isolated standards at different stages of development, produced by the Internet Engineering Task Force (IETF). The ultimate goal is for the two to interwork to enable real-time multimedia communications over IP based networks. RSVP has received final approval from the IETF. Some vendors have begun supporting it in products, however key issues such as access to QoS will not be addressed until work begins on the second version. It is still purely experimental at this time, however the full deployment of products should begin in 1998. RTP (RFC 1889) is expected to be finalized in early 1997.	RSVP is used by a host to request a specific quality of service (QoS) from the network on behalf of an application data flow. RSVP carries this request through the network, visiting each node along the way to the receiver. A visit to each node serves to chart a path that will be taken by actual data in that flow, however no resources are reserved at this time. Once this request reaches the receiver, a response is formulated which propagates back along the same path and in the process, reserves network resources. The receiver must periodically reaffirm its need for allocated network resources, or they will be removed. RTP is a transport protocol for real-time flows. It provides a standard packet header with a media specific timestamp, a payload format and sequence numbering. It also enables feedback on reception quality of flows as well as identification of receivers of a multicast flow. RSVP and RTP are designed to interwork. Whereas RSVP is concerned with allocating network (router) and host resources, RTP is responsible for providing useful media data flow, synchronization and statistics on reception quality.	The sheer mass of existing IP based equipment which is capable of being augmented with RSVP and RTP support will alone, guarantee its success. RSVP and RTP have much industry support from prominent vendors such as Microsoft Corp., Intel Corp., Cisco Systems Inc., and Bay Networks Inc. These protocols are scalable to very large groups. Receivers can make different requests for capacity from the same source. RSVP and RTP are transport medium independent allowing networking over various types of transport networks. Along with other protocols, RSVP and RTP form an integral component of the IETF's Integrated Services Architecture (ISA) which provides a complete solution for multimedia conferencing.	RSVP is still largely unproven. Desktops will need to add RSVP and RTP to their stacks, and routers will require s/w and possibly h/w upgrades. Applications will have to be upgraded to become RSVP aware. RSVP cannot guarantee that reserved resources will actually be available, it is merely a method of prioritizing flows (soft real-time). Much effort is required to determine which applications will gain access to priority processing by the network, and how RSVP's QoS facilities will be mapped to specific transport technologies such as ATM, frame relay, FDDI and SONET. Proponents of ATM feel that standardization of "soft" real-time technologies such as RSVP serve only to undermine the efforts of the ATM Forum. ATM contains all the necessary mechanisms for providing real-time communications without the overhead of higher layer protocols such as IP, RTP or RSVP.

Table 6-1 Competing Technologies in the Broadband Networking Market

Technology	Operational Overview	Pros	Cons
Asynchronous Transfer Mode (ATM) ATM encompasses a variety of standards being produced by the ATM Forum, all at various stages of completion. Standards address signalling (UNI), virtual LAN connectivity (MPOA, LANE), traffic management (TM), and routing (PNNI, I-PNNI) to name just a few. Many prominent vendors have been selling ATM equipment for the past five years, however ATM technology as a whole is considered immature due to its lack of complete standards for routing and traffic management.	ATM was conceived as a real-time transport technology for multimedia and time-sensitive data. As such it implements strict protocols to allocate and monitor network resources such that multimedia streams receive the essential services they require. ATM is a connection oriented technology enabling point-to-point, point-to-multipoint and multipoint-to-point connection topologies. Each connection has a quality of service (QoS) contract associated with it, mutually negotiated and agreed upon by the sender and the network. The ATM connection process involves a setup phase (where the desired connectivity is indicated by the sender), a connection admission control (CAC) phase (where an end-to-end route is determined based on available resources and/or a negotiated service contract) and an ongoing usage control phase (where traffic flows are monitored to guarantee that they do not violate their agreed traffic contract).	The ATM Forum has the strong industry support of hundreds of companies. ATM products are available immediately and are becoming affordable. ATM's quality of service and traffic management facilities offer the ultimate solution for the transmission of real-time, high bandwidth data. Little debate exists as to its superiority in this area. ATM technology is highly scalable in terms of bandwidth (current maximum is 622 Mbps with no theoretical limit), users (ATM has the concept of multicast groups which are developed from spanning trees) and services (ATM is traffic independent and as such extensible to future services such as VoD). ATM provides for a solid investment due to its ability to perform both as a near-term solution (data and LAN connectivity) and a long term solution (multimedia and real-time applications).	Expensive technology at this time due to incomplete standards and relative immaturity of technology. Standards efforts by the ATM Forum are taking too long and producing overly complex specifications. Other technologies with a relatively short time-to-market are undermining the ATM Forum's efforts. Without a simple migration path, ATM technology commonly requires complete networking infrastructure upgrades. ATM technology requires additional costs for retraining staff.

There is significant pressure for network vendors to produce solutions for broadband communications. However in the rush to release products, the real-time facilities of such solutions, namely the quality of service and traffic management mechanisms are commonly not considered. The prime motivators for producing these products are usually low cost, backwards compatibility and excessive bandwidth (which is commonly mistaken to imply real-time). These products are usually attractive to customers, however they still do not offer hard real-time transport services. Media types such as voice and video do not typically perform well over these networks due to their inherent need for stringent real-time processing. Unfortunately, elaborate marketing campaigns serve to mislead customers into purchasing technologies which can only offer short-term solutions to their real-time multimedia networking problems.

6.2.2 End-System Access

End-system access technology enables connectivity between an end-user and a core backbone network. This technology is required in the case where end-users are geographically dispersed on an individual basis and consequently, the installation and operation of a LAN is not cost effective. An example scenario is when corporate employees decide to work at home and require access to company networks and resources. In the past, corporate work-at-home programs were not the prime motivation for the development of higher speed access technologies from the home, in reality it was the financial attractiveness for ISPs to offer higher speed internet surfing capabilities than was currently available via 28.8 modems. However, with the financial advantages of work-at-home programs available to corporations, and the convenience offered to employees, these work-at-home programs are projected to be implemented on a very large scale.

A significant difference exists however between the requirements of the home user wanting to "surf" the internet and the corporate employee wishing to be productive. The home user is simply interested in browsing the internet at an acceptable speed, and is quite willing to accept low speed transfers and long delays if this means a significant cost savings. The corporate user, in contrast, must be productive. They typically require access to company databases, wish to exchange information with colleagues and most important, wish to attend meetings (virtually) using real-time collaboration tools. These may include audio and video conferencing, document sharing and collaborative working. All of these tools imply the need for real-time, high bandwidth and low latency remote access from the home to the corporate network.

As with the core network technology discussed in section 6.2.1, the end-system access technology must also enable real-time, high bandwidth transfers to occur. To reiterate, minimum network requirements are as follows: a per stream throughput in excess of a few megabits, end-to-end delay below 50 ms, and a bounded jitter characteristic. Unfortunately, there has been little need for this type of access in the past, so no practical technology solutions exist at this time. This is a significant barrier to the development of real-time multimedia! Fortunately, this problem is being addressed and a fair number of proposals and new products are being actively developed. Table 6-2 highlights the main access technologies being produced, their specifications and limitations.

Table 6-2 Competing Technologies in the Network Access Market

Technology	Upload / Download Bandwidth	Costs (US$) Initial / Monthly	Pros	Cons
Dial-in Modems	14-28 Kbps Symmetrical	$100 / $30	Extremely affordable, widely implemented and supported. Exploits the RBOC's installed base of copper wire.	Insufficient bandwidth for real-time multimedia.
ISDN (BRI)	56-128 Kbps Symmetrical	$700 / $50	Relatively mature product, with growing availability of modems and adapter cards. Exploits the RBOC's installed base of copper wire.	Difficult to configure and not completely available from all RBOCs and ISPs. Expensive.
Leased Lines	64 Kbps - 1.5 Mbps Symmetrical	$2,500 / $1,000	Proven technology. Dedicated bandwidth pipe.	Expensive. Insufficient bandwidth to justify the cost, especially with ADSL just around the corner.
Cable Modems	99 Kbps - 10 Mbps 500 Kbps - 30Mbps (Shared)	$1,200 / $50	Much more cost-effective solution than ISDN for bandwidth. Exploits the cable company's installed base of coaxial cable	Doesn't scale well since bandwidth is shared among multiple users. No real-time facilities. No guarantee on bandwidth as in ISDN or ADSL. A new technology which is still dealing with incompatibilities of multi-vendor equipment.

Table 6-2 Competing Technologies in the Network Access Market

Technology	Upload / Download Bandwidth	Costs (US$) Initial / Monthly	Pros	Cons
ADSL	16 - 500 Kbps 1.5 - 9 Mbps	N/A	Doesn't require an upgrade to the RBOC's central office switch. Could enable real-time multimedia assuming ATM is used. ADSL can be deployed one customer at a time due to its point-to-point nature.	Range is limited at the higher speeds (9 Mbps link must be within 9000 ft.). Estimates of initial pricing puts a cost of US$2000 per modem. This should decrease with volume usage.
Satellite	None / 400 Kbps	$1,100 / $50	A form of wireless communication, suitable for remote areas, or uni-directional communications.	Uni-directional broadcast technology which requires a modem for the return path. Very low bandwidth technology, not suitable for real-time multimedia.
Wireless Cable	27 Mbps Symmetrical	$500 / $50	A new technology providing a relatively low cost infrastructure offering quick entry into the mass market.	A non-standardized technology. Too early to tell how suitable it will be for real-time multimedia.

The broadband network access market is expected to be extremely lucrative and so there is a lot of interest on the part of various companies to become big players. Telephone and cable operating companies are especially well positioned due to their installed infrastructures of copper twisted pair wire and coaxial cable. Many of the access technologies shown in the table above were designed with a specific cabling framework in mind. Still, there are other players who are betting their luck on the success of wireless or satellite based technologies.There are no guaranteed winners, however the Regional Bell Operating Companies (RBOCs) seem well positioned to become dominant players. For real-time multimedia development to proceed in this environment, stable, interoperable access solutions will have to be adopted and standardized which offer both high bandwidth and low latency connections. Ultimately, ATM technology must be supported by the chosen access technology.

6.2.3 Application Programming Interface

The Application Programming Interface (API) is the application developer's access interface to underlying network services. It is essentially a library of complex routines which may be exploited by the programmer to perform such tasks as network signalling, connection management and data transfer via a high-level interface. The problem with APIs (which are typically provided by the vendor of the underlying hardware) is that the application must be re-written to exploit a particular vendor's API. This is a significant issue for application developers who must make a solid decision as to which networking technology their applications will support. It is widely accepted that ATM is by far the best solution for real-time multimedia applications, however the technology does not extend to the desktop at this time, and it may never. IP is a widely exploited network protocol, however will it service the requirements of real-time multimedia applications adequately? These types of issues only serve to complicate matters.

Fortunately, there is a light at the end of the tunnel. The Winsock Forum has recently introduced its new Winsock v2.0 API. Winsock v2.0 is the Microsoft Windows (Windows 3.1, Windows95, WindowsNT) platform's protocol independent networking interface. In addition to being backwards compatible with previous Winsock compliant applications, the new Winsock offers multiprotocol support, quality of service, a layered provider architecture and numerous performance enhancements to support real-time multimedia. Judging by Microsoft Corporation's market presence, this API will likely become the industry standard for ATM aware applications, supplanting any proprietary or ATM Forum generated API specifications. Luckily for ATM however, this will give native ATM to-the-desktop applications a fighting chance since developers will not completely disregard ATM due to its current immaturity and costly nature. They will be able to develop ISA (IPng, RSVP, RTP) aware applications in the near term, with the option of a smooth migration to ATM in the long term.

6.3 Computing Technology Barriers

Computing technology plays an integral roll in the development of real-time multimedia. Computers offer a rich environment for prototyping and development without which we would not have the rapid and cost effective deployment of new technologies that we have today. Computing technology vendors however must be cognizant of the intended use of their products, and must incorporate this knowledge into their designs.

A prime example of where this has caused some problems is with real-time multimedia. Computer vendors have provided elaborate interfaces for both audio and video I/O in the form of "add-on" capture cards. Users wishing to perform real-time processing of multimedia streams find that the computer's operating system is ill-equipped to deal with real-time data. Typically, the operating system operates in a best-effort fashion, which is inadequate for processing time-sensitive data. This may have been sufficient in the past when the only requirement of multimedia I/O was for recording or playback facilities. In these cases, large buffers were exploited to absorb any latency in the computer's processing cycle. However, future multimedia applications will be employed for video conferencing, where audio and video buffering delays are unacceptable. Clearly, computer vendors need to realize the need for deterministic, real-time operating systems if multimedia development is to proceed on a large scale.

Constantly evolving networking standards are yet another reason why computing standards bodies should closely monitor networking standards and trends. The recent development of the IETF's ISA will clearly change the way multimedia applications are developed. However, these new networking standards require that computer protocol stacks, drivers and often hardware modules be upgraded in order that application developers leverage the advantages of these new networking technologies. Any complacency on the part of computing technology bodies to recognize these needs will only serve to retard the development of real-time multimedia applications.

6.4 Standards Versus Cost Barriers

Standards are widely accepted as the ideal method for producing new technologies. Standards promote portability and interoperability, and can potentially produce a superior technology due to the collaboration of multiple companies and institutions. However, the development of standards can take excessive amounts of time due to the need to satisfy a majority of the member parties participating in the standards process. As they say, "time is money" and this is often a prime motivation for a vendor to produce its own proprietary, non-interoperable solution to a required technology.

An excellent example of such a situation is the ATM standardization process which is taking far too long to complete. ATM standards are being developed with such complexity and delay that vendors such as Cisco Systems and Ipsilon Networks are producing alternative technologies (Tag Switching and IP Switching respectively) with the hope that their solution will acquire the necessary market segment to supplant ATM and any other competing technologies. In addition to being proprietary in nature, these solutions in their rush to be released, compromise many of the values provided by ATM technology; namely its quality of service and traffic management facilities necessary for real-time multimedia applications.

Costs can become a significant barrier to real-time multimedia development especially when dealing with established technologies. IP as an example, was developed as a proof-of-concept technology, and adopted as a *de facto* standard due to its suitability to data communications. However, our data communications needs have changed over the past few decades, especially with the advent of real-time multimedia, however our networking technology has not. Instead of replacing the old technology with an appropriately designed new technology, costs necessitate that we support the existing, installed base of old IP technology. As such, we are limited to performing upgrades and improvements to the older technologies, which proves to be less than ideal in many situations. In order for real-time development to proceed, vendors and consumers alike must realize that a migration path is required to the newer technologies, and that this will incur some costs.

6.5 A View of the Future

There is no question that real-time multimedia is here to stay. The lure of vibrant, colorful and appealing media as a replacement for plain text communication is difficult to resist. Currently, multimedia is at a very young stage of its evolution and does not represent its full potential. The future will yield media quality of extremely high fidelity, uninhibited by its digital transport and communications infrastructure. HDTV grade video and CD quality audio will be widespread, and new technologies such as three dimensional virtual reality will challenge our computing technologies.

The maturation of real-time multimedia technology to this state will require a number of evolutionary steps. First, standards development of real-time networking technology will have to stabilize. At this point, wide-scale ASIC development required for networking components (adapters, switches and edge devices) will begin. This will drive down the cost of equipment and position network bandwidth as a commodity. Once standards for audio and video compression are adopted on an industry-wide scale, ASIC development will begin in this area as well. Performant hardware "add-in" components will evolve to service the unique processing needs of real-time multimedia.

At this point, computing and networking infrastructures should be such that new real-time multimedia services can be easily deployed. The primary markets for services will be corporate initially, in order to absorb the higher pricing of new technologies, and subsequently the consumer market. The corporate market will require *productivity* tools which include audio and video group conferencing and collaborative work tools such as application sharing and cooperative design tools. The consumer market will look primarily for *entertainment* and *service* tools which include video-on-demand, tele-education, multi-person gaming, home shopping, and home banking. The home computer will ultimately become a powerful entity, the primary enabling platform within the household, replacing the need for telephones, televisions and VCRs. These computers will exist throughout the home, as televisions do today. Real-time multimedia is at a crossroad in its life, and what the future holds should be both awesome and exciting.

Chapter 7: Summary and Future Work

7.1 Summary

Multimedia is rapidly becoming a popular field due to the potential richness and dimension offered by such technology. Admittedly, the term *multimedia* itself is largely misunderstood, especially as a result of its overuse within marketing campaigns. At its lowest level, it implies the exploitation of two or more forms of media for the purpose of information representation. This may involve visual, audible, or any other form of sensory information which may be perceived by a human being. Since we are dealing with a human's sensory perception, a dependency on temporal constraints is also implied. Timing, to varying degrees, is a very important component of human perception, independent of the senses involved. This time critical characteristic of media information is considered fundamental and should be addressed by multimedia based systems. Unfortunately, this is not typically the case, and it is where this thesis differs in its assumptions about multimedia. To clarify, this thesis commonly refers to multimedia as *real-time multimedia* in order to emphasize this time dependence. This is not to say that a static image on a computer screen is not multimedia, however this form does not pose any significant technical challenge and is of no interest. The form of multimedia which does pose significant technical challenge is real-time multimedia, which is characterized by *the continuous flow of high bandwidth, time critical, digital information.* The sheer volume of data combined with the constant need to meet time critical deadlines are the two components which contribute to its complexity.

In the past, real-time multimedia systems have been largely unrealizable due to the immaturity of computing and networking technologies. However, within the past few years, this has changed dramatically. The exponential increase in performance of these technologies has given new hope to the development of real-time multimedia applications. As such, the motivation of this thesis was to determine whether, *given our current technological state and availability of hardware and software products, the realization of a true real-time multimedia system was feasible*. The approach taken was to develop a high quality, real-time, multiparty, audio/video conferencing application. This application was deemed to pose a significant challenge due to its need for high quality (high bandwidth), real-time performance (intra-stream and inter-stream requirements for proper human perception), and scalability (inclusion of multiple participants all requiring the same high quality).

The resulting prototype, ATM RendezView was a success. It enabled many users to join a high quality audio/video based conference to see and hear everybody in a controlled and synchronized fashion. Each media stream could be controlled dynamically, allowing a conference member to select any combination of desired media streams. Audio quality could be selected to range from telephone quality to CD quality, and video could be configured up to a maximum quality of 640 by 480 pixels, 24 bit colour, at 30 frames per second. This superior media quality is what set ATM RendezView apart from most other similar prototypes, and also what challenged the application's real-time abilities the most.

The prototype employs Asynchronous Transfer Mode (ATM) natively as its real-time networking technology for several reasons: ATM can offer a Quality of Service (QoS) contract to be associated with each media stream, each media stream can be setup and managed dynamically using ATM's concept of Switched Virtual Circuits (SVCs), and most importantly, ATM offers a multicasting facility which greatly reduced the imposed load on end-systems. Compute-intensive tasks were executed using a UNIX based operating system with real-time processing facilities such as threads and a real-time scheduling class.

Through the success of this prototype, several conclusions can be made. First, the degree of technological advance, especially as applied to computing and networking technology available today, is such that the development of real-time multimedia applications are feasible. Second, ATM technology's hype promoting it as a real-time transport infrastructure for multimedia streams is indeed accurate. ATM technology was instrumental in the success of the prototype, especially as exploited for its multicasting, bandwidth and QoS features. Third, the distributed nature of the architecture employed in this prototype challenges the age-old "Px64" methods of implementing video conferencing environments using expensive, proprietary, non-scalable hardware products. Fourth, the efforts involved in developing this prototype have uncovered some significant barriers facing real-time multimedia development, especially in the arena of standards and interoperability. However, with a concerted effort, most of these barriers can be surmounted.

7.2 Future Work

ATM RendezView, however successful in its goals, offers many streams for future work. One option is a productization of the prototype. This would involve some research into the financial and technical issues involved in porting the prototype to a more mainstream operating environment such as Windows95. It would seem that hardware availability, such as ATM NICs and video CODECs would not be a problem, however the operating system could cause some problems. ATM RendezView requires high bandwidth I/O facilities as well as a real-time processing, which such an operating system may not be able to provide. A thorough market assessment would be required to determine the competitive advantages which could be exploited and the financial viability of such a venture.

A second stream of investigation could concentrate on the networking technology employed. Although ATM is the most ideal technology for this type of application, it is not the only possible solution. ATM technology has been widely deployed as a backbone network infrastructure, however it is less than popular in the LAN as to-the-desktop connectivity. IP-based technologies compose a majority of the installed base of LAN technologies and so their suitability to real-time applications should be investigated. The IETF has recently drafted its Integrated Services Architecture (ISA) which includes such protocols as RSVP, RTP and IPng for providing "soft" real-time services in the LAN. By exploiting ATM technology in the backbone, this type of IP over ATM solution could be adequate for servicing real-time data, thus incurring significant cost savings. This requires much investigation.

A third stream of investigation involves the real-time scalability of the computing end-system. All of the operating systems employed today in personal computers and UNIX based end-systems employ a best-effort scheduling entity for arranging the processing of tasks. Unfortunately, this provides a non-deterministic processing environment which is not suitable for true real-time applications, especially on a large scale. Two solutions are available to resolve this problem. The first being a transition to a real-time kernel. This would enable tasks to obtain quality of service contracts from the processor(s) involved. Unfortunately, it would make no sense to design for this type of system if it was not first widely deployed, as are Microsoft's Windows95 and Macintosh's System 7. An alternate solution would involve a hardware based approach. Several "add-on" cards could be employed each fulfilling a unique task such as video coding, audio processing, and network I/O processing. These would be placed on the host computer's common buss, however they would have a separate, external, high bandwidth real-time buss dedicated for their use. In this fashion, the host computer would only be required for setup and on-going management, and would not be weighed down with the excess processing of high bandwidth data. With the developments in ASIC technology, this option is very attractive from both a cost and performance perspective.

The field of real-time multimedia is wide open in terms of research topics and business opportunities. All that is required is some technical knowledge, a great imagination, and a desire to innovate.

List of Terms

A/D	Analog to Digital Converter
AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ABT	ATM Block Transfer
ACK	Acknowledgment
ADSL	Asymmetrical Digital Subscriber Line
AESA	ATM End-System Address
ANSI	American National Standards Institute
API	Application Programming Interface
ARPANET	Advanced Research Project Agency Network
ASIC	Application Specific Integrated Circuit
ATM	Asynchronous Transfer Mode
bps	Bits Per Second
CAC	Connection Admission Control
CAD	Computer Assisted Drafting
CBR	Constant Bit Rate
CBT	Computer Based Training
CD	Compact Disc
CD ROM	CD Read Only Memory

CDV Cell Delay Variation

CERN European Laboratory for Particle Physics

CISC Complex Instruction Set Computer

CLP Cell Loss Priority

CODEC Coder/Decoder

CPCS Common Part Convergence Sublayer

CPU Central Processing Unit

CRC Cyclic Redundancy Code

CSMA/CD Carrier Sense Multiple Access/Collision Detection

Concurrency Concurrency exists when at least two threads are in progress at the same time. In a uniprocessor system, the processor can switch execution resources between threads, resulting in concurrent execution.

D/A Digital to Analog Converter

DAVIC Digital Audio Visual Council

DBR Deterministic Bit Rate

DNS Domain Name Service

FDDI Fiber Distributed Data Interface

FTP File Transfer Protocol

GCID Global Call ID: UNI v4.0 defines a point-to-multipoint connection as being uniquely identified by a GCID. The GCID is a concatenation of the root's AESA and the negotiated LIJ call id.

GFC Generic Flow Control

GUI Graphical User Interface

HDTV High Definition Television

HEC Header Error Control

HMI Human Machine Interface

HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
I-PNNI	Integrated PNNI
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IMA	Interactive Multimedia Association
IMTC	International Multimedia Teleconferencing Consortium
IMUX	Inverse Multiplexer
IP	Internet Protocol
IPC	Inter-Process Communication
ISA	Integrated Services Architecture
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU-TS	International Telecommunication Union - Telecommunication Standardization
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LANE	LAN Emulation
LIJ	Leaf Initiated Join
MAN	Municipal Area Network
MCU	Multicast Control Unit
MIDI	Musical Instrument Digital Interface
MJPEG	Motion JPEG

MPEG	Motion Picture Experts Group
MPOA	Multi-Protocol Over ATM
NACK	No Acknowledgment
NCSA	National Center for Supercomputing Applications
NIC	Network Interface Card
NFS	Network File System
NNI	Network Node Interface
NPC	Network Parameter Control
NTSC	National Television Standards Committee
OAM	Operations, Administration and Management
OS	Operating System
PCR	Peak Cell Rate
PNNI	Private Network Node Interface
POSIX	Portable Operating System Interface: an evolving growing document that is being produced by the IEEE and standardized by ANSI and ISO.
POSIX.1	First POSIX standard which specifies many basic OS calls that UNIX programmers have come to expect.
POSIX.4	An addition to the POSIX.1 standard which adds real-time functionality such as shared memory, priority scheduling, message queues and synchronized I/O.
POSIX.4a	A threads extension specification to the POSIX.4 standard.
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephone Network
PT	Payload Type
PVC	Permanent Virtual Connection

Parallelism	Parallelism arises when at least two threads are executing simultaneously. When a process has as many threads to be executed and the system contains multiple physical processors on which threads may execute, parallelism is possible.
QoS	Quality of Service
RBOC	Regional Bell Operating Company
RISC	Reduced Instruction Set Computer
RFC	Request for Comments
RSVP	Resource Reservation Protocol
RTP	Real-Time Protocol
SAR	Segmentation and Reassembly
SBR	Statistical Bit Rate
SCR	Sustained Cell Rate
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SPANS	Simple Protocol for ATM Network Signalling
SSCS	Service Specific Convergence Sublayer
SVC	Switched Virtual Connection
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
VCC	Virtual Channel Connection
TM	Traffic Management
UDP	User Datagram Protocol
UNI	User Network Interface
UPC	Usage Parameter Control

UTP5	Category 5 - Unshielded Twisted Pair
VBR	Variable Bit Rate
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VCL	Virtual Channel Link
VPI	Virtual Path Identifier
VR	Virtual Reality
VoD	Video on Demand
WAN	Wide Area Network
WS	Workstation
WWW	World Wide Web

Bibliography

[Ahmad96] Dr. Khalid Ahmad, *Asynchronous Transfer Mode: Technology, Standards and Networks*, BNR/NT Technical Document, Northern Telecom, October 1996

[ATMForum94] ATM Forum Technical Committee, *ATM User-Network Interface Specification*, Version 3.1, The ATM Forum, Sept. 1994

[Bot93] Sonia D. Bot, *BNR/NT Graphical User Interface Guidelines*, Release 1.00, BNR/NT Technical Document, Northern Telecom, March 1993

[Bulterman91] D.C.A. Bulterman, R. van Liere, *Multimedia Synchronization and UNIX*, Proc. of 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany, Nov. 1991

[Chen95] Xiaobao Chen, *End-to-End Synchronization Control Support for Multiservice Applications*, Computer laboratory, University of Cambridge, Nov 1995

[Chen96] Herng-Yow Chen, Ja-Ling Wu, *MultiSync: A Synchronization Model for Multimedia Systems*, IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, Jan. 1996

[Cockcroft95] Adrian Cockcroft, *Sun Performance and Tuning*, SunSoft Press/Prentice Hall, 1995, 0-13-149642-5

[Deitel90] Harvey M. Deitel, *Operating Systems*, 2nd edition, Addison-Wesley Publishing Company, 1990, 0-201-18038-3

[Dowd93] Kevin Dowd, *High Performance Computing: RISC Architectures, Optimizations, & Benchmarks*, O'Reilly & Associates, Inc., 1993, 1-56592-032-5

[EET96] Electronic Engineering Times, *Desktop Videoconferencing is on the Move*, CMP Publications Inc., p63, April 1996

[FORE94] FORE Systems Inc., *ATM Application Programming Interface*, Release 3.0, ForeThought Partners Adapter Documentation, MANU0031 - Rev. B, Mar. 1994

[FORE95a] FORE Systems Inc., *SPANS UNI: Simple Protocol for ATM Network Signalling*, Release 3.0, ForeThought Partners Adapter Documentation, MANU0031 - Rev. B, Mar. 1995

[FORE95b] FORE Systems Inc., *An API for FORE Signalling and Data Transfer (SD API)*, Version 1.0, ForeThought Partners Adapter Documentation, Oct. 30, 1995

[Gallmeister95] Bill O. Gallmeister, *POSIX.4 Programming for the Real World*, O'Reilly & Associates, Inc., 1995, 1-56592-074-0

[Gibbs96] Simon Gibbs, *Index to Multimedia Information Sources*, German National Research Centre for Information Technology, http://viswiz.gmd.de/MultimediaInfo/, December 1996

[Guha95] Aloke Guha, Allalaghatta Pavan, Jonathan Liu, Ajay Rastogi, Todd Steeves, *Supporting Real-Time and Multimedia Applications on the Mercuri Testbed*, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 4, May 1995

[Hardenber96] Jan C. Hardenbergh, *VRML Frequently Asked Questions*, Version 2.0, http://vag.vrml.org/VRML_FAQ.html, February 1996

[Hewitt96] Kathy Hewitt, *Desktop Video Conferencing Products*, DT5-Desktop Video Conferencing Survey, http://www3.ncsu.edu/dox/video/survey.html, September 1996

[InfoWeek96] Information Week, *Online Training Poised for a Boost*, EduPage - Information Week, p32, November 1996

[Laird94] Sarah Laird, Joanne Youngman, *Video Development Environment*, Parallax Graphics Inc., Part No. 917-026526 Rev C, 1994

[Lewis96] Bil Lewis, Daniel J. Berg, *Threads Primer: A Guide to Multithreaded Programming*, SunSoft Press, 1996

[Minoli94] D. Minoli, R. Keinath, *Distributed Multimedia Through Broadband Communications Services*, Artech House, Inc., 1994, 0-89006-689-2

[Nicolaou90] Cosmos Nicolaou, *An Architecture for Real-Time Multimedia Communication Systems*, IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, Apr. 1990

[OSF93] Open Software Foundation, *OSF/Motif Style Guide*, Release 1.2, PTR Prentice-Hall, Inc., 1993, 0-13-643123-2

[Parallax94] Parallax Graphics Inc., *XVideo Technical Overview*, Release 1.1, Parallax Graphics Inc., 1994

[Pretty97] Russell Pretty, *ATM RendezView: Distributed Architecture for Multipoint Conferencing*, M.A.Sc. Thesis, Department of Electrical Engineering, University of Ottawa, Canada, Feb 1997

[Quick96] Gregory Quick, Tom Maresca, *Videoconferencing Seen as Hot Category at Retail*, Computer Retail Week, v6, n137, p35, June 1996

[SRI93] SRI International, *Digital Video in Advanced Multimedia*, Videophones & Videoconferencing Equipment, SRI Consulting International, http://future.sri.com:8080/, August 1993

[Samudra95] Pradeep Samudra, *Draft of UNI Signaling 4.0*, ATM Forum: Technical Committee, Signaling Subworking Group, 94-1018R5, Aug. 1995

[Siglar96] Jamie Siglar, *Multimedia Authoring Systems FAQ*, Version 1.4.3, http://www.tiac.net/users/jasiglar/MMASFAQ.HTML, March 1996

[Stallings94] William Stallings, *Data and Computer Communications*, 4th edition, Macmillan Publishing Company, 1994, 0-02-415441-5

[Steinmetz95] Ralf Steinmetz, *Analyzing the Multimedia Operating System*, IEEE Multimedia, Spring 1995, Vol. 2, No. 1, pp. 68-84

[Steinmetz96] Ralf Steinmetz, Gerold Blakowski, *A Media Synchronization Survey: Reference Model, Specification, and Case Studies*, IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, Jan. 1996

[Stevens90] W. Richard Stevens, *UNIX Network Programming*, Prentice Hall Inc., 1990, 0-13-949876-1

[Stevens92] W. Richard Stevens, *Advanced Programming in the UNIX Environment*, Addison-Wesley Publishing Co. Inc., 1992, 0-201-56317-7

[SunSoft94a] SunSoft Press, *pthreads and Solaris threads: A comparison of two user level threads APIs*, Early Access Edition, Sun Microsystems Inc., May 1994

[SunSoft94b] SunSoft Press, *Generic Audio Device Interface*, Solaris 2.4 Reference Manual Answerbook, Sun Microsystems Inc., 1994

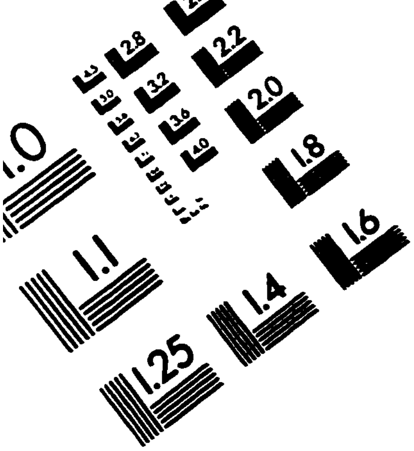
[SunSoft94c] SunSoft Press, *Multithreaded Programming Guide*, Solaris 2.4 Software Developer Answerbook, Sun Microsystems Inc., 1994

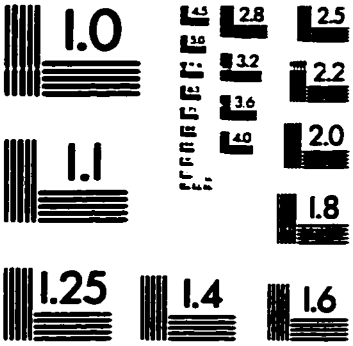
[SunSoft94d] SunSoft Press, *Security, Performance, and Accounting Administration*, Solaris 2.4 System Administrator Answerbook, Sun Microsystems Inc., Mar. 1994

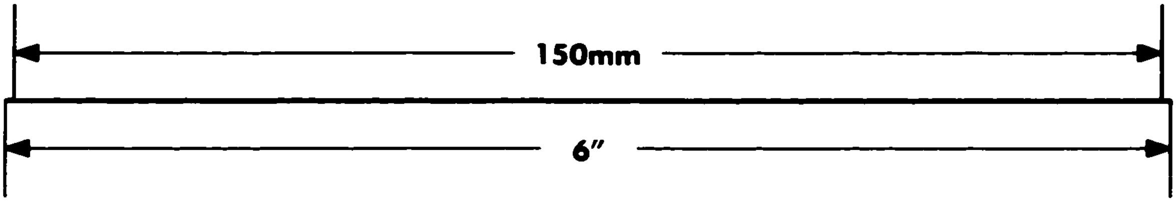
[SunSoft94e] SunSoft Press, *System Services Guide*, Solaris 2.4 Software Developer Answerbook, Sun Microsystems Inc., 1994

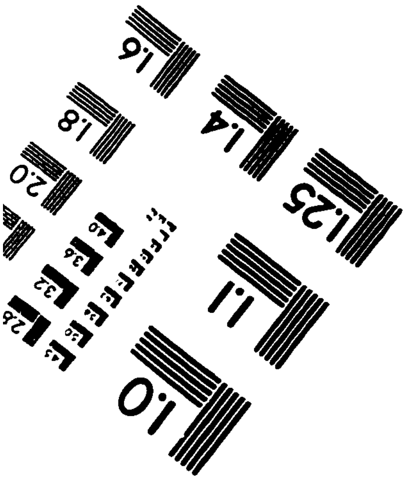
[Young96] Peter Young, *Virtual Reality Systems*, Visualisation Research Group, Centre for Software Maintenance, University of Durham, http://www.dur.ac.uk/%7Edcs3py/pages/work/Documents/VR-Survey/, February 1996

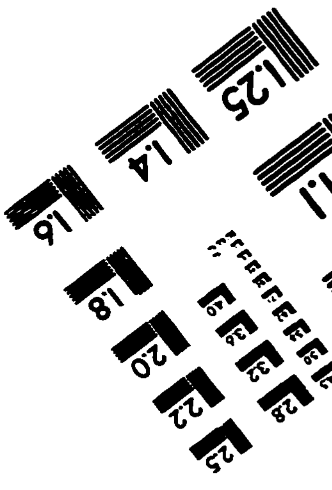
TEST TARGET (QA-3)











APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989